软件配置指南

Inspur INOS 11.3.1(Inspur 6650&6850 交换机)

首版日期：2016-08-03

中国总部
浪潮思科网络科技有限公司
地址：中国山东省济南市高新区浪潮路 1036 号浪潮科技园
网址：http://www.icntnetworks.com
联系电话：400-691-1766

前言

文档规范

本文档会采用下列命令规范：

规范	描述
^或 Ctrl	尖角号^和 Ctrl 均表示键盘上的 Ctrl 键。例如，**^D** 或 **Ctrl-D** 表示应该在按下 D 键的同时，按下键盘上的 Ctrl 键（文档中会以大写字母表示键位，但在实际使用中不区分大小写）。
粗体字	命令、关键字和用户输入的信息均以**粗体字**表示。
斜体字	文档的标题、首次出现的技术术语，以及应该由配置人员提供的具体参数均以*斜体字*表示。
`Courier` 字体	终端会话和系统显示的信息均以 `Courier` 字体表示。
加粗的 **`Courier`** 字体	用户必须输入的文字由加粗的 **`Courier`** 字体表示。
[x]	方括号中的参数为可选参数。
…	在命令语法的后面添加省略号（即 3 个连续的无空格不加粗英文句号）表示这个参数可以重复添加。
|	这条称为管道符的竖线表示需要从多个关键词或参数中选择一个来使用。
[x | y]	如同时有多个可选关键字可以输入，则这些关键字都会置于方括号中，并相互之间用管道符隔开。
{x | y}	如必须从几个关键字中选择一个输入，则这些关键字都会置于大括号中，并相互之间用管道符隔开。
[x {y | z}]	方括号与大括号嵌套使用表示用户可以视需要从这些可选或备选参数中选择一个参数使用，或者必须从这些可选或备选参数中选择一个参数使用。在方括号中嵌套一个包含管道符的大括号，表示在这个可选参数中包含一个必选项。
string	不带引号的字符串。不要再字符串前后使用引号，否则引号也会被包含在字符串中。①
< >	非打印字符②（如密码）会置于尖括号中。
[ ]	方括号中显示的选项为系统默认执行的操作。
!, #	命令行之前出现叹号（!）或井号（#）表示这句话是备注信息。

① 例如，当用户输入如旗标（banner）这类信息时，不要再在输入的信息前后添加引号，除非用户确实希望输入的信息中包含信号。——译者注

② 即用户输入时，系统也不会显示的信息。——译者注

读者提示信息的规范

本文档会采用下列规范插入读者提示信息：

注释： 表示*读者应该注意*。注释信息中包含的是一些对于读者很有帮助的建议，或者本材料中没有包含的参考文件。

提示： 表示*下面的信息可以帮助读者解决实际的问题*。

注意： 表示*读者此时应该提高警惕*。此时，读者的操作有可能会引发设备故障或者数据丢失。

省时： 表示*这里描述的内容可以节省用户的时间*。读者如果执行这一段文字中提到的操作可以达到事半功倍的效果。

警告： 重要的安全提示信息。

警告的标记是在提示安全风险。此时，用户的操作有可能会导致自己受到人身伤害。在操作任何设备时，都要了解电路有可能给人体带来的风险，对防止出现意外的操作流程做到耳熟能详。要用每条警告信息最后的编号找到这台设备所携带的安全警告信息译本。编号 1071。

将这条信息记录下来。

相关文档

注释： 在对设备进行安装和升级之前，请参考设备的版本信息。

- Inspur Inspur 6650 交换机文档：
  http://www.icntnetworks.com
- Inspur SFP、SFP+和 QSFP+模块文档，包括兼容性矩阵：
  http://www.icntnetworks.com
- 错误消息解码器：
  http://www.icntnetworks.com

获取文档与提交服务申请

要了解关于如何获取文档、提交服务申请和收集其他信息的方法，可以阅读《全新浪潮产品文档》月刊，这份文档中会提供所有最新和刚刚更新的 Inspur 技术文档，获取连接为：
http://www.icntnetworks.com
用户可以以 RSS feed 的形式订阅《全新浪潮产品文档》，通过阅读软件让文档信息直接发送到桌面。RSS feed 为免费服务，Inspur 目前支持 RSS 2.0 版。

命令行界面的使用

关于使用命令行界面的信息

命令模式

Inspur INOS 系统的用户界面分为很多不同的模式。用户可以使用的命令取决于其当前所在的模式。在系统提示符下输入问号（?）可以看到当前这种命令模式下可以使用的命令。
用户可以通过控制台（后文称 console）连接、Telnet 连接、SSH 连接或者浏览器来发起一条 CLI 会话。
在发起会话时，用户会首先进入到用户模式下，这种模式通常称为用户 EXEC 模式。用户 EXEC

模式下可以使用的配置命令相当有限。比如，大部分用户 EXEC 命令都是一次性的命令，例如显示当前配置状态的 **show** 命令、清空计时器或接口的 **clear** 命令等。当设备重新启动时，用户 EXEC 命令是不会保存的。

要想能够使用所有命令，必须进入特权 EXEC 模式下。一般来说，用户必须输入一个命令才能进入到特权 EXEC 模式下。在这个模式下，用户可以输入特权 EXEC 命令或者进入全局配置模式。

在配置模式下（无论是全局配置模式、接口配置模式还是线路配置模式），用户可以对当前的运行配置进行修改。如果保存配置文件，那么这些配置命令就会保存下来，在设备重启之后仍然会生效。要想进入各类配置模式，必须首先进入到全局配置模式当中，然后再从全局配置模式进入接口配置模式和线路配置模式。

下表描述了主要的命令模式、进入各个模式的方法、各个模式的命令提示符以及如何离开这个模式。

表 1：命令模式总结

模式	**进入方法**	**命令提示符**	**离开方法**	**关于这个模式**
用户 EXEC 模式	使用 telnet、SSH 或 console 线路发起连接	Device>	输入 **logout** 或 **quit**	在这个模式下可以： • 修改终端设置 • 执行基本测试 • 显示系统信息
特权 EXEC 模式	在用户 EXEC 模式下输入命令 **enable**	Device#	输入 **disable**	在这个模式下，可以查看用户输入的命令。可以使用密码来限制对这个模式的访问
全局配置模式	在特权 EXEC 模式下输入命令 **configure**	Device(config)#	要退出到特权 EXEC 模式，输入 **exit** 或 **end**，或者按 **Ctrl-Z**	在这个模式下，可以配置那些应用于整台设备的参数
VLAN 配置模式	在全局配置模式下输入命令 **vlan** *vlan-id*	Device(config-vlan)#	要退出到全局配置模式，输入命令 **exit**。 要返回特权 EXEC 模式，按 **Ctrl-Z** 或者输入 **end**	在这个模式下，可以配置 VLAN 参数。如果 VTP 工作在透明模式下，那么用户可以在这个模式下创建扩展范围的 VLAN（即 VLAN ID 大于 1005 的 VLAN）并且将配置保存到设备的启

				动配置文件中
接口配置模式	在全局配置模式下输入命令 **interface**（及接口编号）	Device(config-if)#	要退出到全局配置模式，输入命令 **exit**。 要返回特权 EXEC 模式，按 **Ctrl-Z** 或者输入 **end**	在这个模式下，可以配置以太网端口的参数
线路配置模式	在全局配置模式下使用命令 **line vty** 或 **line console** 指定一条线路	Device(config-line)#	要退出到全局配置模式，输入命令 **exit**。 要返回特权 EXEC 模式，按 **Ctrl-Z** 或者输入 **end**	在这个模式下，可以配置终端线路的参数

理解命令的缩写形式

用户只需要把命令输入到设备足以分辨出这条命令的那个字母即可。
下面这个示例显示了如何用缩写的形式输入特权 EXEC 命令 **show configuration**：
Device# show conf

命令的 no 形式与 default 形式

几乎所有命令都有 **no** 的形式。简而言之，no 这个关键字的作用是禁用一项特性或者功能，或者逆向执行某条命令的操作。比如，接口配置模式下的命令 **no shutdown** 可以逆向执行关闭接口的命令。如果在输入这条命令时没有包含 **no** 这个关键字，设备就会重新启用之前已经禁用的特性，或者启用一项在默认状态下即为禁用的特性。
配置命令也有一种 **default** 形式。命令的 **default** 形式可以将这条命令的设置恢复为默认设置。鉴于绝大多数命令在默认状态下都是禁用的，因此对于这些命令来说，**default** 形式与 **no** 形式是相同的。不过，也有一些命令在默认状态下是启用的，这些命令的某些变量会被设置为某个默认值。此时，命令的 **default** 形式就会将这些变量恢复为默认值。

CLI 错误消息

下表显示了用户在使用 CLI 界面配置设备的过程中，有可能遇到的一些错误消息。
表 2：常见的 CLI 错误消息

错误消息	**含义**	**如何获取帮助信息**
`% Ambiguous command: "show con"`	输入的内容尚不足以让设备识别出这条命令	再次输入这条命令，并且在命令之后输入一个问号（**?**），命令行和问号之间不要有空格。

		此时，这条命令后面还可以输入哪些关键词就会显示出来
`% Incomplete command.`	没有输入这条命令所包含的所有必需关键字或参数	再次输入这条命令，并且在命令之后输入一个问号（?），并且在命令行和问号之间留出一个空格。 此时，这条命令后面还可以输入哪些关键词就会显示出来
`% Invalid input detected at '^' marker.`	命令输入有误。尖角号（^）所指即为输入有误之处。	输入问号（?）让系统显示所有在这种命令模式下可以使用的命令。 此时，这条命令后面还可以输入哪些关键词就会显示出来

配置日志记录

用户可以使用日志记录对设备配置所作的修改，并且随时查看日志信息。用户可以使用配置修改日志记录与通告（Configuration Change Logging and Notifications）特性来追踪各个用户各个会话对配置所作的修改。日志记录特性会追踪设备上应用的每条配置命令、输入各个配置命令的用户、输入各个配置命令的时间，以及这条命令的解析器返回代码。这个特性中包含了一种只要配置出现变更，就向注册应用发送异步通告的机制。用户可以选择将通告信息发送到系统日志当中。

注释： 只有通过 CLI 或 HTTP 所修改的配置会被日志记录下来。

使用帮助系统

用户可以在系统提示符中输入问号（?）让系统显示各个命令模式下可以使用的命令，也可以就这些命令获取一个相关关键字或参数的列表。

总步骤

1. help

2. *abbreviated-command-entry* **?**

3. *abbreviated-command-entry* <Tab>

4. ?

5. *command* **?**

6. *command keyword* **?**

具体步骤

	命令或操作	目的
步骤 1	**help** **示例：**	在任一命令模式下获取帮助系统的简短描述信息

	Device# **help**	

步骤 2	*abbreviated-command-entry* **?**
	
	示例：
	`Device# di?`
	`dir disable disconnect`
		获取以某个字符串开头的命令列表

步骤 3	*abbreviated-command-entry* <Tab>
	
	示例：
	`Device# sh conf<tab>`
	`Device# show configuration`
		补全一条只输入了一部分的命令

步骤 4	**?** 示例： `Device> ?`	显示某种命令模式下所有可以使用的命令
步骤 5	*command* **?** 示例： `Device> show ?`	显示某条命令相关的关键字

步骤 6	*command keyword* **?**
	
	示例：
	`Device(config)# cdp holdtime ?`
	`<10-255> Length of time (in sec)`
	`that receiver must keep this packet`
		显示某个关键字相关的参数

如何使用 CLI 界面配置特性

配置命令历史

软件提供了一种历史（或曰命令记录）特性来记录用户曾经输入的命令。命令历史特性在需要回忆一些冗长的或者复杂的命令与条目时（包括访问控制列表）格外能够发挥用处。用户可以自定义这一特性来满足自己的需求。

修改命令历史缓冲区大小

在默认情况下，设备会在历史缓冲区中记录 10 条命令。不过用户可以针对当前的终端会话，或者针对某条线路的所有用户会话修改这个参数。这个流程是可选的。

总步骤

1. terminal history [**size** *number-of-lines*]

具体步骤

	命令或操作	**目的**

步骤 1	**terminal history** [**size** *number-of-lines*] **示例：** `Device# terminal history size 200`	在特权 EXEC 模式下，修改设备在当前终端会话中记录的命令行数量。用户可以将这个参数修改为 0 到 256 之间的任意值

查看命令

要查看历史缓冲区中记录的命令，可以执行下表中的其中一项操作。这一步操作是可选的。

注释： 只有在可以兼容 ANSI 的终端（如 VT100）上可以使用方向键。

总步骤

1. Ctrl-P 或使用**向上的方向键**

2. Ctrl-N或使用**向下的方向键**

3. show history

具体步骤

	命令或操作	目的
步骤 1	**Ctrl-P** 或使用**向上的方向键**	查看历史缓冲区中的命令，系统会首先显示最近输入的命令。重复按键可以让系统继续显示前一条输入的命令
步骤 2	**Ctrl-N** 或使用**向下的方向键**	在通过 Ctrl-P 或向上的方向键查看历史缓冲区之后，返回最近输入的命令。重复按键可以让设备按时间顺序显示后一条输入的命令
步骤 3	**show history**	列出在特权 EXEC 模式中最后输入的几条命令。用户可以通过修改全局配置模式下的 **terminal history** 命令和线路配置命令模式下的 **history** 命令来控制系统显示的命令数量

禁用命令历史特性

命令历史特性是自动启用的。用户可以针对当前的终端会话或者整个命令行禁用这个特性。这个流程是可选的。

总步骤

1. terminal no history

具体步骤

	命令或操作	目的
步骤 1	**terminal no history** **示例：** `Device# terminal no history`	在特权 EXEC 模式下，针对当前终端会话禁用这一特性

启用与禁用编辑特性

虽然增强的编辑模式是自动启用的，但用户也可以禁用并重新启用这个特性。

总步骤

1. terminal editing

2. terminal no editing

具体步骤

	命令或操作	目的
步骤 1	**terminal editing** **示例：** `Device#` **terminal editing**	在特权 EXEC 模式下，针对当前终端会话重新启用增强的编辑模式
步骤 2	**terminal no editing** **示例：** `Device#` **terminal no editing**	在特权 EXEC 模式下，针对当前终端会话禁用增强的编辑模式

使用快捷键编辑命令

快捷键可以在用户需要编辑命令时提供帮助。快捷键的使用是可选的。

注释： 只有在可以兼容 ANSI 的终端（如 VT100）上可以使用方向键。

表 3：编辑命令

编辑命令	描述
Ctrl-B 或使用**向左的方向键**	将光标向回移动一个字符
Ctrl-F 或使用**向右的方向键**	将光标向前移动一个字符
Ctrl-A	将光标移动到命令行的开始
Ctrl-E	将光标移动到命令行的末尾
Esc B	将光标向回移动一个词
Esc F	将光标向前移动一个词
Ctrl-T	将光标所在的字符与光标左侧的字符交换
Delete 或 **Backspace 键**	清除光标左侧的字符
Ctrl-D	删除光标所在的字符
Ctrl-K	删除所有从光标所在位置到命令行结尾的字符
Ctrl-U 或 **Ctrl-X**	删除所有从光标所在位置到命令行起始的字符
Ctrl-W	删除光标左侧的词
Esc D	删除从光标所在位置到这个词结尾的字符
Esc C	将光标所在的字符转化为大写字母
Esc L	将光标所在的字符转换为小写字母
Esc U	将从光标所在位置到这个词结尾的字符转换为大写字母
Ctrl-V 或 **Esc Q**	将某个组合键指定为一条快捷的可执行命令
回车键	如果终端的屏幕无法显示全部信息，则向下滚动一行或一屏的显示信息 注释：More 这个提示符的作用是告诉管理员，还有一些信息终端屏幕上没有显示出来，比如在使用 show 命令查看输入信息时有时就会显示这个提示符。只要用户看到 More 这个提示符，就可以使用**回车键**或者**空格键**查看后面的信息。

空格键	向下滚动一屏
Ctrl-L 或 **Ctrl-R**	在设备突然向屏幕中发送了一条消息的情况下，重新显示当前的命令行

编辑缩进的命令行

有时，命令的长度会超出屏幕一行可以显示的宽度，此时用户可以使用命令的缩进特性。当光标到达最右端时，命令行就会向左边转换 10 个空格。此时，用户就看不到命令行最前面的 10 个字符了，但是用户可以将光标回滚，查看在命令最初输入的信息。这个快捷键是可选的。

要回滚到命令条目的最开始，可以反复按 **Ctrl-B** 或**向左的方向键**，也可以按下 **Ctrl-A** 让光标直接移动到这一行的最开始。

注释： 只有在可以兼容 ANSI 的终端（如 VT100）上可以使用方向键。

下面的示例显示了如何缩进长度超出了屏幕一行宽度的命令。

总步骤

1. access-list

2. Ctrl-A

3. Return key

具体步骤

	命令或操作	目的
步骤 1	**access-list** **示例：** `Device(config)#` **access-list 101 permit tcp 10.15.22.25 255.255.255.0 10.15.22.35** `Device(config)# $` **101 permit tcp 10.15.22.25 255.255.255.0 10.15.22.35 255.25** `Device(config)# $`**t tcp 10.15.22.25 255.255.255.0 131.108.1.20 255.255.255.0 eq** `Device(config)# $`**15.22.25 255.255.255.0 10.15.22.35 255.255.255.0 eq 45**	显示长度超过一行的全局配置命令。 当光标第一次到达一行的最末端时，这一条就会向右缩进 10 个空格并且重新显示。美元符号（$）表示这一行向左缩进过。每当光标到达一行最右端时，这一行信息都会向左缩进 10 个空格
步骤 2	**Ctrl-A** **示例：** `Device(config)#` **access-list 101 permit tcp 10.15.22.25 255.255.255.0 10.15.2$**	查看完整的配置语句。 美元符号（$）出现在一行的最末端，表示这一行向右缩进过。
步骤 3	**回车键**	执行这条命令。 软件会认为终端屏幕的宽度是 80 列。如果用户的宽度并不是 80 列，可以使用特权 EXEC 模式下的命令 **terminal width** 来修改终端的宽度。 用户可以借助命令历史特性

		来回顾和修改之前输入过的，这些包含缩进的复杂命令及条目

搜索和过滤 show 和 more 命令的输出信息

用户可以搜索和过滤 **show** 和 **more** 命令显示的输出信息。当用户需要从大量输出信息中寻找自己所需的信息，或者希望输出信息中不包含某些无用信息时，就可以采取这种做法。这些命令都是可选的。

总步骤

1. {**show** | **more**} *command* | {**begin** | **include** | **exclude**} *regular-expression*

具体步骤

	命令或操作	目的
步骤 1	{**show**	**more**} *command*	{**begin**	**include**	**exclude**} *regular-expression* **示例：** `Device# show interfaces	include protocol` `Vlan1 is up, line protocol is up` `Vlan10 is up, line protocol is down` `GigabitEthernet1/0/1 is up, line protocol is down` `GigabitEthernet1/0/2 is up, line protocol is up`	搜索并过滤输出信息。 正则表达式是区分大小写的。比如，如果输入**| exclude output**，那么包含 **output** 的那些输出信息行就不会显示出来，但包含 **output** 的信息还是会显示。

访问 CLI 界面

用户可以通过 console 连接、通过 Telnet、SSH 或者浏览器来访问 CLI 界面。

用户可以通过主用交换机来管理交换机堆栈和堆栈成员接口，但不能分别管理堆栈的各个成员交换机。用户可以通过一台或多台堆栈成员的 console 端口或以太网管理端口来连接主用交换机。如果用户打算向主用交换机发起多条 CLI 会话，一定要小心，因为你在一条会话中输入的命令并不会在另一条会话中显示出来。因此，用户容易忘记自己输入的命令。

注释： 在管理交换机堆栈时，我们推荐只建立一条 CLI 会话。

如果用户想要配置某个堆栈成员端口，那就一定要在 CLI 命令接口编号中包含堆栈成员的编号。

要想对备用交换机进行调试，可以在主用交换机上使用特权 EXEC 命令 **session standby INOS** 来访问备用交换机的 INOS 控制台。要对某个堆栈成员进行调试，可以在主用交换机上使用特权 EXEC 命令 **session switch** *stack-member-number* 来访问堆栈成员的 CLI 操作界面。如需了解这些命令的信息，可以查看交换机的命令指南。

通过 Console 连接或 Telnet 来访问 CLI 界面

在访问 CLI 之前，用户必须将一台终端或者 PC 连接到设备的 console 接口，或者将一台 PC 连接设备的以太网管理端口，然后再给设备加电。这一点在所有设备附带的硬件安装指南中都会提到。

如果这台设备已经进行了配置，那么用户就既可以通过本地的 console 连接来访问 CLI 界面也可以通过远程的 Telnet 会话来访问设备的 CLI 界面，但设备必须首先针对远程访问进行了配置。

用户可以使用下列方法之一来与这台设备建立连接：

- 将一台管理工作站或或拨号调制解调器连接到设备的 console 端口，或者将一台 PC 连接到设备的以太网管理端口。如需了解连接 console 端口或以太网管理端口的信息，可以查看设备的硬件安装指南。
- 使用 Telnet TCP/IP 或者加密的安全外壳（SSH）从远端管理工作在连接设备。此时，这台设备必须能够与 Telnet 或 SSH 客户端之间通过网络建立连接，同时这台设备还必须配置有进入特权 EXEC 模式的加密密码。
    - 设备支持最多 16 条并行的 Telnet 会话。任何一位 Telnet 用户对设备所作的修改都会影响到所有 Telnet 会话。
    - 设备支持最多 5 条并行的 SSH 会话。

在通过 console 端口、以太网管理端口、Telnet 会话或 SSH 会话建立连接之后，用户就可以在管理工作站上看到这台设备用户 EXEC 模式的提示符了。

第 1 部分 音视频桥接

音视频桥接

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个

特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

音视频桥接网络简介

关于音视频桥接技术（AVB）的信息

过去，音频设备和视频设备采用的都是模拟的单用途点到点单向链路。虽然音视频设备后来转而使用数字传输链路，但点到点单向链路的结构却依旧延续了下来。采用专用连接的模型让音视频设备必须通过大量冗杂的线缆来连接各类专业应用和用户应用，这使得布线很难进行管理和操作。
为了更快适应基于以太网的音频/视频部署方案，用一种可以实现互操作的方式部署音频/视频设备，IEEE 指定了 IEEE 音视频桥接标准，即 IEEE 802.1BA 标准。这种标准定义了一种端点设备和网络能够实现高度融合的机制，让高质量的 A/V 数据流可以通过以太网架构在用户应用和专业音频视频设备之间传输。
注释： 堆栈系统不支持 AVB；
EtherChannel 接口不支持 AVB；
只有启用了 STP 的网络可以支持 AVB。
提供 AVB 支持的许可证
只有下列两个级别的许可证可以支持 AVB：
- ipbase
- ipservices

AVB 的优势
AVB 是一种标准的机制，它可以让音频-视频流量在以太网中传输，这项标准的优势在于：
- 可以对最大延迟值给出保障；
- 可以做到事件同步
- 可以对带宽提供保障
- 可以提供专业传输

AVB 网络的组成
只有在每台设备都支持 AVB 的环境中，才可以运行 AVB 协议。AVB 网络包含 AVB 数据源（AVB talker）、AVB 接收方（AVB listener）、AVB 交换机和主时钟源。
- AVB 数据源——一类 AVB 终端，是数据流的源或生成设备。如麦克风、摄像机等即为此类；
- AVB 接收方——一类 AVB 终端，是数据流的目的或读取设备。如扬声器、显示器等即为此类；
- AVB 交换机——符合 IEEE802.1 AVB 标准的以太网交换机；
- AVB 数据流——流预留遵循 SRP（流预留协议）的数据流。

注释： 有时候，我们会使用“网桥”一词。此时，我们指的就是交换机。
IEEE 802.1BA 标准要求 AVB 数据源必须能够充当主时钟源。在常见的网络部署方案中，网络节点也可以充当主时钟源，前提是这个网络节点是时钟源或者这个网络节点可以从一台能够

充当主时钟源的设备那里获取到时间信息，并且使用 IEEE 802.1AS 向 AVB 网络提供时间信息。

图 1 所示为一个简单的 AVB 网络示例，其中包含了 AVB 网络的各个不同组件。

图 1：AVB 网络

图原文	图译文
Audio Player	音频播放器
AVB Talker	AVB 数据源
AVB Switch	AVB 交换机
AVB Switch	AVB 交换机
Microphones	麦克风
AVB Talker	AVB 数据源
Speaker	扬声器
AVB Listener	AVB 接收方

音频/视频端点（如麦克风、扬声器等等）常常是模拟设备。AVB 端点的制造商增加了 DSP（数字信号处理器，Digital Signal Processors）和 I/O 设备，以便对音频/视频执行密集的处理，同时将端点的流量汇总到一个 AVB 以太网接口，如图 2 所示。

图 2：制造商的音频 I/O 系统

图原文	图译文
Microphones	麦克风
Vendor I/O	制造商 I/O
AVB Switch	AVB 交换机

支持 AVB 的 SKU

下列 Inspur 6850 和 Inspur 6650 SKU 可以支持 AVB：

- S6650-24TD
- S6650-48TD
- S6650-48TQ

关于 gPTP（通用精确时间协议）的信息

通用精确时间协议（gPTP）是一个 IEEE 802.1AS 标准，这种协议提供的机制可以让一个 AVB 网络中的网桥与端点设备实现时钟同步。协议定义了如何从多个可感知时间的网桥、数据源和接收方中选举主时钟（BMCA）的机制。在可感知事件的网络中，需要建立一个时钟分层结构，而主时钟就是这个分层结构的根，自主时钟发出的时间信息会逐层分发给网络的其他节点，实现时钟的同步。

时间同步也需要对链路延迟和网络节点中的交换机延迟进行研判。而 gptp 交换机属于 IEEE 1588 边界时钟，这类交换机也会使用对等体到对等体的延迟机制来判断链路的延迟时间。交换机计算出来的延迟值会包含在 PTP 消息的修正字段中，被一同转发给端点设备。数据源和接收方会以这个 gPTP 时间作为共享的时钟参考点，用它来接替和恢复媒体时钟。gPTP 目前只定义了域 0，这也就是交换机支持的域。

在那些被 STP 协议阻塞的端口上，同样会运行对等体到对等体的中继机制。但其他 PTP 消息都不会通过被阻塞的端口对外转发。

在一个 PTP 域中，最佳主时钟（BMC）算法会将时钟和端口用一种分层的形式进行排列，其中包括下列时钟和端口状态：

时钟：

- 主时钟（GM/GMC）
- 边界时钟（BC）

端口状态

- 主（Master）
- 从（Slave）
- 被动（Passive）

关于多流预留协议（MSRP）的信息

多流预留协议（Multiple Stream Reservation Protocol）为终端站点提供了一种预留网络资源的机制，其目的在于保障数据流的收发按照终端所请求的 QoS 来完成。这是一台 AVB 设备（包括数据源、接收方和交换机）上的核心协议之一。这个协议可以让数据源通过一个由 AVB 交换机组成的网络来通告数据流，并且让接收方通过注册来接收这些数据流。

MSRP 是支持 AVB 的核心软件协议模块。它可以实现数据流的建立与断开。这个协议可以与 gPTP 协议一起使用，以更新数据流的延迟信息，也可以与 QoS 策略一起使用来调配硬件资源，保障有充足的带宽可以转发数据流。此外，这个协议也可以给令牌整形器（credit based shaper）提供其所需的 QoS 整形参数。

MSRP 的功能

MSRP 可以执行下列功能：

- 让数据源通告数据流，让接收方发现并注册数据流；
- 在数据源和（一个或多个）接收方之间通过以太网建立一条数据通道；
- 为 AVB 数据流提供带宽保障；
- 保证流量不超出延迟上限；
- 发现并通告数据源与各个接收方之间最大的端到端延迟；
- 当数据源和接收方之间的路径无法满足带宽需求时，报告错误原因和错误出现的位置；
- 支持不同延迟目标的多种流量类型；
- 限制 AVB 流量以确保尽力而为流量（best effort traffic）的资源不被完全挤占（starvation）；
- MSRP 数据源的宣告消息不会通过那些被 STP 阻塞的端口转发出去；
- MSRP 会侦听 STP TCN 通告消息，以创建 MSRP 宣告消息断开/修改/建立数据流。

关于 HQoS 的信息

AVB 网络可以为那些对传输时间比较敏感的音频和视频流量提供带宽和最小延迟的保障。AVB 根据从数据源到接收方之间的最大延迟值目标，将类型 A 和类型 B 定义为了时间敏感型数据流。

这两类数据流的延迟目标值如下：

- SR-类型 A：2ms
- SR 类型 B：50ms

将每一跳的最大延迟值相加就可以得到类型 A 的总端到端延迟（小于 2ms）或类型 B 的总端到端延迟（小于 50ms）。数据源和接收方相隔 7 跳的这种典型的 AVB 部署方案就可以满足上述延迟值的要求。

优先级代码点的作用是将流量映射为某种数据流。数据帧的转发行为就是依据这个优先级来

决策的。令牌整形器的作用是按照管理员给某条出站队列所预留的带宽，来对这些数据流的传输进行整形，让这类数据流的延迟值可以满足要求。

自 Inspur XE Denali 16.3.2 起，AVB 就开始支持分层 QoS。AVB 分层 QoS 策略是一种双层的嵌套策略（Parent-Child Policy）。AVB 父策略（Parent Policy）会将音频、视频流量（即 SR 类型 A、SR 类型 B 流量）和网络控制数据包这三类数据，与标准的以太网流量（即非 SR 流量）隔离开，并且按照不同的方式进行处理。分层 QoS 让用户可以在不同的策略级上指定 QoS 操作，这样做的目的是为了增加流量管理的颗粒度①。通过分层策略，用户可以：

- 使用父策略对子策略中的不同队列进行整形；
- 针对汇总流量应用某种 policy map，对其进行操作；
- 针对某种流量类型应用 policy map 策略。

① 增加颗粒度是业内常见用语，表“细化”之意。所谓增加流量管理的颗粒度，也即细化对不同类型流量的管理。——译者注

用户可以通过 **policy-map** *AVB-Output-Child-Policy* 和 **policy-map** *AVB-Input-Child-Policy* 这两条命令，来仅针对入站和出站方向 HQoS 子策略的 class-map 及其操作进行修改。

注释： 用户不应该修改子策略中的 PCP 来映射父策略中配置的 PCP。比如，SR 类型 A cos 3 和 SR 类型 B Cos2。

分层限速（Hierarchical Policing）

入站接口和出站接口都支持部署分层限速。分层 QoS 会将 SR 类和非 SR 类相关的规则分别放入父策略和子策略中。AVBSR 类完全是由 MSRP 客户端进行控制的，因此包含 SR 类型属性的父策略都是由 MSRP 管理的。终端用户对那些包含非 SR 类属性的子策略拥有完全的控制权，因此终端用户只能对子策略进行修改。

AVB HQoS 策略是可以由用户进行修改的，如果用户将配置保存到了启动配置中，那么设备就会保存用户所作的配置。所以，即使设备进行了重启，重启后 AVBHQoS 子策略的配置依然会生效。

关于多 VLAN 注册协议（MVRP）的信息

多 VLAN 注册协议（MVRP）是一种基于 MRP 的应用。MVRP 可以针对每个 VLAN ID 动态维护动态 VLAN 注册条目中的内容，并且将它们的信息转发给其他的网桥。这些信息可以帮助那些启动了 MVRP 的设备动态建立和更新自己的数据库，了解哪些 VLAN ID 所对应的 VLAN 中当前有活动的成员设备，以及可以通过哪些端口访问这些设备。

从 AVB 的角度来看，MVRP 是数据源和接收方必须使用的协议。MVRP 是独立于 AVB 的，它是 VLAN 交换机上的一种 IEEE 802.1Q 需求。但对于 AVB 环境来说，在交换机上手动配置 VLAN 也没有问题。

注释： 要想让 MVRP 正常工作，就需要让 VTP 工作在禁用模式或者透明模式下。

配置 AVB 网络

配置 AVB

在这一节中，我们会描述可以针对 AVB 所作的各类配置。

在交换机上启用 AVB

用户可以在交换机上使用下列命令来启用 AVB。

总步骤

1. **enable**
2. **configure terminal**
3. **avb**
4. **end**

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**avb** 示例： `Device(config)# avb`	在交换机上启用 AVB
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

接下来做什么？

要在交换机上禁用 AVB，应该使用这条命令的 **no** 形式。

在设备上配置 AVB

用户可以沿着连接路径，使用下面的命令将 AVB 设备的接口配置为 dot1q trunk 端口。

总步骤

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **switchport mode trunk**
5. **exit**
6. **vlan 2**
7. **avb**
8. **end**

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device#` **configure terminal**	
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface te1/1/1**	定义要配置为 trunk 的接口，并进入接口配置模式
步骤 4	**switchport mode trunk** 示例： `Device(config-if)#` **switchport mode trunk**	将这个端口配置为 trunk 端口
步骤 5	**exit** 示例： `Device(config-if)#` **exit**	返回全局配置模式
步骤 6	**vlan 2** 示例： `Device(config)#` **vlan 2**	在交换机上配置 Vlan 2
步骤 7	**avb** 示例： `Device(config-vlan)#` **avb**	在特定接口上启用 AVB
步骤 8	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式

接下来做什么？

要在交换机上禁用 AVB，应该使用这条命令的 **no** 形式。

配置 gPTP

在这一节中，我们会描述可以针对 gPTP 所作的各类配置。

启用 gPTP

当交换机上启用了 AVB 时，交换机就会针对 AVB 启用 gPTP。

用户也可以使用下面的命令启用 gPTP。

总步骤

1. enable

2. configure terminal

3. ptp profile dot1as

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ptp profile dot1as** 示例： `Device(config)# ptp profile dot1as`	在端口上启用 gPTP
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置 PTP 时钟值

用户可以使用下面的命令配置 ptp 时钟优先级 1 和优先级 2 的值。

总步骤

1. enable

2. configure terminal

3. ptp priority1

4. ptp priority2

5. exit

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ptp priority1** 示例： `Device(config)# ptp priority1`	配置 ptp 时钟优先级 1 的值 0-255：这是 ptp 时钟优先级的取值范围。用户应在这个范围内选择一个数值。 **注释：** 如果将优先级 1 的数值配置为 255，那么这个时钟就无法成为主时钟。
步骤 4	**ptp priority2**	配置 ptp 时钟优先级 2 的值 0-255：这是 ptp 时钟优先级的取值范围。用户应在

	示例： `Device(config)# ptp priority2`	这个范围内选择一个数值。
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置 HQoS

在这一节中，我们会描述可以针对 HQoS 所作的各类配置。

启用 HQoS

当交换机上启用了 AVB 时，交换机就会针对 AVB 启用 HQoS。

从平面策略迁移到分层策略——指导方针与限制条件

在将针对 AVB 的平面策略迁移到分层策略时，用户可以参照下面的指导方针：

如果将系统从 Inspur INOS XE Denali 16.3.1 系统升级到 Inspur INOS XE Denali 16.3.2 系统，那么设备启动配置文件中保存的 QoS 策略就会出错，并且失效。读者可以按照下面的步骤在设备上正确配置 HQoS 策略：

1 在全局使用 **no avb** 命令禁用 AVB

注释： 在禁用 AVB 时，所有 policy-map 和 class-map 都会自动从配置文件中删除。但访问控制列表并不会自动被删除。用户必须手动删除访问控制列表。用户应该在将系统升级到 Inspur INOS XE Denali 16.3.2 系统之前，删除所有的 QoS 策略。

2 使用 **avb** 命令启用 AVB。在启用 AVB 时，交换机就会针对 AVB 启用 HQoS。

- 我们不推荐将设备从分层策略支持的版本迁移到平面策略支持的版本；
- 用户只能修改子策略。但父策略完全是由 MSRP 管理的；
- 命令 **show running config** 只会显示子策略；
- 从 Inspur INOS XE Denali 16.3.2 开始，命令 **show running config interface** 就不会再显示策略包含的任何详细信息。用户可以使用 **show policy-map interface** 这条命令来显示所有策略中包含的详细信息。

分层 QoS 策略的格式

下面的示例所示为入站接口的分层标记策略：

```
policy-map AVB-Input-Child-Policy
class VOIP-DATA-CLASS
set dscp EF
class MULTIMEDIA-CONF-CLASS
set dscp AF41
class BULK-DATA-CLASS
set dscp AF11
class TRANSACTIONAL-DATA-CLASS
set dscp AF21
class SCAVENGER-DATA-CLASS
set dscp CS1
class SIGNALING-CLASS
set dscp CS3
class class-default
```

```
set dscp default
policy-map AVB-Input-Policy-Remark-AB
class AVB-SR-A-CLASS
set cos 0 (set 0 for boundary & SR class A PCP value for core port)
class AVB-SR-B-CLASS
set cos 0 (set 0 for boundary & SR class B PCP value for core port)
class class-default
service-policy AVB-Input-Child-Policy
policy-map AVB-Input-Policy-Remark-A
class AVB-SR-A-CLASS
set cos 0 (set 0 for boundary & SR class A PCP value for core port)
class class-default
service-policy AVB-Input-Child-Policy
policy-map AVB-Input-Policy-Remark-B
class AVB-SR-B-CLASS
set cos 0 (set 0 for boundary & SR class B PCP value for core port)
class class-default
service-policy AVB-Input-Child-Policy
policy-map AVB-Input-Policy-Remark-None
class class-default
service-policy AVB-Input-Child-Policy
```

下面的示例所示为出站接口的分层队列策略：

```
policy-map AVB-Output-Child-Policy
class VOIP-PRIORITY-QUEUE
bandwidth remaining percent 30
queue-buffers ratio 10
class MULTIMEDIA-CONFERENCING-STREAMING-QUEUE
bandwidth remaining percent 15
queue-limit dscp AF41 percent 80
queue-limit dscp AF31 percent 80
queue-limit dscp AF42 percent 90
queue-limit dscp AF32 percent 90
queue-buffers ratio 10
class TRANSACTIONAL-DATA-QUEUE
bandwidth remaining percent 15
queue-limit dscp AF21 percent 80
queue-limit dscp AF22 percent 90
queue-buffers ratio 10
class BULK-SCAVENGER-DATA-QUEUE
bandwidth remaining percent 15
queue-limit dscp AF11 percent 80
queue-limit dscp AF12 percent 90
queue-limit dscp CS1 percent 80
queue-buffers ratio 15
```

```
class class-default
bandwidth remaining percent 25
queue-buffers ratio 25
policy-map AVB-Output-Policy
class AVB-SR-A-CLASS
priority level 1 (Shaper value based on stream registration)
class AVB-SR-B-CLASS
priority level 2 (Shaper value based on stream registration)
class CONTROL-MGMT-QUEUE
priority level 3 percent 15
class class-default
bandwidth remaining percent 100
queue-buffers ratio 80
service-policy AVB-Output-Child-Policy
```

配置 MVRP

在这一节中，我们会描述可以针对 MVRP 所作的各类配置。

启用 HQoS

用户可以在拓扑里的交换机上启用 MVRP，通过下面的命令将 VLAN 信息发布出去。

注释： 在通过 MVRP 启用动态 vlan 创建之前，必须先将 VTP 模式修改为 **transparent** 或 **off**。

总步骤

1. enable

2. configure terminal

3. mvrp global

4. vtp mode {transparent | off}

5. mvrp vlan create

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**mvrp global** 示例： `Device(config)# mvrp global`	进入 MVRP 全局配置模式
步骤 4	**vtp mode {transparent | off}**	将 VTP 设置为 **transparent** 模式或 **off** 模式

	示例： `Device(config)#` **vtp mode transparent** 示例： `Device(config)#` **vtp mode off**	
步骤 5	**mvrp vlan create** 示例： `Device(config)#` **mvrp vlan create**	在交换机上启用 MVRP

在交换机接口上配置 MVRP

用户可以使用下面的命令在交换机接口上配置 MVRP。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. mvrp registration {*fixed* | *forbidden* | *normal*}

5. mvrp timer {*join* | *leave* | *leave-all* | *periodic*}

6. exit

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface te1/1/1**	定义要配置为 trunk 的接口，并进入接口配置模式
步骤 4	**mvrp registration** {*fixed* | *forbidden* | *normal*} 示例： `Device(config-if)#` **mvrp registration fixed**	将 MVRP 注册到 MAD 实例中。 • fixed：固定注册 • forbidden：禁止注册 • normal：正常注册
步骤 5	**mvrp timer** {*join* | *leave* | *leave-all* | *periodic*}	配置 MVRP 计时器。 • join：计时器控制应用到 ASM 的传输机会的时

	示例： `Device(config-if)#` **mvrp timer join**	间间隔。 • leave：计时器控制 RSM 在 LV 状态等待，然后过渡到 MT 状态。 • leave-all：计时器控制 LeaveAll SM 创建 LeaveAll PDU 的频率 • perioic：周期计时器
步骤 6	**exit** 示例： `Device(config-if)#` **exit**	返回全局配置模式

监控 AVB 网络

监控 AVB

要查看 AVB 的详细信息，可以使用下表中的命令：

命令	目的
show avb domain	显示 AVB 域
show avb streams	显示 AVB 数据流信息

监控 gPTP

要查看 gPTP 协议的详细信息，可以使用下表中的命令：

命令	目的
show ptp brief	显示接口上的简单 ptp 状态
show ptp clock	显示 ptp 时钟信息
show ptp parent	显示父时钟信息
show ptp port	显示 ptp 端口信息
show platform software fed switch active ptp if-id {interface-id}	显示关于端口 ptp 状态的具体信息

监控 MSRP

要查看 MSRP 的详细信息，可以使用下表中的命令：

命令	目的
show msrp streams	显示 MSRP 数据流信息
show msrp streams detailed	显示 MSRP 数据流的详细信息
show msrp streams brief	简化显示 MSRP 数据流的信息
show msrp port bandwidth	显示 MSRP 端口带宽的信息

监控 HQoS

要查看 HQoS 的详细信息，可以使用下表中的命令：

命令	目的
show run	显示所有子 policy map 的详细信息
show policy-map	显示 policy map 配置的详细信息
show policy-map interface *interface-id* **[input | output]**	显示 AVB QoS 统计数据。入站数据包数量和出站字节数都会列入 QoS 统计数据当中

监控 MVRP

要查看 MVRP 的详细信息，可以使用下表中的命令：

命令	目的
show mvrp summary	显示 MVRP 汇总信息
show mvrp interface	显示接口的 MVRP 信息

AVB 的配置与监控示例

AVB 示例

这个示例显示了查看 AVB 域的方法。

```
Device#show avb domain
AVB Class-A
Priority Code Point : 3
VLAN : 2
Core ports : 1
Boundary ports : 67
AVB Class-B
Priority Code Point : 2
VLAN : 2
Core ports : 1
Boundary ports : 67
-------------------------------------------------------------------
-----------
Interface State Delay PCP VID Information
-------------------------------------------------------------------
-----------
Te1/0/1 down N/A Oper state not up
Te1/0/2 down N/A Oper state not up
Te1/0/3 down N/A Oper state not up
Te1/0/4 down N/A Oper state not up
```

```
Te1/0/5 up N/A Port is not asCapable
Te1/0/6 down N/A Oper state not up
Te1/0/7 down N/A Oper state not up
Te1/0/8 down N/A Oper state not up
Te1/0/9 down N/A Oper state not up
Te1/0/10 down N/A Oper state not up
Te1/0/11 down N/A Oper state not up
Te1/0/12 down N/A Oper state not up
Te1/0/13 down N/A Oper state not up
Te1/0/14 down N/A Oper state not up
Te1/0/15 down N/A Oper state not up
Te1/0/16 down N/A Oper state not up
Te1/0/17 down N/A Oper state not up
Te1/0/18 down N/A Oper state not up
Te1/0/19 up N/A Port is not asCapable
Te1/0/20 down N/A Oper state not up
Te1/0/21 down N/A Oper state not up
Te1/0/22 down N/A Oper state not up
Te1/0/23 up N/A Port is not asCapable
Te1/0/24 down N/A Oper state not up
Te1/0/25 down N/A Oper state not up
Te1/0/26 down N/A Oper state not up
Te1/0/27 down N/A Oper state not up
Te1/0/28 down N/A Oper state not up
Te1/0/29 up N/A Port is not asCapable
Te1/0/30 down N/A Oper state not up
Te1/0/31 down N/A Oper state not up
Te1/0/32 down N/A Oper state not up
Te1/0/33 down N/A Oper state not up
Te1/0/34 down N/A Oper state not up
Te1/0/35 up N/A Port is not asCapable
Te1/0/36 down N/A Oper state not up
Te1/0/37 down N/A Oper state not up
Te1/0/38 down N/A Oper state not up
Te1/0/39 up 507ns
Class- A core 3 2
Class- B core 2 2
Te1/0/40 down N/A Oper state not up
Te1/0/41 down N/A Oper state not up
Te1/0/42 down N/A Oper state not up
Te1/0/43 down N/A Oper state not up
Te1/0/44 down N/A Oper state not up
Te1/0/45 down N/A Oper state not up
Te1/0/46 down N/A Oper state not up
```

```
Te1/0/47 down N/A Oper state not up
Te1/0/48 down N/A Oper state not up
Te1/1/1 down N/A Oper state not up
Te1/1/2 down N/A Oper state not up
Te1/1/3 down N/A Oper state not up
Te1/1/4 down N/A Oper state not up
Te1/1/5 down N/A Oper state not up
Te1/1/6 down N/A Oper state not up
Te1/1/7 down N/A Oper state not up
Te1/1/8 down N/A Oper state not up
Te1/1/9 down N/A Oper state not up
Te1/1/10 down N/A Oper state not up
Te1/1/11 down N/A Oper state not up
Te1/1/12 down N/A Oper state not up
Te1/1/13 down N/A Oper state not up
Te1/1/14 down N/A Oper state not up
Te1/1/15 down N/A Oper state not up
Te1/1/16 down N/A Oper state not up
Fo1/1/1 down N/A Oper state not up
Fo1/1/2 down N/A Oper state not up
Fo1/1/3 down N/A Oper state not up
Fo1/1/4 down N/A Oper state not up
```

这个示例显示了查看 AVB 数据流信息的方法。

```
Device#show avb streams
Stream ID: 0011.0100.0001:1 Incoming Interface: Te1/1/1
Destination : 91E0.F000.FE00
Class : A
Rank : 1
Bandwidth : 6400 Kbit/s
Outgoing Interfaces:
------------------------------------------------------------------------
-------
Interface State Time of Last Update Information
------------------------------------------------------------------------
-------
Te1/1/1 Ready Tue Apr 26 01:25:40.634
Stream ID: 0011.0100.0002:2 Incoming Interface: Te1/1/1
Destination : 91E0.F000.FE01
Class : A
Rank : 1
Bandwidth : 6400 Kbit/s
Outgoing Interfaces:
------------------------------------------------------------------------
```

```
-------
Interface State Time of Last Update Information
-------------------------------------------------------------------
-------
Te1/1/1 Ready Tue Apr 26 01:25:40.634
```

这条命令可以用来查看各个接口关于 ptp 状态的简单信息。

```
Device#show ptp brief
Interface
Domain
PTP State
FortyGigabitEthernet1/1/1 0 FAULTY
FortyGigabitEthernet1/1/2 0 SLAVE
GigabitEthernet1/1/1 0 FAULTY
GigabitEthernet1/1/2 0 FAULTY
GigabitEthernet1/1/3 0 FAULTY
GigabitEthernet1/1/4 0 FAULTY
TenGigabitEthernet1/0/1 0 FAULTY
TenGigabitEthernet1/0/2 0 FAULTY
TenGigabitEthernet1/0/3 0 MASTER
TenGigabitEthernet1/0/4 0 FAULTY
TenGigabitEthernet1/0/5 0 FAULTY
TenGigabitEthernet1/0/6 0 FAULTY
TenGigabitEthernet1/0/7 0 MASTER
TenGigabitEthernet1/0/8 0 FAULTY
TenGigabitEthernet1/0/9 0 FAULTY
TenGigabitEthernet1/0/10 0 FAULTY
TenGigabitEthernet1/0/11 0 MASTER
TenGigabitEthernet1/0/12 0 FAULTY
TenGigabitEthernet1/0/13 0 FAULTY
TenGigabitEthernet1/0/14 0 FAULTY
TenGigabitEthernet1/0/15 0 FAULTY
TenGigabitEthernet1/0/16 0 FAULTY
TenGigabitEthernet1/0/17 0 FAULTY
TenGigabitEthernet1/0/18 0 FAULTY
TenGigabitEthernet1/0/19 0 MASTER
TenGigabitEthernet1/0/20 0 FAULTY
TenGigabitEthernet1/0/21 0 FAULTY
TenGigabitEthernet1/0/22 0 FAULTY
TenGigabitEthernet1/0/23 0 FAULTY
TenGigabitEthernet1/0/24 0 FAULTY
TenGigabitEthernet1/1/1 0 FAULTY
TenGigabitEthernet1/1/2 0 FAULTY
TenGigabitEthernet1/1/3 0 FAULTY
TenGigabitEthernet1/1/4 0 FAULTY
```

```
TenGigabitEthernet1/1/5 0 FAULTY
TenGigabitEthernet1/1/6 0 FAULTY
TenGigabitEthernet1/1/7 0 FAULTY
TenGigabitEthernet1/1/8 0 FAULTY
```

这条命令可以用来查看 ptp 时钟信息。

```
Device#show ptp clock
PTP CLOCK INFO
PTP Device Type: Boundary clock
PTP Device Profile: IEEE 802/1AS Profile
Clock Identity: 0x4:6C:9D:FF:FE:4F:95:0
Clock Domain: 0
Number of PTP ports: 38
PTP Packet priority: 4
Priority1: 128
Priority2: 128
Clock Quality:
Class: 248
Accuracy: Unknown
Offset (log variance): 16640
Offset From Master(ns): 0
Mean Path Delay(ns): 0
Steps Removed: 3
Local clock time: 00:12:13 UTC Jan 1 1970
```

这条命令可以用来查看父时钟信息。

```
Device#show ptp parent
PTP PARENT PROPERTIES
Parent Clock:
Parent Clock Identity: 0xB0:7D:47:FF:FE:9E:B6:80
Parent Port Number: 3
Observed Parent Offset (log variance): 16640
Observed Parent Clock Phase Change Rate: N/A
Grandmaster Clock:
Grandmaster Clock Identity: 0x4:6C:9D:FF:FE:67:3A:80
Grandmaster Clock Quality:
Class: 248
Accuracy: Unknown
Offset (log variance): 16640
Priority1: 0
Priority2: 128
```

这条命令可以用来查看 ptp 端口信息。

```
Device#show ptp port
```

```
PTP PORT DATASET: FortyGigabitEthernet1/1/1
Port identity: clock identity: 0x4:6C:9D:FF:FE:4E:3A:80
Port identity: port number: 1
PTP version: 2
Port state: FAULTY
Delay request interval(log mean): 5
Announce receipt time out: 3
Peer mean path delay(ns): 0
Announce interval(log mean): 1
Sync interval(log mean): 0
Delay Mechanism: End to End
Peer delay request interval(log mean): 0
Sync fault limit: 500000000
PTP PORT DATASET: FortyGigabitEthernet1/1/2
Port identity: clock identity: 0x4:6C:9D:FF:FE:4E:3A:80
Port identity: port number: 2
PTP version: 2
Port state: FAULTY
Delay request interval(log mean): 5
Announce receipt time out: 3
Peer mean path delay(ns): 0
Announce interval(log mean): 1

--More—
```

这条命令可以用来查看一个特定端口的端口信息。

```
Device#show ptp port gi1/0/26
PTP PORT DATASET: GigabitEthernet1/0/26
Port identity: clock identity: 0x4:6C:9D:FF:FE:4E:3A:80
Port identity: port number: 28
PTP version: 2
Port state: MASTER
Delay request interval(log mean): 5
Announce receipt time out: 3
Peer mean path delay(ns): 0
Announce interval(log mean): 1
Sync interval(log mean): 0
Delay Mechanism: Peer to Peer
Peer delay request interval(log mean): 0
Sync fault limit: 500000000
```

这条命令可以用来查看

```
Device#show platform software fed switch active ptp if-id 0x20
Displaying port data for if_id 20
```

```
=====================================
Port Mac Address 04:6C:9D:4E:3A:9A
Port Clock Identity 04:6C:9D:FF:FE:4E:3A:80
Port number 28
PTP Version 2
domain_value 0
dot1as capable: FALSE
sync_recpt_timeout_time_interval 375000000 nanoseconds
sync_interval 125000000 nanoseconds
neighbor_rate_ratio 0.000000
neighbor_prop_delay 0 nanoseconds
compute_neighbor_rate_ratio: TRUE
compute_neighbor_prop_delay: TRUE
port_enabled: TRUE
ptt_port_enabled: TRUE
current_log_pdelay_req_interval 0
pdelay_req_interval 0 nanoseconds
allowed_lost_responses 3
neighbor_prop_delay_threshold 2000 nanoseconds
is_measuring_delay : FALSE
Port state: : MASTER
sync_seq_num 22023
delay_req_seq_num 23857
num sync messages transmitted 0
num sync messages received 0
num followup messages transmitted 0
num followup messages received 0
num pdelay requests transmitted 285695
num pdelay requests received 0
num pdelay responses transmitted 0
num pdelay responses received 0
num pdelay followup responses transmitted 0
num pdelay followup responses received 0
```

MSRP 示例

这个示例显示了查看 MSRP 数据流信息的方法。

```
Device#show msrp streams
-------------------------------------------------------------------------
-----------
Stream ID Talker Listener
Advertise Fail Ready ReadyFail AskFail
R | D R | D R | D R | D R | D
-------------------------------------------------------------------------
```

```
-----------
yy:yy:yy:yy:yy:yy:0001 1 | 2 0 | 0 1 | 0 0 | 1 1 | 0
zz:zz:zz:zz:zz:zz:0002 1 | 0 0 | 1 1 | 0 0 | 0 0 | 1
-----------------------------------------------------------------------------------------------------------------
```

这个示例显示了查看 MSRP 数据流详细信息的方法。

```
Device#show msrp streams detail
Stream ID: 0011.0100.0001:1
Stream Age: 01:57:46 (since Mon Apr 25 23:41:11.413)
Create Time: Mon Apr 25 23:41:11.413
Destination Address: 91E0.F000.FE00
VLAN Identifier: 1
Data Frame Priority: 3 (Class A)
MaxFrameSize: 100
MaxIntervalFrames: 1 frames/125us
Stream Bandwidth: 6400 Kbit/s
Rank: 1
Received Accumulated Latency: 20
Stream Attributes Table:
-------------------------------------------------------------------------
-------
Interface Attr State Direction Type
-------------------------------------------------------------------------
-------
Gi1/0/1 Register Talker Advertise
Attribute Age: 01:57:46 (since Mon Apr 25 23:41:11.413)
MRP Applicant: Very Anxious Observer, send None
MRP Registrar: In
Accumulated Latency: 20
----
Te1/1/1 Declare Talker Advertise
Attribute Age: 00:19:52 (since Tue Apr 26 01:19:05.525)
MRP Applicant: Quiet Active, send None
MRP Registrar: In
Accumulated Latency: 20
----
Te1/1/1 Register Listener Ready
Attribute Age: 00:13:17 (since Tue Apr 26 01:25:40.635)
MRP Applicant: Very Anxious Observer, send None
MRP Registrar: In
----
Gi1/0/1 Declare Listener Ready
Attribute Age: 00:13:17 (since Tue Apr 26 01:25:40.649)
MRP Applicant: Quiet Active, send None
MRP Registrar: In
```

这个示例显示了查看 MSRP 数据流简化信息的方法。

```
Device#show msrp streams brief
Legend: R = Registered, D = Declared.
-------------------------------------------------------------------------
-----------
Stream ID Destination Bandwidth Talkers Listeners
Fail
Address (Kbit/s) R | D R | D
-------------------------------------------------------------------------
-----------
0011.0100.0001:1 91E0.F000.FE00 6400 1 | 1 1 | 1
No
0011.0100.0002:2 91E0.F000.FE01 6400 1 | 1 1 | 1
No
0011.0100.0003:3 91E0.F000.FE02 6400 1 | 1 1 | 1
No
0011.0100.0004:4 91E0.F000.FE03 6400 1 | 1 1 | 1
No
0011.0100.0005:5 91E0.F000.FE04 6400 1 | 1 1 | 1
No
0011.0100.0006:6 91E0.F000.FE05 6400 1 | 1 1 | 1
No
0011.0100.0007:7 91E0.F000.FE06 6400 1 | 1 1 | 1
No
0011.0100.0008:8 91E0.F000.FE07 6400 1 | 1 1 | 1
No
0011.0100.0009:9 91E0.F000.FE08 6400 1 | 1 1 | 1
No
0011.0100.000A:10 91E0.F000.FE09 6400 1 | 1 1 | 1
No
```

这个示例显示了查看 MSR 端口带宽信息的方法。

```
Device#show msrp port bandwidth
-------------------------------------------------------------------------
-----------
Ethernet Capacity Assigned Available Reserved
Interface (Kbit/s) A | B A | B A | B
-------------------------------------------------------------------------
-----------
Te1/0/1 10000000 75 | 0 75 | 75 0 | 0
Te1/0/2 10000000 75 | 0 75 | 75 0 | 0
Te1/0/3 1000000 75 | 0 75 | 75 0 | 0
Te1/0/4 10000000 75 | 0 75 | 75 0 | 0
```

```
Te1/0/5 10000000 75 | 0 75 | 75 0 | 0
Te1/0/6 10000000 75 | 0 75 | 75 0 | 0
Te1/0/8 10000000 75 | 0 75 | 75 0 | 0
Te1/0/9 10000000 75 | 0 75 | 75 0 | 0
Te1/0/10 10000000 75 | 0 75 | 75 0 | 0
Te1/0/11 10000000 75 | 0 75 | 75 0 | 0
Te1/0/12 10000000 75 | 0 75 | 75 0 | 0
Te1/0/13 1000000 75 | 0 75 | 75 0 | 0
Te1/0/14 10000000 75 | 0 75 | 75 0 | 0
Te1/0/15 10000000 75 | 0 75 | 75 0 | 0
Te1/0/16 10000000 75 | 0 75 | 75 0 | 0
Te1/0/17 10000000 75 | 0 75 | 75 0 | 0
Te1/0/18 10000000 75 | 0 75 | 75 0 | 0
Te1/0/19 1000000 75 | 0 75 | 75 0 | 0
Te1/0/20 10000000 75 | 0 75 | 75 0 | 0
Te1/0/21 10000000 75 | 0 75 | 75 0 | 0
Te1/0/22 10000000 75 | 0 75 | 75 0 | 0
Te1/0/23 10000000 75 | 0 75 | 75 0 | 0
Te1/0/24 10000000 75 | 0 75 | 75 0 | 0
Gi1/1/1 1000000 75 | 0 75 | 75 0 | 0
Gi1/1/2 1000000 75 | 0 75 | 75 0 | 0
Gi1/1/3 1000000 75 | 0 75 | 75 0 | 0
Gi1/1/4 1000000 75 | 0 75 | 75 0 | 0
Te1/1/1 10000000 75 | 0 75 | 75 0 | 0
Te1/1/2 10000000 75 | 0 75 | 75 0 | 0
Te1/1/3 10000000 75 | 0 75 | 75 0 | 0
Te1/1/4 10000000 75 | 0 75 | 75 0 | 0
Te1/1/5 10000000 75 | 0 75 | 75 0 | 0
Te1/1/6 10000000 75 | 0 75 | 75 0 | 0
Te1/1/7 10000000 75 | 0 75 | 75 0 | 0
Te1/1/8 10000000 75 | 0 75 | 75 0 | 0
Fo1/1/1 40000000 75 | 0 75 | 75 0 | 0
Fo1/1/2 40000000 75 | 0 75 | 75 0 | 0
```

HQoS 示例

这个示例显示了在启用 AVB 的情况下，查看所有 policy-map 详细配置的方法。

```
Device#show policy-map
Policy Map AVB-Input-Policy-Remark-B
Class AVB-SR-CLASS-A
set cos 3
Class AVB-SR-CLASS-B
set cos 0
Class class-default
```

```
service-policy AVB-Input-Child-Policy
Policy Map AVB-Input-Policy-Remark-A
Class AVB-SR-CLASS-A
set cos 0
Class AVB-SR-CLASS-B
set cos 2
Class class-default
service-policy AVB-Input-Child-Policy
Policy Map AVB-Output-Policy-Default
Class AVB-SR-CLASS-A
priority level 1 1 (%)
Class AVB-SR-CLASS-B
priority level 2 1 (%)
Class AVB-CONTROL-MGMT-QUEUE
priority level 3 15 (%)
Class class-default
bandwidth remaining 100 (%)
queue-buffers ratio 70
service-policy AVB-Output-Child-Policy
Policy Map AVB-Input-Policy-Remark-AB
Class AVB-SR-CLASS-A
set cos 0
Class AVB-SR-CLASS-B
set cos 0
Class class-default
service-policy AVB-Input-Child-Policy
Policy Map AVB-Input-Policy-Remark-None
Class AVB-SR-CLASS-A
set cos 3
Class AVB-SR-CLASS-B
set cos 2
Class class-default
service-policy AVB-Input-Child-Policy
Policy Map AVB-Input-Child-Policy
Class AVB-VOIP-DATA-CLASS
set dscp ef
Class AVB-MULTIMEDIA-CONF-CLASS
set dscp af41
Class AVB-BULK-DATA-CLASS
set dscp af11
Class AVB-TRANSACTIONAL-DATA-CLASS
set dscp af21
Class AVB-SCAVENGER-DATA-CLASS
set dscp cs1
```

```
Class AVB-SIGNALING-CLASS
set dscp cs3
Class class-default
set dscp default
Policy Map AVB-Output-Child-Policy
Class AVB-VOIP-PRIORITY-QUEUE
bandwidth remaining 30 (%)
queue-buffers ratio 30
Class AVB-MULTIMEDIA-CONF-STREAMING-QUEUE
bandwidth remaining 15 (%)
queue-limit dscp af41 percent 80
queue-limit dscp af31 percent 80
queue-limit dscp af42 percent 90
queue-limit dscp af32 percent 90
queue-buffers ratio 15
Class AVB-TRANSACTIONAL-DATA-QUEUE
bandwidth remaining 15 (%)
queue-limit dscp af21 percent 80
queue-limit dscp af22 percent 90
queue-buffers ratio 15
Class AVB-BULK-SCAVENGER-DATA-QUEUE
bandwidth remaining 15 (%)
queue-limit dscp af11 percent 80
queue-limit dscp af12 percent 90
queue-limit dscp cs1 percent 80
queue-buffers ratio 15
Class class-default
bandwidth remaining 25 (%)
queue-buffers ratio 25
```

这个示例显示了在禁用 AVB 的情况下，查看所有 policy-map 详细配置的方法。

```
Device#show policy-map
Building configuration...
Current configuration : 2079 bytes
!
policy-map AVB-Input-Child-Policy
class AVB-VOIP-DATA-CLASS
set dscp ef
class AVB-MULTIMEDIA-CONF-CLASS
set dscp af41
class AVB-BULK-DATA-CLASS
set dscp af11
class AVB-TRANSACTIONAL-DATA-CLASS
set dscp af21
```

```
class AVB-SCAVENGER-DATA-CLASS
set dscp cs1
class AVB-SIGNALING-CLASS
set dscp cs3
class class-default
set dscp default
policy-map AVB-Output-Child-Policy
class AVB-VOIP-PRIORITY-QUEUE
bandwidth remaining percent 30
queue-buffers ratio 30
class AVB-MULTIMEDIA-CONF-STREAMING-QUEUE
bandwidth remaining percent 15
queue-limit dscp af41 percent 80
queue-limit dscp af31 percent 80
queue-limit dscp af42 percent 90
queue-limit dscp af32 percent 90
queue-buffers ratio 15
class AVB-TRANSACTIONAL-DATA-QUEUE
bandwidth remaining percent 15
queue-limit dscp af21 percent 80
queue-limit dscp af22 percent 90
queue-buffers ratio 15
class AVB-BULK-SCAVENGER-DATA-QUEUE
bandwidth remaining percent 15
queue-limit dscp af11 percent 80
queue-limit dscp af12 percent 90
queue-limit dscp cs1 percent 80
queue-buffers ratio 15
class class-default
bandwidth remaining percent 25
queue-buffers ratio 25
!
end
```

这个示例显示了在启用 AVB 的情况下，查看所有 class-map 详细配置的方法。

```
Device#show class-map
Class Map match-any AVB-VOIP-DATA-CLASS (id 31)
Match dscp ef (46)
Match cos 5
Class Map match-any AVB-BULK-DATA-CLASS (id 33)
Match access-group name AVB-BULK-DATA-CLASS-ACL
Class Map match-any AVB-VOIP-PRIORITY-QUEUE (id 37)
Match dscp cs4 (32) cs5 (40) ef (46)
Match precedence 4 5
```

```
Match cos 5
Class Map match-any AVB-MULTIMEDIA-CONF-CLASS (id 32)
Match access-group name AVB-MULTIMEDIA-CONF-CLASS-ACL
Class Map match-any AVB-SIGNALING-CLASS (id 36)
Match access-group name AVB-SIGNALING-CLASS-ACL
Class Map match-any AVB-MULTIMEDIA-CONF-STREAMING-QUEUE (id 38)
Match dscp af41 (34) af42 (36) af43 (38)
Match dscp af31 (26) af32 (28) af33 (30)
Match cos 4
Class Map match-any AVB-BULK-SCAVENGER-DATA-QUEUE (id 40)
Match dscp cs1 (8) af11 (10) af12 (12) af13 (14)
Match precedence 1
Match cos 1
Class Map match-any AVB-TRANSACTIONAL-DATA-CLASS (id 34)
Match access-group name AVB-TRANSACTIONAL-DATA-CLASS-ACL
Class Map match-any AVB-TRANSACTIONAL-DATA-QUEUE (id 39)
Match dscp af21 (18) af22 (20) af23 (22)
Class Map match-any AVB-SR-CLASS-B (id 42)
Match cos 2
Class Map match-any AVB-SR-CLASS-A (id 41)
Match cos 3
Class Map match-any AVB-SCAVENGER-DATA-CLASS (id 35)
Match access-group name AVB-SCAVENGER-DATA-CLASS-ACL
Class Map match-any AVB-CONTROL-MGMT-QUEUE (id 43)
Match ip dscp cs2 (16)
Match ip dscp cs3 (24)
Match ip dscp cs6 (48)
Match ip dscp cs7 (56)
Match ip precedence 6
Match ip precedence 7
Match ip precedence 3
Match ip precedence 2
Match cos 6
Match cos 7
```

这个示例显示了在禁用 AVB 的情况下，查看所有 class-map 详细配置的方法。

```
Device#show class-map
Building configuration...
Current configuration : 2650 bytes
!
class-map match-any AVB-VOIP-DATA-CLASS
match dscp ef
match cos 5
class-map match-any AVB-BULK-DATA-CLASS
```

```
match access-group name AVB-BULK-DATA-CLASS-ACL
class-map match-any AVB-VOIP-PRIORITY-QUEUE
match dscp cs4 cs5 ef
match precedence 4 5
match cos 5
class-map match-any AVB-MULTIMEDIA-CONF-CLASS
match access-group name AVB-MULTIMEDIA-CONF-CLASS-ACL
class-map match-any AVB-SIGNALING-CLASS
match access-group name AVB-SIGNALING-CLASS-ACL
class-map match-any AVB-MULTIMEDIA-CONF-STREAMING-QUEUE
match dscp af41 af42 af43
match dscp af31 af32 af33
match cos 4
class-map match-any AVB-BULK-SCAVENGER-DATA-QUEUE
match dscp cs1 af11 af12 af13
match precedence 1
match cos 1
class-map match-any AVB-TRANSACTIONAL-DATA-CLASS
match access-group name AVB-TRANSACTIONAL-DATA-CLASS-ACL
class-map match-any AVB-TRANSACTIONAL-DATA-QUEUE
match dscp af21 af22 af23
class-map match-any AVB-SCAVENGER-DATA-CLASS
match access-group name AVB-SCAVENGER-DATA-CLASS-ACL
end
```

这个示例显示了查看所有 AVB QoS 统计数据的方法。

```
Device#show policy-map interface gigabitEthernet 1/0/15
GigabitEthernet1/0/15
Service-policy input: AVB-Input-Policy-Remark-AB
Class-map: AVB-SR-CLASS-A (match-any)
0 packets
Match: cos 3
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
cos 0
Class-map: AVB-SR-CLASS-B (match-any)
0 packets
Match: cos 2
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
cos 0
Class-map: class-default (match-any)
```

```
0 packets
Match: any
Service-policy : AVB-Input-Child-Policy
Class-map: AVB-VOIP-DATA-CLASS (match-any)
0 packets
Match: dscp ef (46)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 5
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
cos 3
Class-map: AVB-MULTIMEDIA-CONF-CLASS (match-any)
0 packets
Match: access-group name AVB-MULTIMEDIA-CONF-CLASS-ACL
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp af41
Class-map: AVB-BULK-DATA-CLASS (match-any)
0 packets
Match: access-group name AVB-BULK-DATA-CLASS-ACL
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp af11
Class-map: AVB-TRANSACTIONAL-DATA-CLASS (match-any)
0 packets
Match: access-group name AVB-TRANSACTIONAL-DATA-CLASS-ACL
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp af21
Class-map: AVB-SCAVENGER-DATA-CLASS (match-any)
0 packets
Match: access-group name AVB-SCAVENGER-DATA-CLASS-ACL
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp cs1
Class-map: AVB-SIGNALING-CLASS (match-any)
0 packets
Match: access-group name AVB-SIGNALING-CLASS-ACL
```

```
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp cs3
Class-map: class-default (match-any)
0 packets
Match: any
QoS Set
dscp default
Service-policy output: AVB-Output-Policy-Default
queue stats for all priority classes:
Queueing
priority level 3
(total drops) 0
(bytes output) 7595
queue stats for all priority classes:
Queueing
priority level 2
(total drops) 0
(bytes output) 0
queue stats for all priority classes:
Queueing
priority level 1
(total drops) 0
(bytes output) 0
Class-map: AVB-SR-CLASS-A (match-any)
0 packets
Match: cos 3
0 packets, 0 bytes
5 minute rate 0 bps
Priority: 1% (10000 kbps), burst bytes 250000,
Priority Level: 1
Class-map: AVB-SR-CLASS-B (match-any)
0 packets
Match: cos 2
0 packets, 0 bytes
5 minute rate 0 bps
Priority: 1% (10000 kbps), burst bytes 250000,
Priority Level: 2
Class-map: AVB-CONTROL-MGMT-QUEUE (match-any)
0 packets
Match: ip dscp cs2 (16)
0 packets, 0 bytes
5 minute rate 0 bps
```

```
Match: ip dscp cs3 (24)
0 packets, 0 bytes
5 minute rate 0 bps
Match: ip dscp cs6 (48)
0 packets, 0 bytes
5 minute rate 0 bps
Match: ip dscp cs7 (56)
0 packets, 0 bytes
5 minute rate 0 bps
Match: ip precedence 6
0 packets, 0 bytes
5 minute rate 0 bps
Match: ip precedence 7
0 packets, 0 bytes
5 minute rate 0 bps
Match: ip precedence 3
0 packets, 0 bytes
5 minute rate 0 bps
Match: ip precedence 2
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 6
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 7
0 packets, 0 bytes
5 minute rate 0 bps
Priority: 15% (150000 kbps), burst bytes 3750000,
Priority Level: 3
Class-map: class-default (match-any)
0 packets
Match: any
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 80%
queue-buffers ratio 70
Service-policy : AVB-Output-Child-Policy
Class-map: AVB-VOIP-PRIORITY-QUEUE (match-any)
0 packets
Match: dscp cs4 (32) cs5 (40) ef (46)
0 packets, 0 bytes
5 minute rate 0 bps
Match: precedence 4 5
```

```
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 5
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 30%
queue-buffers ratio 30
Class-map: AVB-MULTIMEDIA-CONF-STREAMING-QUEUE (match-any)
0 packets
Match: dscp af41 (34) af42 (36) af43 (38)
0 packets, 0 bytes
5 minute rate 0 bps
Match: dscp af31 (26) af32 (28) af33 (30)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 4
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
queue-limit dscp 26 percent 80
queue-limit dscp 28 percent 90
queue-limit dscp 34 percent 80
queue-limit dscp 36 percent 90
(total drops) 0
(bytes output) 0
bandwidth remaining 15%
queue-buffers ratio 15
Class-map: AVB-TRANSACTIONAL-DATA-QUEUE (match-any)
0 packets
Match: dscp af21 (18) af22 (20) af23 (22)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 0
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
queue-limit dscp 18 percent 80
queue-limit dscp 20 percent 90
(total drops) 0
(bytes output) 0
bandwidth remaining 15%
```

```
queue-buffers ratio 15
Class-map: AVB-BULK-SCAVENGER-DATA-QUEUE (match-any)
0 packets
Match: dscp cs1 (8) af11 (10) af12 (12) af13 (14)
0 packets, 0 bytes
5 minute rate 0 bps
Match: precedence 1
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 1
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
queue-limit dscp 8 percent 80
queue-limit dscp 10 percent 80
queue-limit dscp 12 percent 90
(total drops) 0
(bytes output) 0
bandwidth remaining 15%
queue-buffers ratio 15
Class-map: class-default (match-any)
0 packets
Match: any
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 25%
queue-buffers ratio 25
```

MVRP 示例

这个示例显示了查看 MVRP 汇总信息的方法。

```
Device#show mvrp summary
MVRP global state : enabled
MVRP VLAN creation : enabled
VLANs created via MVRP : 2,567
MAC learning auto provision : disabled
Learning disabled on VLANs : none
```

这个示例显示了查看接口 MVRP 信息的方法。

```
Device#show mvrp interface
Port Status Registrar State
Te1/0/47 on normal
Te1/1/3 off normal
```

Port Join Timeout Leave Timeout Leaveall Timeout Periodic Timeout
Te1/0/47 20 60 1000 100
Te1/1/3 20 60 1000 100
Port
Te1/0/47
Te1/1/3
Vlans Declared
1-2,567,900
none
Port
Te1/0/47
Te1/1/3
Vlans Registered
2,567
none
Port Vlans Registered and in Spanning Tree Forwarding State
Te1/0/47 2,567
Te1/1/3 none

第 2 部分 园区交换矩阵

园区交换矩

园区交换矩 为通过基于策略的分层结构来搭建虚拟网络提供了基本的基础设施。在这一部分文档中，我们会介绍如何在 Inspur 交换机上配置园区交换矩 。

园区交换矩 概述

园区交换矩 为通过基于策略的分层结构来搭建虚拟网络提供了基本的基础设施。交换矩 覆盖（Fabric Overlay）可以提供诸如主机移动性、增强安全性等服务，这些服务是对常规交换与路由功能的一种补充。

园区交换矩 覆盖功能由三大部分构成：

- 控制平
- 数据平
- 策略平

理解交换矩 域的组成

下图显示了构成交换矩 域的组成成分。

User/Group Repository	用户/组 存储库
Fabric Domain (Overlay)	交换矩 域 （覆盖式）
Fabric Edge Nodes	交换矩 边缘节点
Host DB	主机 DB
Control-Plane Nodes	控制平 节点
Fabric Border Nodes	交换矩 边界节点

- 交换矩 边缘设备：为连接到交换矩 域的用户和设备提供连通性。交换矩 边缘设备会识别并认证端点设备，并在交换矩 主机追踪数据库中注册端点 ID 信息。这类设备会在入站接口对数据进行封装，在出站接口对数据进行解封装，转发往返于矩 所连端点的流 。
- 交换矩 控制平 设备：在交换矩 主机追踪数据库中提供覆盖的可达性信息，以及端

点与路由位置标识（routing locator）的映射。控制平　设备会从交换矩　边缘设备接收到本地端点的注册信息，并且解析从边缘设备发送的请求以寻找远程端点的位置。用户在网络中最多可以配置 3 台控制平　设备，以实现控制平　设备的冗余。

- 交换矩　边界节点：将传统三层网络或者不同的交换矩　域连接到本地域，并将可达性信息和策略信息（如 VRP 和 SGT 信息）从一个域转换到另一个域。
- 虚拟设备（Virtual Context）：使用 VRP（虚拟路由与转发）来创建多个三层路由表实例，以此提供设备级的虚拟化服务。虚拟设备或 VRP 可以达到分离 IP 地址的目的，实现覆盖的地址控制和流　的　离。在交换矩　域中，用户最多可以配置 32 台虚拟设备。
- 主机池：将交换矩　域中的端点分组到 IP 地址池中，用 VLAN ID 和 IP 子网来识别这些端点设备。

园区交换矩　配置指南

- 在配置园区交换矩　成员设备时，应该参考下　的指导方　和　制条件：
- 每个交换矩　域中不要配置超过 3 控制平　的设备；
- 每台交换矩　边缘设备最多支持 2000 台主机；
- 每个控制平　设备最多支持注册 5000 个交换矩　边缘设备；
- 每个交换矩　域中不要配置超过 32 台虚拟设备；

如何配置交换矩　覆盖

配置交换矩　边缘设备

用户可以按照下列步　来配置交换矩　边缘设备：

在开始前

给每台边缘设备配置一个 loopback0 IP 地址，以确保这台设备是可达的。

总步

1. enable

2. configure terminal

3. fabric auto

4. domain {**default** | **name** *fabric domain name*}

5. control-plane *ipv4 address* **auth_key** *key*

6. border *ipv4 address*

7. context name *name* **id** *ID*

8. host-pool name *name*

9. vlan *ID*

10. gateway *IP address/ mask*

11. context name *name*

12. use-dhcp *IP address*

13. exit

14. show fabric domain

具体步

	命令或操作	**目的**
步　1	**enable** 示例：	进入特权 EXEC 模式。在提示时输入密码

	Device> **enable**	
步 2	**configure terminal** 示例： Device# **configure terminal**	进入全局配置模式
步 3	**fabric auto** 示例： Device(config)#**fabric auto**	启用自动交换矩 功能并进入自动交换矩 配置模式
步 4	**domain** {**default** | **name** *fabric domain name*} 示例： Device(config-fabric-auto)#**domain default** Device(config-fabric-auto)#**domain name *exampledomain***	配置 认的交换矩 域并进入域配置模式。**name** 这个关 字的作用是让用户添加新的交换矩 域。如果在这条命令前 添加 **no**，则系统就会删 这个交换矩 域
步 5	**control-plane** *ipv4 address* **auth_key** *key* 示例： Device(config-fabric-auto-domain)#**control-plane *198.51.100.2*** **auth_key *examplekey123***	配置控制平 设备的 IP 地址和认证密 ，其目的是让交换矩 边缘设备能够与控制平 设备进行通信。如果输入命令 **no control-plane** *ipv4 address* **auth_key** *key* 那么设备就会从交换矩域中删 这台控制平 设备
步 6	**border** *ipv4 address* 示例： Device(config-fabric-auto-domain)#**border *198.51.100.4***	配置交换矩 边界设备的 IP 地址，其目的是让交换矩 边缘设备能够与这台交换矩 边界设备进行通信
步 7	**context name** *name* **id** *ID* 示例： Device(config-fabric-auto-domain)#**context name *example-context* id *10***	在交换矩 域中创建一台新的虚拟设备并且给这台设备分配一个 ID
步 8	**host-pool name** *name* 示例： Device(config-fabric-auto-domain)#**host-pool name *VOICE_DOMAIN***	创建一个 IP 地址池，将交换矩 域的端点设备划入，并进入主机池配置模式
步 9	**vlan** *ID*	创建一个 VLAN ID 并且将其与主机池进行关联

	示例： `Device(config-fabric-auto-domain-host-pool)#`**vlan *10***	
步　10	**gateway** *IP address/ mask* 示例： `Device(config-fabric-auto-domain-host-pool)#`**gateway *192.168.1.254/24***	给主机池配置路由网关的 IP 地址和子网掩码。这个地址和子网掩码的会用来将端点标识符映射到 RLOC
步　11	**context name** *name* 示例： `Device(config-fabric-auto-domain-host-pool)#`**context name *example-context***	将一台虚拟设备或者 VRF 与主机池进行关联。在交换矩　域中，用户最多可以配置 32 台虚拟设备
步　12	**use-dhcp** *IP address* 示例： `Device(config-fabric-auto-domain-host-pool)#`**use-dhcp *172.10.1.1***	给主机池配置 DHCP 服务器地址。用户可以给主机池配置多个 DHCP 地址。要想删　DHCP 服务器的地址，可以使用命令 **no use-dhcp** *IP address* 进行删　。如果使用命令 **no use-dhcp**，那么设备就会删　所有的 DHCP 地址
步　13	**exit** 示例： `Device(config-fabric-auto-domain)#` **exit**	
步　14	**show fabric domain** 示例： `Device#` **show fabric domain**	显示交换矩　域的配置。在配置过程中，系统会自动创建出一些其他的 CLI 命令。要想进一步了解相关信息，可以查看交换矩　边缘设备的自动配置命令

配置控制平　设备

要配置控制平　的设备，可以使用下　的 LISP 命令：

在开始前

给每台边缘设备配置一个 loopback0 IP 地址，以确保这台设备是可达的。

总步

1. enable

2. configure terminal

3. router lisp

4. site *site-name*

5. authentication-key *key*

6. eid-prefix [**instance-id** *instance-id*] *eid-prefix* **accept-more-specifics**

7. exit

8. Repeat Step 4 to Step 7 to configure another LISP site.

9. ipv4 map-server

10. ipv4 map-resolver

11. end

具体步

	命令或操作	目的
步 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步 3	**router lisp** 示例： `Device(config)# router lisp`	进入位置标识 ID/分离协议（LISP）配置模式
步 4	**site** *site-name* 示例： `Device(config-router-lisp)# site FD_Default`	在控制平 设备上配置一个 LISP 站点，并进入 LISP 站点配置模式
步 5	**authentication-key** *key* 示例： `Device(config-router-lisp-site)# authentication-key examplekey`	配置用来创建 HMAC（基于散列函数的消息认证代码）SHA-1（安全散列算法）散列值的密码，以便在边缘设备向控制平 设备进行注册时，对其发送的映射注册（map-register）消息进行认证

步 6	**eid-prefix** [**instance-id** *instance-id*] *eid-prefix* **accept-more-specifics**
	
	示例：
	`Device(config-router-lisp-site)# eid-prefix 10.1.0.0/16 accept-more-specifics`
	`Device(config-router-lisp-site)# eid-prefix instance-id 10 10.1.0.0/16 accept-more-specifics`
		配置一个主机池或者端点标识符前缀列表，列表中的前缀都是在边缘设备向控制平 设备进行注册时，可以放在映射注册消息中的前缀。用户可以选择是否将某个 EID 前缀设置得比配置的 EID 更加精确。关 字 **instance-id** 中包含向控制平 设备注册时，指定的实例 ID（即用户希望包含在主机池中的那些虚拟设备所使用的实例 ID）与主机池。

步 7	**exit** 示例：	离开 LISP 站点配置模式并返回 LISP 配置模式

	Device(config-router-lisp-site)# exit	
步 8	重复第 4 步到第 7 步来配置另一个 LISP 站点	
步 9	**ipv4 map-server** 示例： Device(config-router-lisp)# ipv4 map-server	将设备配置为一台 IPv4 控制平 设备
步 10	**ipv4 map-resolver** 示例： Device(config-router-lisp)# ipv4 map-resolver	在交换矩 域中，控制平 设备会充当映射服务器和映射解析器。启用控制平 设备及 IPv4 LSP 映射解析器功能
步 11	**end** 示例： Device(config-router-lisp)# end	离开 LISP 配置模式并返回特权 EXEC 模式

配置边界设备

要配置边缘设备，可以使用下 的 LISP 命令：

在开始前

给每台边缘设备配置一个 loopback0 IP 地址，以确保这台设备是可达的。

总步

1. enable

2. configure terminal

3. router lisp

4. encapsulation vxlan

5. eid-table default instance-id *instance-id*

6. map-cache *eid-prefix ipv4 address/subnet mask* **map-request**

7. ipv4 sgt

8. ipv4 proxy-etr

9. ipv4 proxy-itr *ipv4 address*

10. exit

11. ip route *ipv4-prefix next-hop*

12. exit

具体步

	命令或操作	**目的**
步 1	**enable** 示例： Device> **enable**	进入特权 EXEC 模式。在提示时输入密码
步 2	**configure terminal**	进入全局配置模式

	示例： `Device# configure terminal`	
步 3	**router lisp** 示例： `Device(config)# router lisp`	进入 LISP 配置模式
步 4	**encapsulation vxlan** 示例： `Device(config-router-lisp)# encapsulation vxlan`	设置基于 VXLAN 的封装
步 5	**eid-table default instance-id** *instance-id* 示例： `Device(config-router-lisp)# eid-table default instance-id 0`	将指定的实例 ID 与 认 EID 表进行关联。 控制平 设备的消息中会包含这个实例 ID 及其关联的 EID 前缀
步 6	**map-cache** *eid-prefix ipv4 address/subnet mask* **map-request** 示例： `Device(config-router-lisp)# map-cache 10.1.1.0/24 map-request`	通过添加映射缓存（map-cache）的方式，配置 态 IPv4 的 EID-RLOC 映射关系，其执行的操作是通过发送映射请求（map-request），来请求指定的动态 EID 或主机池
步 7	**ipv4 sgt** 示例： `Device(config-router-lisp)# ipv4 sgt`	在交换矩 中启用对 SGT（安全组标记，Security Group Tags）的传输。要想进一步了解相关信息，可以查看 Inspur TrustSec 配置指南
步 8	**ipv4 proxy-etr** 示例： `Device(config-router-lisp)# ipv4 proxy-etr`	在交换矩 域中启用边界设备服务
步 9	**ipv4 proxy-itr** *ipv4 address* 示例： `Device(config-router-lisp)# ipv4 proxy-itr 10.1.1.1`	通过配置设备，使其充当一台 IPv4 代理入站道路由器（PITR）；同时配置充当数据包封装源地址的那个接口 IP 地址。IPv4 位置标识地址会充当数据包或映射请求消息的源地址
步 10	**exit** 示例：	离开 LISP 站点配置模式并返回全局配置模式

	Device(config-router-lisp)# exit	
步　11	**ip route** *ipv4-prefix next-hop* 示例： Device(config)# ip route 0.0.0.0 0.0.0.0 10.10.10.1	配置一条 IPv4　态路由
步　12	**exit** 示例： Device(config)# exit	离开 LISP 站点配置模式并返回 EXEC 配置模式

交换矩　边缘设备的自动配置命令

一些基于 LISP 的配置、SGT（安全组标记）配置和 EID-RLOC 映射配置也是交换矩　覆盖环境中的一部分，这些配置也会自动创建，并且可以在运行配置中显示出来。

比如，用户可以观察下　这个在边缘设备上所作的配置案例（环回接口的地址为 2.1.1.1/32）：

```
device(config)#fabric auto
device(config-fabric-auto)#domain default
device(config-fabric-auto-domain)#control-plane 192.168.1.4 auth-key example-key1
device(config-fabric-auto-domain)#control-plane 192.168.1.5 auth-key example-key2
device(config-fabric-auto-domain)#border 192.168.1.6
device(config-fabric-auto-domain)#context name example-context ID 10
device(config-fabric-auto-domain)#host-pool name VOICE_DOMAIN
device(config-fabric-auto-domain-host-pool)#vlan 10
device(config-fabric-auto-domain-host-pool)#context example-context
device(config-fabric-auto-domain-host-pool)#gateway 192.168.1.254/24
device(config-fabric-auto-domain-host-pool)#use-dhcp 209.165.201.6
```

这是这台交换矩　边缘配置的输出信息示例：

```
device#show running-config
router lisp
encapsulation vxlan
locator-set default.RLOC
IPv4-interface Loopback0 priority 10 weight 10
exit
!
eid-table default instance-id 0
exit
!
eid-table vrf example-context instance-id 10
dynamic-eid example-context.EID.VOICE_DOMAIN
database-mapping 192.168.1.0/24 locator-set default.RLOC
exit
!
```

```
exit
!
loc-reach-algorithm lsb-reports ignore
disable-ttl-propagate
ipv4 sgt
ipv4 use-petr 192.168.1.6 priority 10 weight 10
ipv4 itr map-resolver 192.168.1.4
ipv4 itr map-resolver 192.168.1.5
ipv4 itr
ipv4 etr map-server 192.168.1.4 key example-key1
ipv4 etr map-server 192.168.1.5 key example-key2
ipv4 etr
exit
!
```

要想进一步了解相关信息，可以查看*配置 LISP* 部分。

示例：交换矩　边缘设备、边界设备与控制平　设备的配置

```
device#show running-config
!
ip vrf example-context
description Auto-provisioned vrf for neighborhood example-context
!
ip dhcp relay information option vpn
ip dhcp relay information option
!
ip dhcp snooping vlan 10
ip dhcp snooping
!
!
fabric auto
!
domain default
control-plane 192.168.1.4 auth-key example-key1
control-plane 192.168.1.5 auth-key example-key2
border 192.168.1.6
context name example-context id 10
!
host-pool name VOICE_DOMAIN
context example-context
vlan 10
gateway 192.168.1.254/24
use-dhcp 209.65.201.6
exit
```

```
exit
exit
!
vlan 10
name VOICE_DOMAIN
!
interface Vlan10
ip vrf forwarding example-context
ip dhcp relay source-interface Loopback0
ip address 192.168.1.254 255.255.255.0
ip helper-address global 209.65.201.6
no ip redirects
ip local-proxy-arp
ip route-cache same-interface
no lisp mobility liveness test
lisp mobility example-context.EID.VOICE_DOMAIN
!
router lisp
encapsulation vxlan
locator-set default.RLOC
IPv4-interface Loopback0 priority 10 weight 10
exit
!
eid-table default instance-id 0
exit
!
eid-table vrf example-context instance-id 10
dynamic-eid example-context.EID.VOICE_DOMAIN
database-mapping 192.168.1.0/24 locator-set default.RLOC
exit
!
exit
!
loc-reach-algorithm lsb-reports ignore
disable-ttl-propagate
ipv4 sgt
ipv4 use-petr 192.168.1.6 priority 10 weight 10
ipv4 itr map-resolver 192.168.1.4
ipv4 itr map-resolver 192.168.1.5
ipv4 itr
ipv4 etr map-server 192.168.1.4 key example-key1
ipv4 etr map-server 192.168.1.5 key example-key2
ipv4 etr
exit
```

```
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Link Status is Control Link Status : Data Link Status
Controllers configured in the Mobility Domain:
IP Public IP Group Name Multicast IP Link Status
------------------------------------------------------------------------------
9.6.136.10 - MG-AK 0.0.0.0 UP : UP
```

：多个接入点检测到了同一台接口设备，但这台设备却将它们显示为了相互独立的 群或者很多可疑设备组成的 群。这是什么原因？

答：接入点必代是这台设备的 RF 邻居，这样这台设备才会将这些接入点检测到的设备视为整体，而接入点 要花一些时一来建立邻居关系。在设备重新启动或者执行过修改 RF 组等事件发生后的几分 ，对于设备 群的认定会出现失准的情况。

：我是否可以用一台设备对两个工作在监控模式（monitor mode）的接入点进行融合？

答：不可以，不能用一台设备对两台工作在监控模式的接入点进行融合，只能使用 MSE 来融合工作在监控模式的接入点。

：如何查看邻居接入点？

答：可以使用这条命令来查看邻居接入点：**show ap** *ap_name* **auto-rf dot11**{**24ghz** | **5ghz**}

这个示例显示了查看邻居接入点的方法。

```
Device#show ap name AS-5508-5-AP3 auto-rf dot11 24ghz
<snippet>
Nearby APs
AP 0C85.259E.C350 slot 0 : -12 dBm on 1 (10.10.0.5)
AP 0C85.25AB.CCA0 slot 0 : -24 dBm on 6 (10.10.0.5)
AP 0C85.25C7.B7A0 slot 0 : -26 dBm on 11 (10.10.0.5)
AP 0C85.25DE.2C10 slot 0 : -24 dBm on 6 (10.10.0.5)
AP 0C85.25DE.C8E0 slot 0 : -14 dBm on 11 (10.10.0.5)
AP 0C85.25DF.3280 slot 0 : -31 dBm on 6 (10.10.0.5)
AP 0CD9.96BA.5600 slot 0 : -44 dBm on 6 (10.0.0.2)
AP 24B6.5734.C570 slot 0 : -48 dBm on 11 (10.0.0.2)
<snippet>
```

：CleanAir 上可以使用哪些调试（debug）命令？

答：CleanAir 上的调试命令包括：

debug cleanair {**all** | **error** | **event** | **internal-event** | **nmsp** | **packet**}

debug rrm {**all** | **channel** | **detail** | **error** | **group** | **ha** | **manager** | **message** | **packet** | **power** | **prealarm** | **profile** | **radar** | **rf-change** | **scale** | **spectrum**}

：为何在出现干扰设备时，没有生成 CleanAir 警报？

答：请检查接入点是否具有 CleanAir 功能，以及接入点和设备上是否都启用了 CleanAir？

：Inspur 6850 和 6650 系列交换机是否可以充当移动代理（MA，Mobility Agent）？

答：是的，Inspur 6850 和 6650 系列交换机都可以充当 MA。

：MA 上是否支持 CleanAir 配置？

答：自 3.3SE 版开始，MA 上即开始支持 CleanAir 配置。用户可以在 MA 上使用下列两条 CleanAir 命令：

- **show ap dot11 5ghz cleanair config**
- **show ap dot11 24ghz cleanair config**

其他参考资料

相关文档

相关主	文档名
CleanAir 命令及具体内容	CleanAir 命令参考手册，Inspur INOS XE 3SE 版（Inspur 6650 交换机）
可用性配置	可用性配置指南，Inspur INOS 11.3.1
可用性命令及具体内容	可用性命令参考手册，Inspur INOS 11.3.1

误消息解码器

描述	接
用户如 搜索和解析这个版本的系统 误消息，可以使用 误消息解码器这 工具	http://www.icntnetworks.com
Inspur 支持（Inspur Support） 可以为用户提供大 在线资源，其中包括排 的文档和工具，以及对 Inspur 产品与技术中若干 的解析。 用户如 获取关于所购产品的安全与技术信息，可以选择订 各类相关服务，譬如产品告警工具（通过最新产品 信息汇总进行访 ）、Inspur 技术服务通讯以及资讯聚合 送（RSS Feeds）。 在 Inspur 支持 中访 大多数工具都要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

第 3 部分 接口与硬件构成

配置接口特征

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于配置接口特征的信息

接口类型

在这一节中，我们会描述设备支持的各类不同接口。在本章后续内容中，我们会介绍物理接口特征的配置流程。

注释： 在支持堆栈的设备上，后面板上的堆栈端口都不是以太网端口，也不能进行配置。

基于端口的 VLAN

一个 VLAN 就是一个忽略用户物理位置，完全根据功能、组别或应用进行逻辑分割而形成的交换型网络。在一个端口接收到的数据包只会转发给处于同一个 VLAN 中的端口。处于不同 VLAN 中的网络设备无法直接进行通信，除非有一台三层设备在 VLAN 之间路由通信流量。VLAN 分割相当于通过 VLAN 给流量中插入了一个真正的防火墙，每个 VLAN 都有自己的 MAC 地址表。当管理员将交换机的一个本地端口关联到一个 VLAN 中时，或者当交换机借助 VTP（VLAN 中继协议）通过干道（trunk）从邻居那里学习到一个 VLAN 时，亦或当用户手动创

建一个 VLAN 时，一个 VLAN 即告生成。VLAN 中可以包含同一堆栈中不同交换机的端口。
要想配置 VLAN，可以使用全局配置模式的命令 **vlan** *vlan-id* 进入 VLAN 配置模式。正常范围的 VLAN（VLAN ID 从 1 到 1005 之间的 VLAN）会被保存到 VLAN 数据库中。如果 VTP 的版本为版本 1 或版本 2，那么要想配置扩展范围的 VLAN（VLAN ID 从 1006 到 4094 之间的 VLAN），那么用户必须首先将 VTP 模式设置为透明。在透明模式中配置的扩展范围 VLAN 不会被添加到 VLAN 数据库中，这些 VLAN 会保存在设备的运行配置当中。如果使用的是 VTP 版本 3，那么用户可以在客户端或服务器模式下创建扩展范围 VLAN，这些扩展范围 VLAN 也会保存在 VLAN 数据库中。
在一个交换机堆栈当中，VLAN 数据库会下载到堆栈中的所有交换机上，堆栈中的所有交换机也会共同组件同一个 VLAN 数据库。对于堆栈中的所有交换机来说，运行配置和保存配置是一样的。
用户在使用接口配置模式命令 **switchport** 将端口添加到一个 VLAN 中时：

- 首先指定接口；
- 对于 trunk 端口，应设置 trunk 特征，如有需要，可以定义这个端口可以传输哪些 VLAN；
- 对于 access 端口，应该定义这个端口属于哪个 VLAN

交换机端口
交换机端口是物理接口集成的纯二层端口。交换机端口可以属于一个或多个 VLAN。交换机端口可以是 access 端口或 turnk 端口。用户可以将端口设置为 access 端口或 trunk 端口，也可以使用 DTP 协议让各个端口分别与链路另一端的端口进行协商，根据协商结果设置这些端口的 switchport 模式。交换机端口可以管理物理接口和对应的二层协议，但是并不能处理路由或桥接功能。
用户可以使用接口配置模式的命令 **switchport** 来配置交换机端口。
Access 端口
access 端口只属于一个 VLAN，也只会携带这个 VLAN 中的流量（除非这个端口被配置为语音 VLAN 端口）。流量在收发时，封装的都是本征（native）格式，也就是交换机不会在流量上打上 VLAN 标记。当 access 端口接收到流量时，它就会认为这个流量属于这个端口所在的 VLAN。如果 access 端口接收到的是打上了标记的数据包（无论是 ISL 标记还是 IEEE 802.1Q 标记），那么它就会丢弃这个数据包，交换机也不会学习这个数据包所携带的源地址。
access 端口支持：

- 手动将静态 access 端口划分给一个 VLAN（或者使用 802.1x 通过 RADIUS 服务器来分配 VLAN）

用户也可以对一个连接 Inspur IP 电话的 access 端口进行配置，让这个端口用一个 VLAN 传输语音流量，用另一个 VLAN 传输与电话相连的设备所发送的数据流量。
Trunk 端口
trunk 端口可以承载多个 VLAN 的流量，而且在默认情况下，trunk 端口是 VLAN 数据库中所有 VLAN 的成员端口。
虽然在默认情况下，trunk 端口是 VTP 所知的所有 VLAN 的成员，但用户可以给每个 trunk 端口可以传输的 VLAN 列表进行配置，来限制 trunk 端口在各个 VLAN 中的成员身份。修改所支持的 VLAN 列表并不会对其他端口构成影响，只与这个 trunk 端口有关。在默认情况下，所有 VLAN（VLAN ID 为 1 到 4094 的 VLAN）都在支持 VLAN 列表当中。但只有当 VTP 学习到一个 VLAN，且该 VLAN 处于启用状态时，trunk 端口才会称为这个 VLAN 的成员端口。如果 VTP 学习到了一个新的、处于启用状态的 VLAN，而这个 VLAN 又在这个 trunk 端口的支持 VLAN 列表当中，那么这个 trunk 端口就会自动成为这个 VLAN 的成员端口，所有通过这个端口往返于该 VLAN 的流量也都会得到转发。如果 VTP 学习到了一个新的、处于启用状态的

VLAN，但这个 VLAN 并不在这个 trunk 端口的支持 VLAN 列表当中，那么这个端口就不会成为该 VLAN 的成员端口，经过这个端口往返与该 VLAN 的流量也不会得到转发。

隧道端口

隧道端口用于 IEEE 802.1Q 隧道技术，其目的是对服务提供商网络中那些使用相同 VLAN 编号的客户进行相互的流量分割。用户可以从服务提供商边缘交换机的隧道端口上配置一条异步链路来连接客户交换机的 IEEE 802.1Q trunk 端口。进入边缘交换机隧道端口的数据包，虽然已经打上了一层客户 VLAN 的 IEEE 802.1Q 标记，但是还会再被封装上一层 IEEE 802.1Q 标记（这个标记称为隧道标记[metro tag]），这个标记中会为每个客户提供一个在服务提供商网络中唯一的 VLAN ID。打上双层标记的数据包在穿越服务提供商网络时既可以保留原始的客户 VLAN，也可以与其他客户的流量进行区分。出站接口同样是隧道端口，在这里交换机会移除隧道标记，露出客户网络打上的原始 VLAN 编号。

隧道端口不能是 trunk 端口或者 access 端口，这类端口必须属于一个与其他客户皆不同的 VLAN。

路由端口

路由端口是一种在操作上类似于路由器端口的物理端口，这类端口未必需要连接到路由器。路由端口不像 access 端口那样需要划分到某个 VLAN 当中。路由端口在操作层面类似于一个普通的路由器接口，但路由端口并不支持 VLAN 子接口。路由端口可以配置三层路由协议。路由端口是纯三层接口，并不支持诸如 DTP 和 STP 这样的二层协议。

用户需要使用接口配置命令 **no switchport** 将接口配置为三层模式，通过这种方法来配置路由端口。接下来，用户可以给这个端口分配 IP 地址、启用路由功能，或者使用全局配置命令 **ip routing** 和 **router** *protocol* 来给端口配置路由协议。

注释： 在输入接口配置模式命令 **no switchport** 之后，接口会先关闭再重新打开，此时接口可能会向直连的设备发送一些消息。如果用户将一个二层接口配置为三层接口，那么之前与这个接口有关的配置信息有可能就会丢失。

软件并没有限制用户可以分配给路由端口哪些编号。不过，由于硬件的限制，这个编号与用户给其他特性配置的编号间的相互关系，有可能会影响 CPU 的性能。

注释： IP Base 镜像支持静态路由和路由信息协议（RIP，Routing Information Protocol）。如果希望支持所有三层路由协议，或者想要回退会桥接端口，用户必须在独立设备或者主用设备上启用 IP Services 镜像。

交换虚拟接口

交换虚拟接口（SVI）是将一个交换端口 VLAN 作为一个接口来使用，在网络系统中发挥路由或桥接的功能。用户可以给一个 VLAN 关联一个 SVI。用户可以通过配置，让一个 VLAN 的 SVI 接口只复杂路由 VLAN 间的流量，或者为设备提供 IP 主机连通性。在默认情况下，系统会为默认 VLAN（VLAN 1）创建一个 SVI，以实现远程设备管理。其他 VLAN 的 SVI 则需要由用户手动进行配置。

注释： VLAN 1 这个接口是无法删除的。

SVI 只会为系统提供 IP 主机连通性。在管理员输入接口配置命令 **vlan** 来创建某个 VLAN 接口时，系统就会针对这个 VLAN 创建出 SVI。在使用 ISL 或 IEEE 802.1Q 封装的 trunk 的链路上，这个 VLAN 会对应数据帧所携带的 VLAN 标记；对于 access 端口，这个 VLAN 则会对应用户配置的 VLAN ID。用户可以给希望路由流量的每个 VLAN 都配置一个 VLAN 接口，然后给这些接口分配 IP 地址。

虽然交换机堆栈或者交换机设备支持最多配置 1005 个 VLAN 和 SVI 接口，但由于硬件的限制，用户配置的 SVI 数量、用户配置的路由端口、以及用户给其他特性配置的编号，这三之间的相互关系会影响 CPU 的性能。

在创建 SVI 时，如果没有关联物理端口，那么这个 SVI 就不会生效。

SVI 自动状态排除

在满足下列条件时：当一个 SVI 所对应的 VLAN 中很多端口，而这个 SVI 的线路状态为 up：

- 设备的 VLAN 数据库中包含这个 VLAN，且这个 VLAN 处于活动（active）状态；
- 设备已经创建了这个 VLAN 接口，且这个接口没有被管理关闭（administratively down）；
- 在这个 VLAN 中，至少有一个二层端口（access 端口或 trunk 端口），且链路处于 up 状态，且该端口在 VLAN 中处于生成树的转发状态。

注释： 当属于这条 VLAN 链路的第 1 个 switchport 启用且进入生成树转发状态时，这个 VLAN 接口的协议链路状态就会进入 up 状态。

当一个 VLAN 中包含很多端口，那么这个 VLAN 默认的操作是，当 VLAN 中的所有端口关闭时，这个 SVI 也会关闭。用户可以在端口上配置 SVI 自动状态排除特性，让设备在执行 SVI 线路状态计算时不将这个端口考虑在内。例如，如果这个 VLAN 中唯一处于活动状态的端口是一个监控端口，用户也许就需要在这个端口上配置自动状态排除特性，以防这个 VLAN 因其他端口状态为 down 而关闭。在端口启用时，**autostate exclude** 这条命令就会应用于这个端口所启用的所有 VLAN。

当 VLAN 中有一个二层端口经历了一段时间实现收敛（即经历了 STP 从侦听-学习状态向转发状态的过渡）之后，VLAN 接口也会随着打开。这是为了防止像路由协议这类的特性按照这些 VLAN 接口处于正常状态的方式使用这些接口，也可以降低出现其他问题（如路由黑洞）的可能性。

EtherChannel 端口组

EtherChannel 端口组可以将多个交换机端口视为一个交换机端口来使用。在设备与设备之间、设备与服务器之间，这些端口组会充当一个逻辑端口，为流量提供高带宽的连接。EtherChannel 会在信道的多条链路之间执行负载分担。如果 EtherChannel 中有一条链路出现了故障，那么这条故障链路之前承载的流量就会改由其他链路来转发。用户可以将多个 trunk 端口打包为一个逻辑 trunk 端口，将多个 access 端口打包为一个逻辑 access 端口，将多个隧道端口打包为一个逻辑隧道端口，或者将多个路由端口打包为一个逻辑路由端口。大多数可以在一个物理端口或者一个这样的汇聚端口上运行的协议，都无法识别出端口组中那些成员物理端口。但也有例外，比如 DTP、思科发现协议（CDP，Cisco Discovery Protocol）、端口汇聚协议（PAgP，Port Aggregation Protocol）这些协议就会只针对物理端口运行。

在用户配置 EtherChannel 时，应该创建一个 port-channel 逻辑接口，然后给 EtherChannel 分配物理接口。对于三层接口来说，用户应使用全局配置模式下的命令 **interface port-channel** 来手动创建逻辑接口，继而使用接口配置模式的命令 **channel-group** 来给 EtherChannel 分配接口。对于二层接口来说，用户可以使用接口配置命令 **channel-group** 来动态创建 port-channel 逻辑接口。这条命令可以实现物理端口与逻辑端口之间的绑定。

多千兆以太网

多千兆以太网（mGig）特性可以让用户在 Inspur 802.11ac Wave2 接入点（AP）的以太网端口上配置超过 1Gbs 的速率。这项技术可以支持 100Mbps、1Gbps、2.5Gbps 和 5Gbps 的速率，这项技术支持通过传统的 5 类线和高速线缆对带宽执行自动协商。在下列交换机上，Inspur 3800 系列接入点支持多千兆以太网：

下面是支持 mGig 特性的 Inspur 交换机型号：

- WS-C6650-8X24PD
- WS-C6650-8X24UQ
- WS-C6650-12X48FD
- WS-C6650-12X48UQ

- WS-C6650-12X48UR
- WS-C6650-12X48UZ

多千兆以太网支持多种速率，端口会首先相互交换一些自动协商信号，来根据信道两边所支持的最高速率建立连接。在高噪声环境中，如果用户在接口上启用了端口速率降档特性，那么当交稿速率的链路无法建立，或者已经建立的链路质量降低到 PHY 需要重新建立链路的地步时，线路速率就会自动降级为一个比较低的速率。下面是推荐使用的降档速率值：

- 10Gbs（降档至 5Gbs）
- 5Gbs（降档至 2.5Gbs）
- 2.5Gbs（降档至 1Gbs）
- 1Gbs（降档至 100Mbs）

以太网端口供电

具备 PoE 功能的交换机端口会自动在下列直连设备发现电路中没有电源时对其供电：

- Inspur 预先定义的用电设备（如 Inspur IP 电话或 Inspur Aironet 接入点）
- 符合 IEEE 802.3af 标准的用电设备

交换机 USB 端口的使用

USB Mini 类型 B Console 端口

设备包含下列类型的 console 端口：

- USB mini-类型 B console 端口；
- RJ-45 console 端口。

Console 的输出信息可以同时显示在连接这两类端口的设备，但 console 每次只能接受其中的一个端口发送输入信息。在默认情况下，USB 端口的优先级高于 RJ-45 端口。

注释： 如果用 Windows PC 连接 USB 端口，需要在 PC 上安装驱动程序。用户可以查看硬件安装指南中的驱动程序安装教程。

用户可以使用一头为 USB 类型 A，另一头为 USB mini 类型 B 的线缆将 PC 或者其他设备与网络设备连接起来。连接的设备上必须安装一个终端模拟应用。当这台设备检测到自己与一台支持主机功能的设备之间建立了有效的 USB 连接时，它就会立刻禁用从 RJ-45 console 端口接收输入信息的做法，转而接收从 USB console 端口接收到的信息。断开 USB 连接后，从 RJ-45 console 连接中接收输入信息的做法也会立刻得到恢复。通过设备的 LED 显示灯可以看出目前哪条 console 连接是生效的。

Console 端口变更日志

在软件启动时，会有一条日志消息显示当前是否有有效的 USB 或 RJ-45 console 连接。堆栈中的每台设备都会发出这样的日志。每台设备都会首先显示 RJ-45 这种媒体类型。

在下面的输出信息示例中，第 1 台设备连接了一条 USB console 线缆。但由于引导加载程序还没有变更为 USB console，所以设备 1 的第 1 条日志消息显示的是 RJ-45 console 连接。过了一段时间之后，console 连接变更为 USB console 的日志信息就显示了出来。而第 2 台设备和第 3 台设备连接的都是 RJ-45 console 线缆。

```
switch-stack-1
*Mar 1 00:01:00.171: %USB_CONSOLE-6-MEDIA_RJ45: Console media-type is RJ45.
*Mar 1 00:01:00.431: %USB_CONSOLE-6-MEDIA_USB: Console media-type is USB.
switch-stack-2
*Mar 1 00:01:09.835: %USB_CONSOLE-6-MEDIA_RJ45: Console media-type is RJ45.
switch-stack-3
```

```
*Mar 1 00:01:10.523: %USB_CONSOLE-6-MEDIA_RJ45: Console media-type is RJ45.
```

当 USB 线缆断开，或者 PC 移除了 USB 连接时，硬件就会自动变更为使用 RJ-45 console 接口：

```
switch-stack-1
Mar 1 00:20:48.635: %USB_CONSOLE-6-MEDIA_RJ45: Console media-type is RJ45.
```

用户可以对 console 类型进行配置让设备永远使用 RJ-45 这种 console 类型，也可以给 USB 连接配置一个静默超时时间。

接口连接

在同一个 VLAN 中的设备可以通过任何交换机实现通信。而不同 VLAN 中的端口则无法在不通过路由设备转发的情况下交换数据。对于标准的二层设备来说，处于不同 VLAN 中的端口需要通过一台路由器才能交换信息。如果给交换机启用路由功能，那么用户可以给 VLAN 20 和 VLAN 30 配置 SVI 接口，并且给它们分配上 IP 地址，这样数据包就可以在不需要外接路由器的情况下直接实现主机 A 到主机 B 的通信了。

图 4：通过交换机连接 VLAN

Layer 3 switch with routing enabled	启用了三层功能的交换机
Host A	主机 A
Host B	主机 B

注释： 运行 LAN Base 镜像的设备只支持给 SVI 接口配置 16 条静态路由。

默认以太网接口配置

如果接口处于三层模式下，而用户又要配置二层参数，那么可以使用接口配置命令 **switchport**（不添加任何参数）来将这个接口设置为二层模式。在输入这条命令之后，接口会先关闭再重新打开，此时接口可能会向其连接的设备发送消息。在用户将三层模式的接口切换为二层模式时，之前对这个接口所作的配置有可能会丢失，这个接口会回到默认配置的状态。

下表显示了以太网接口的默认配置，其中包括一些只应用于二层接口的特性。

表 6：默认二层以太网接口配置

特性	**默认设置**
操作模式	二层或交换模式（即配置 **switchport** 命令）
支持的 VLAN 范围	VLAN 1-4094
（access 端口所在的）默认 VLAN	VLAN1（仅限二层接口）
（IEEE 802.1Q trunk 链路的）本征 VLAN	VLAN1（仅限二层接口）
VLAN 中继（VLAN trunking）	交换端口模式为 dynamic auto（支持 DTP）（基线二层接口）
端口启用状态	所有端口均启用
端口描述	无描述
速率	自动协商（10-Gigabit 接口不支持）
双工模式	自动协商（10-Gigabit 接口不支持）
流量控制	流量控制设置为 **receive: off**。对于发送的数

	据包来说，流量控制始终是关闭的。
EtherChannel（PAgP）	所有以太网端口均关闭
端口阻塞（未知组播与未知单播）	禁用（而非阻塞）（仅限二层接口）
广播、组播与单播风暴控制	禁用
保护端口（protected port）	禁用（仅限二层接口）
端口安全	禁用（仅限二层接口）
PortFast	禁用
auto-MDIX	启用 注释：交换机有可能不支持预先定义的用电设备（如不能完全支持 IEEE 802.3af 的 Inspur IP 电话和接入点），如果该用电设备是通过交叉线与交换机相连。这一点无论交换机端口上是否启用了 auto-MDIX 都是一样的。
以太网供电（PoE）	启用（自动）

接口的速率与双工模式

交换机上的以太网接口可以工作在 10、100、1000 或 10000Mb/s 的速率下，可以工作在全双工和半双工模式。在全双工模式下，两个站点可以同时发送和接收流量。一般来说，10Mb/s 的端口会工作在半双工模式下，这表示站点同时只能接收流量或者发送流量。
交换机型号包括 Gigabit Ethernet（即 10/100/1000-Mb/s）端口，10Gigabit Ehternet 端口和支持 SFP（小型可插拔）模块的 SFP 模块插槽。

速率与双工配置指南

在配置接口速率和双工模式时，应该留意下面的指导方针：

- 10-Gigabit 以太网端口不支持速率而双工特性。这类端口只能工作在 10000Mb/s 这种速率和全双工模式下。
- Gigabit 以太网（10/100/1000-Mb/s）端口支持所有速率选项和所有双工模式选项（自动协商、半双工和全双工）。但当 Gigabit 以太网端口工作在 1000Mb/s 速率下时不支持半双工模式。
- 对于 SFP 模块端口，速率和双工的 CLI 选项会因 SFP 模块的类型不同而变化：
  - 1000BASE-x（其中-x 包括-BX、-CWDM、-LX、-SX 和-ZX）SFP 模块端口支持在接口配置命令 **speed** 后面添加关键字 **nonegotiate**，但不支持双工选项。
  - 1000BASE-T SFP 模块支持的速率和双工配置选项与 10/100/1000-Mb/s 端口相同。
- 如果线路两端都支持自动协商，我们强烈推荐采用 auto 这种协商模式的默认配置。
- 如果一个接口支持自动协商而另一端不支持自动协商，用户就需要在两边的端口上都配置双工和速率，不要在支持的那一边接口上配置 **auto**。
- 如果启用了 STP，那么但端口重新配置时，设备会用最多 30 秒的时间来检查网络中是否有环路。在 STP 重新配置的阶段，端口的 LED 等会显示橙色。

注意： 修改接口的速率和双工模式的配置之后，接口有可能在重新配置的过程中关闭并且再次打开。

IEEE 802.3x 流量控制

流量控制可以让直连的以太网端口在网络出现拥塞的时候对流量的速率进行控制，让出现拥塞的节点暂停另一端的链路操作。如果一个端口因经历拥塞而无法接收到任何流量，那么这个端口就会向另一端的端口发送一个暂停帧，让对方停止发送数据，直至网络条件恢复为止。在接收到暂停帧时，发送方设备会停止发送数据包，这可以防止因链路拥塞而导致丢包。

注释： 交换机端口可以接收暂停帧，但不能发送暂停帧。

用户可以使用接口配置命令 **flow control** 来设置接口 **receive**（接收）暂停帧的方式，可以选择的方式包括 **on**、**off** 或 **desired**。默认的状态为 **off**。

如果设置为 **desired**，那么接口就可以与需要发送流量控制数据包的直连设备或者虽不必需，但有能力发送流量控制数据包的直连设备进行交互。

用户可以参照下列规则在设备上设置流量控制：

- **receive on**（或 **desired**）：该端口无法发送暂停数据帧，但是可以与需要或者能够发送暂停数据帧的设备进行交互；这个端口可以接收暂停数据帧。
- **receive off**：流量控制在双方向都无法实现。如果出现拥塞，那么设备不会向链路对端连接的设备发送指示，双方设备也都不会发送或接收暂停数据帧。

注释： 要了解命令设置的具体信息，以及通过设置在本地和远端端口上实现的流量控制效果，可以查看系统版本命令参考手册中关于接口配置命令 **flow control** 的说明。

三层接口

设备支持下列三层接口：

- SVI：用户应该给需要路由流量的 VLAN 配置 SVI 接口。在管理员输入全局配置命令 **interface vlan** 和 VLAN ID 参数时，系统就会针对这个 VLAN 创建出 SVI。要想删除 SVI，可以在全局配置模式下输入命令 **no interface vlan**，但 VLAN 1 这个接口是无法删除的。

注释： 在创建 SVI 时，如果这个 SVI 没有关联物理接口，那么这个 SVI 是不会生效（active）的。

  在创建 SVI 时，用户也可以在这个 SVI 中的端口上配置 SVI 自动状态排除特性，让系统在计算 SVI 线路状态时不考虑这个端口的状态。
- 路由端口：路由端口是使用接口配置模式命令 **no switchport** 配置为三层模式的那些物理端口。
- 三层 EtherChannel 端口：EtherChannel 接口是由路由端口组成的。

三层设备可以为每个路由端口和 SVI 接口分配一个 IP 地址。

用户可以在一台设备上或者一个设备堆栈中配置多少个 SVI 并没有限制。不过，由于硬件的限制，SVI 的数量、路由端口的数量与用户给其他特性配置的编号间的相互关系，有可能会影响 CPU 的性能。在设备已经达到硬件资源的上限时，如果继续创建路由端口或者 SVI 接口，就会出现下列情况：

- 如果用户尝试创建一个新的路由端口，那么设备就会生成一个消息，显示当前已经没有足够的资源可以将这个接口切换为路由端口，而这个接口也会继续作为交换端口。
- 如果用户想要创建一个扩展范围 VLAN，那么设备就会生成一个错误消息，创建扩展 VLAN 的操作也会被设备拒绝。
- 如果 VLAN 中继协议（VTP）向设备通告了一个新的 VLAN，那么这台设备就会发送一条消息，表示自己已经没有足够的硬件资源，并且关闭那个 VLAN。用户可以通过用户 EXEC

模式下得到命令 **show vlan** 来查看处于中止（suspended）状态下的 VLAN。

- 如果设备尝试启动时，配置文件中的包含的 VLAN 和路由端口数量超出了设备硬件资源可以支持的范围，设备会创建出响应的 VLAN，但路由端口会被设备关闭，设备会发送一条消息表示之所以关闭接口是应该设备的硬件资源不足。

所有三层接口都需要设置 IP 地址才能路由流量。下面我们会演示如何将一个接口设置为三层接口，以及如何给接口分配 IP 地址。

注释： 如果物理端口处于二层模式下（这是默认设置），那么用户就必须输入接口配置模式命令 **no switchport** 将这个接口设置为三层模式。输入 no switchport 这条命令会让接口被禁用并且重新启用，此时设备可能会创建一条消息发送给直连的设备。此外，如果用户将一个二层模式的接口设置为三层模式，那么之前与这个接口有关的配置有可能会被删除，这个接口也会恢复为默认的配置。

如何配置接口特征

配置接口

下面是配置所有接口是都应该参照的一般流程。

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** 示例： `Device(config)#` **interface gigabitethernet1/0/1** `Device(config-if)#`	指定接口类型、设备编号（仅适用于支持堆栈的交换机）和连接器编号。 **注释：** 在接口类型和接口编号之间不需要添加空格。比如，在这一行中，可以输入 **gigabitethernet 1/0/1**、**gigabitethernet1/0/1**、**gi 1/0/1** 或者 **gi1/0/1**。
步骤 4	根据需求给每个接口配置接口配置模式下的命令	定义这个接口上要运行的协议和应用。当用户输入下一跳接口命令，或者输入 **end** 返回特权 EXEC 模式时，之前配置的命令就会被应用在这个接口上
步骤 5	**interface range** 或 **interface rangemacro**	（可选）配置一个范围的接口 **注释：** 要想配置一个范围的接口，这些接口必须类型相同，需要配置的特性和选项也相同。
步骤 6	**show interfaces**	显示交换机上所有接口的列表，或者用户要求显示的接口列表。系统会给设备的每个接口、或者用户指定的那个接口提供一份报告。

为接口添加描述信息

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. description *string*

5. end

6. show interfaces *interface-id* **description**

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet1/0/2**	指定要添加描述信息的接口，并且进入该接口的接口配置模式
步骤 4	**description** *string* 示例： `Device(config-if)#` **description Connects to Marketing**	给接口添加（最多 240 个字符的）描述信息
步骤 5	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 6	**show interfaces** *interface-id* **description**	验证输入的信息
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置接口范围

要给多个接口同时配置相同的参数，可以使用 **interface range** 这条全局配置命令。
在进入到接口范围配置模式之后，用户输入的所有命令参数都会应用到这个范围内的所有接口，直到用户推出该模式为止。

总步骤

1. enable

2. configure terminal

3. interface range {*port-range* | **macro** *macro_name*}

4. end

5. show interfaces [*interface-id*]

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface range** {*port-range* | **macro** *macro_name*} 示例： `Device(config)#` **interface range macro**	指定要配置的接口范围（VLAN 或物理端口），并进入接口范围配置模式。 • 用户可以使用命令 **interface range** 来配置最多 5 个端口范围，或者配置预先定义的宏指令； • 关于 **macro** 这个变量，我们会在接口范围宏指令的配置与使用进行介绍； • 如果用逗号隔开多个端口范围（*port-range*），那么必须给每个条目输入接口类型，并且逗号前后都要留有空格； • 如果用连字符隔开多个端口范围（*port-range*），那么不必重复输入接口类型，但是在连字符前面必须输入一个空格。 **注释：**在接口范围配置模式下，输入普通的配置命令后，这些配置命令就会应用到这个范围内的所有接口。每条命令在输入后，系统就会执行。
步骤 4	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 5	**show interfaces** *interface-id*	验证这个接口范围的配置

	示例： `Device#` **show interfaces**	
步骤 6	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

接口范围宏指令的配置与使用

用户可以创建一个接口范围宏指令，以便在配置时自动选择接口范围。用户在全局配置模式命令 **interface range macro** 中使用 **macro** 这个关键字之前，必须首先使用全局配置命令 **define interface-range** 来定义宏。

总步骤

1. enable

2. configure terminal

3. define interface-range *macro_name interface-range*

4. interface range macro *macro_name*

5. end

6. show running-config | include define

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**define interface-range** *macro_name interface-range* 示例： `Device(config)#` **define interface-range enet_list gigabitethernet1/0/1 - 2**	定义接口范围宏，并且将其保存在 NVRAM 中。 • *macro_name* 是一个最大 32 字符的字符串； • 一个宏中可以包含最多 5 个由逗号分隔的接口范围； • 每个 *interface-range* 中包含的接口必须类型相同。 **注释**：用户在全局配置模式命令 **interface range macro** 中使用 **macro** 这个关键字之前，必须首先使用全局配置命令 **define interface-range** 来定义宏。
步骤 4	**interface range macro** *macro_name*	使用在名为 *macro_name* 的接口范围宏中保存的值，选择要配置的接口范围。

	示例： `Device(config)#` **interface range macro enet_list**	用户可以使用普通的配置命令将配置应用到之前定义的宏所包含的所有接口上
步骤 5	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 6	**show running-config | include define** 示例： `Device#` **show running-config | include define**	显示定义的接口范围宏配置
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置以太网接口

设置接口速率与双工模式参数

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. speed {**10** | **100** | **1000** | **2500** | **5000** | **10000** | **auto** [**10** | **100** | **1000** | **2500** | **5000** | **10000**] | **nonegotiate**}

5. duplex {**auto** | **full** | **half**}

6. end

7. show interfaces *interface-id*

8. copy running-config startup-config

9. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device#` **configure terminal**	
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet1/0/2**	指定要配置的物理接口，并且进入该接口的接口配置模式
步骤 4	**speed** {**10** | **100** | **1000** | **2500** | **5000** | **10000** | **auto** [**10** | **100** | **1000** | **2500** | **5000** | **10000**] | **nonegotiate**} 示例： `Device(config-if)#` **speed 10**	给该接口输入合理的速率参数： • 输入 **10**、**100**、**1000**、**2500**、**5000** 或 **10000**，给这个接口设置速率； • 输入 **auto** 让接口与直连设备自动协商速率。如果指定一个速率，同时还设置了 **auto** 这个关键字，那么这个端口只会就管理员指定的速率范围进行自动协商。 • 只有在 SFP 模块端口上才能使用关键字 **nonegotiate**。SPF 模块端口只能工作在 1000Mb/s 速率，用户可以将其配置为当直连设备不支持自动协商时，即不进行协商
步骤 5	**duplex** {**auto** | **full** | **half**} 示例： `Device(config-if)#` **duplex half**	这条命令无法在 10-Gigabit 以太网接口上使用。 给接口输入双工参数； （针对那些工作在 10 或 100Mb/s 速率的接口）启用半双工模式。用户不能将工作在 1000Mb/s 速率下的接口配置为半双工模式。
步骤 6	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 7	**show interfaces** *interface-id* **description** 示例： `Device#` **show interfaces gigabitethernet1/0/3**	显示接口速率与双工模式的配置
步骤 8	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中
步骤 9	**copy running-config startup-config** 示例： `Device#` **copy running-config**	（可选）将输入的条目保存到配置文件中

	startup-config	

配置多千兆以太网参数

总步骤

1. interface tengigabitethernet *interface number*

2. speed auto

3. downshift-enable

4. end

5. show interfaces downshift

6. show intefaces *interface--number* **downshift**

7. show intefaces downshift module *module-number*

8. show ap name *ap-name* **ethernet statistics**

具体步骤

	命令或操作	目的
步骤 1	**interface tengigabitethernet** *interface number* 示例： `Device(config)#`**interface tengigabitethernet 1/1/37**	配置 10 Gigabit 以太网接口
步骤 2	**speed auto** 示例： `Device(config-if)#`**speed auto**	将速率设置为自动速率协商
步骤 3	**downshift-enable** 示例： `Device(config-if)#`**downshift-enable**	在指定接口上启用降档特性。在启用了降档特性之后，当链路质量不佳或者链路连续断开时，这个接口的速率就会降档至一个较低的速率。
步骤 4	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 5	**show interfaces downshift** 示例： `Device#` **show interfaces downshift**	（可选）显示所有多千兆端口的降档状态
步骤 6	**show intefaces** *interface--number* **downshift**	（可选）显示指定多千兆端口的降档状态

	示例： `Device#` **show interfaces TenGigabitEthernet 1/0/1 downshift**	
步骤 7	**show intefaces downshift module** *module-number* 示例： `Device#` **show interface downshift module 1**	（可选）显示指定模块的降档状态
步骤 8	**show ap name** *ap-name* **ethernet statistics** 示例： `Device#`**show ap name testAP ethernet statistics**	（可选）显示指定 AP 的以太网统计数据

配置 IEEE 802.3x 流量控制

总步骤

1. configure terminal

2. interface *interface-id*

3. flowcontrol {**receive**} {**on** | **off** | **desired**}

4. end

5. show interfaces *interface-id*

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet1/0/1**	指定要配置的物理接口，并且进入该接口的接口配置模式
步骤 3	**flowcontrol** {**receive**} {**on**	**off**	**desired**} 示例： `Device(config-if)#` **flowcontrol receive on**	给端口配置流量控制模式

步骤 4	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 5	**show interfaces** *interface-id* 示例： `Device#` **show interfaces gigabitethernet1/0/1**	显示接口的流量控制设置
步骤 6	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置三层接口

总步骤

1. enable

2. configure terminal

3. interface {**gigabitethernet** *interface-id*} | {**vlan** *vlan-id*} | {**port-channel** *port-channel-number*}

4. no switchport

5. ip address *ip_address subnet_mask*

6. no shutdown

7. end

8. show interfaces [*interface-id*]

9. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** {**gigabitethernet** *interface-id*}	{**vlan** *vlan-id*}	{**port-channel** *port-channel-number*}	指定要配置的三层接口，并且进入该接口的接口配置模式

	示例： `Device(config)#` **interface gigabitethernet1/0/2**	
步骤 4	**no switchport** 示例： `Device(config-if)#` **no switchport**	让物理接口进入三层模式
步骤 5	**ip address** *ip_address subnet_mask* 示例： `Device(config-if)#` **ip address 192.20.135.21 255.255.255.0**	配置 IP 地址和 IP 子网
步骤 6	**no shutdown** 示例： `Device(config-if)#` **no shutdown**	启用接口
步骤 7	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 8	**show interfaces** [*interface-id*] 示例： `Device#` **show interfaces gigabitethernet1/0/3**	查看配置信息
步骤 9	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置逻辑的三层 GRE 隧道接口

在开始前

通用路由封装（GRE）是一种隧道协议，其作用是将网络层协议封装到一个虚拟的点到点链路中。GRE 隧道只提供封装，但不提供加密。

警惕： 从 Inspur INOS XE 3.7.2E 版本开始，Inspur 交换机的硬件开始对 GRE 隧道技术提供支持。如果在配置 GRE 时没有配置隧道可选项，那么设备就会对数据包执行硬件交换。如果

在配置 GRE 时也配置了隧道可选项（如密钥、校验和等等），那么设备就会对数据包执行软件交换。设备支持最多 10 条 GRE 隧道。

注释： GRE 隧道不支持诸如访问控制列表（ACL）和服务质量（QoS）这类特性。

要配置 GRE 隧道，需要执行下面的配置任务：

总步骤

1. interface tunnel *number*

2. ip address*ip_addresssubnet_mask*

3. tunnel source{*ip_address* |*type_number*}

4. tunnel destination{*host_name* | *ip_address*}

5. tunnel mode gre ip

6. end

具体步骤

	命令或操作	目的
步骤 1	**interface tunnel** *number* 示例： `Device(config)#`**interface tunnel 2**	在接口上启用隧道技术
步骤 2	**ip address***ip_addresssubnet_mask* 示例： `Device(config)#`**ip address 100.1.1.1 255.255.255.0**	配置 IP 地址和 IP 子网
步骤 3	**tunnel source**{*ip_address* |*type_number*} 示例： `Device(config)#`**tunnel source 10.10.10.1**	配置隧道源
步骤 4	**tunnel destination**{*host_name*	*ip_address*} 示例： `Device(config)#`**tunnel destination 10.10.10.2**	配置隧道目的
步骤 5	**tunnel mode gre ip** 示例： `Device(config)#`**tunnel mode gre ip**	配置隧道模式
步骤 6	**end** 示例：	离开配置模式

	`Device(config)#`**`end`**	

配置 SVI 自动状态排除

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. switchport autostate exclude

5. end

6. show running config interface *interface-id*

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet1/0/2**	指定要配置的二层接口（物理端口或 port channel），并且进入该接口的接口配置模式
步骤 4	**switchport autostate exclude** 示例： `Device(config-if)#` **switchport autostate exclude**	在定义 SVI 线路状态（为 up 或 down 时）不考虑 access 或 trunk 端口
步骤 5	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 6	**show running config interface** *interface-id*	（可选）显示运行配置 验证配置信息
步骤 7	**copy running-config startup-config** 示例：	（可选）将输入的条目保存到配置文件中

	Device# **copy running-config startup-config**	

关闭接口与重启接口

如果关闭一个接口，那么这个接口上的所有功能也会随之被禁用，设备也会在所有监控命令的显示信息中将这个接口标记为不可用接口。接口关闭的信息会通过所有动态路由协议通告给其他网络服务器。任何路由更新信息中都不会提到这个接口。

总步骤

1. enable

2. configure terminal

3. interface {**vlan** *vlan-id*} | { **gigabitethernet***interface-id*} | {**port-channel** *port-channel-number*}

4. shutdown

5. no shutdown

6. end

7. show running-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： Device> **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： Device# **configure terminal**	进入全局配置模式
步骤 3	**interface** {**vlan** *vlan-id*}	{ **gigabitethernet***interface-id*}	{**port-channel** *port-channel-number*} 示例： Device(config)# **interface gigabitethernet1/0/2**	选择要配置的接口
步骤 4	**shutdown** 示例： Device(config-if)# **shutdown**	关闭接口
步骤 5	**no shutdown** 示例： Device(config-if)# **no**	重新打开接口

	shutdown	
步骤 6	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 7	**show running-config** 示例： `Device#` **show running-config**	查看配置的命令

配置 Console 接口的媒体类型

用户可以按照下面的步骤将 console 的媒体类型设置为 RJ-45。如果将 console 配置为 RJ-45，那么 USB console 的操作状态就会被禁用，设备只会接受通过 RJ-45 console 接口输入的命令。这条配置命令会应用于堆栈中的所有交换机。

总步骤

1. enable

2. configure terminal

3. line console 0

4. media-type rj45

5. end

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**line console 0** 示例： `Device(config)#` **line console 0**	配置 console，进入线路配置模式
步骤 4	**media-type rj45** 示例： `Device(config-line)#` **media-type rj45**	将 console 的媒体类型配置为只支持 RJ-45 端口。如果不输入这条命令，那么当两种类型的端口都有设备连接时，设备默认会接受 USB 端口发起的连接
步骤 5	**end**	返回特权 EXEC 模式

	示例： `Device(config)#` **end**	
步骤 6	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 USB 静默超时时间

用户可以对静默超时时间进行配置，使得在 USB console 端口被激活，但是在一段指定时间之内没有输入操作的情况下，RJ-45 console 端口被重新激活。当 USB console 端口由于超时而失效，用户可以断开再重新连接 USB 线缆，这样连接就会恢复。

注释： 用户配置的静默超时时间会应用于堆栈中的所有设备。不过，一台设备超时并不会导致堆栈中的其他设备也同时超时。

总步骤

1. enable

2. configure terminal

3. line console 0

4. usb-inactivity-timeout *timeout-minutes*

5. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**line console 0** 示例： `Device(config)#` **line console 0**	配置 console，进入线路配置模式
步骤 4	**usb-inactivity-timeout** *timeout-minutes* 示例： `Device(config-line)#` **usb-**	给 console 端口指定一个静默超时时间。时间范围为 1 到 240 分钟。默认时间为无超时时间

	inactivity-timeout 30	
步骤 5	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中

监控接口的特征

监控接口状态

用户可以在特权 EXEC 提示符中输入命令来显示出与接口有关的信息，其中包括软件与硬件的版本、配置命令以及关于接口的统计数据。

表 7：接口的 show 命令

命令	目的
show interfaces *interface-id* **status** [**err-disabled**]	显示接口状态或者处于 error-disabled 状态的接口列表
show interfaces [*interface-id*] **switchport**	显示交换端口（即非路由端口）的管理状态和操作状态。用户可以使用这条命令来查看端口是处于路由模式还是交换模式
show interfaces [*interface-id*] **description**	显示特定接口或者所有接口配置的描述信息，以及接口状态
show ip interface [*interface-id*]	显示所有配置了 IP 路由特性的接口或者某个特定接口的可用性状态
show interface [*interface-id*] **stats**	根据接口交换路径显示这个接口的入站数据包和出站数据包
show interfaces *interface-id*	（可选）显示这个接口的速率和双工模式
show interfaces transceiver dom-supported-list	（可选）显示所连 SFP 模块上的 DOM（数字光学检测）状态
show interfaces transceiver properties	（可选）显示这个接口的温度、电压或总电流
show interfaces [*interface-id*] [{**transceiver properties** | **detail**}] *module number*]	显示关于 SFP 模块的物理状态与操作状态
show running-config interface [*interface-id*]	显示 RAM 中关于这个接口的运行配置
show version	显示硬件配置、软件版本、配置文件的名称与源，以及启动镜像文件
show controllers ethernet-controller *interface-id* **phy**	显示这个接口上 auto-MDIX 的操作状态

接口与计时器的清除与重置

表 8：清除接口的命令

命令	目的
clear counters [*interface-id*]	清除接口计时器
clear interface *interface-id*	重置接口的硬件逻辑
clear line [*number* | **console 0** | **vty** *number*]	重置异步串行线路的硬件逻辑

注释： 特权 EXEC 模式命令 **clear counters** 不会清除通过 SNMP（简单网络管理协议）获得的计时器，这条命令只会清除那些可以通过 **show interface** 命令显示出来的特权 EXEC 命令。

接口特征的配置示例

向接口添加描述信息：示例

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTRL/Z.
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# description Connects to Marketing
Device(config-if)# end
Device# show interfaces gigabitethernet1/0/2 description
Interface Status Protocol Description
Gi1/0/2 admin down down Connects to Marketing
```

显示接口的降档状态：示例

这个示例显示了查看所有多千兆端口降档状态的方法。

```
Device# show interfaces downshift
Port Enabled Active AdminSpeed OperSpeed
Te2/0/37 yes no auto auto
Te2/0/38 yes no auto a-10G
Te2/0/39 yes no auto auto
Te2/0/40 yes no auto a-10G
Te2/0/41 yes no auto auto
Te2/0/42 yes no auto auto
Te2/0/43 yes yes auto a-5000
Te2/0/44 yes no auto auto
Te2/0/45 yes yes auto a-2500
Te2/0/46 yes no auto auto
Te2/0/47 yes no auto a-10G
Te2/0/48 yes no auto auto
```

这个示例显示了查看某个特定多千兆端口降档状态的方法。

```
Device# show interfaces te2/0/43 downshift
Port Enabled Active AdminSpeed OperSpeed
Te2/0/43 yes yes 10G 5000
```

输出信息中各个部分的表意如下表所示：

Port	显示接口的编号
Enabled	显示在这个接口上，降档特性的状态是启用（yes）还是禁用（no）
Active	显示这个接口是否执行了降档
AdminSpeed	显示用户设置的（或）默认的接口速率
OperSpeed	显示这个接口当前的操作速率

配置接口范围：示例

这个示例显示了如何使用全局配置命令 **interface range** 将交换机 1 第 1-4 号端口的速率设置为 100Mb/s：

```
Device# configure terminal
Device(config)# interface range gigabitethernet1/0/1 - 4
Device(config-if-range)# speed 100
```

这个示例显示了如何使用逗号向接口范围中添加不同接口类型串，让 Gigabit 以太网端口 1-3，和 10-Gigabit 以太网 1 和 2 端口，接收流量控制暂停数据帧：

```
Device# configure terminal
Device(config)# interface range gigabitethernet1/0/1 - 3 , tengigabitethernet1/0/1 - 2
Device(config-if-range)# flowcontrol receive on
```

在接口范围模式下输入多条配置命令时，每当用户输入一条命令，这条命令立刻就会执行。这些命令不会按照批处理的形式执行，也不会在用户离开接口范围模式时执行。如果用户在设备正在执行命令时离开接口范围配置模式，那么用户输入的一部分命令可能就不会被应用在这个范围中的所有接口上。所以，在离开接口范围配置模式之前，要等待命令提示符重新出现。

接口范围宏的配置与使用：示例

这个示例显示了如何定义一个名为 *enet_list* 的接口范围，在其中包含交换机 1 上的端口 1 和端口 2，以及如何验证宏的配置。

```
Device# configure terminal
Device(config)# define interface-range enet_list gigabitethernet1/0/1 - 2
Device(config)# end
Device# show running-config | include define
define interface-range enet_list GigabitEthernet1/0/1 - 2
```

这个示例显示了如何创建一个名为 *macro1* 的多接口宏：

```
Device# configure terminal
Device(config)# define interface-range macro1 gigabitethernet1/0/1 - 2, gigabitethernet1/0/5
- 7, tengigabitethernet1/0/1 -2
Device(config)# end
```

这个示例显示了如何进入接口范围宏 *enet_list* 的接口范围配置模式：

```
Device# configure terminal
Device(config)# interface range macro enet_list
Device(config-if-range)#
```

这个示例显示了如何删除接口范围宏 *enet_list*，并且验证这个宏已经被删除：

```
Device# configure terminal
Device(config)# no define interface-range enet_list
Device(config)# end
Device# show run | include define
Device#
```

设置接口速率与双工模式：示例

这个示例显示了如何在一个 10/100/1000Mb/s 端口上，将其接口速率设置为 100Mb/s，同时将其双工模式设置为半双工：

```
Device# configure terminal
Device(config)# interface gigabitethernet1/0/3
Device(config-if)# speed 10
Device(config-if)# duplex half
```

这个示例显示了如何在一个 10/100/1000Mb/s 端口上，将其接口速率设置为 100Mb/s：

```
Device# configure terminal
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# speed 100
```

配置三层接口：示例

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# no switchport
Device(config-if)# ip address 192.20.135.21 255.255.255.0
Device(config-if)# no shutdown
```

配置 Console 媒体类型：示例

在这个示例中，我们禁用了 USB Console 这种媒体类型，启用了 RJ-45 Console 这种媒体类型。

```
Device# configure terminal
Device(config)# line console 0
Device(config-line)# media-type rj45
```

上述配置会导致堆栈中所有当前处于活动状态的 USB Console 媒体类型全部中断。此时，终端上会显示一条日志。这个示例显示了交换机 1 上的 console 媒体类型被转换为了 RJ-45。

```
*Mar 1 00:25:36.860: %USB_CONSOLE-6-CONFIG_DISABLE: Console media-type USB disabled by
```

```
system configuration, media-type reverted to RJ45.
```

此时，堆栈中没有交换机运行用户通过 USB console 端口输入命令。当用 console 线缆与交换机相连时，系统就会显示一条日志消息。如果有 USB console 线缆连接到交换机 2，交换机也会阻止 USB console 输入信息。

```
*Mar 1 00:34:27.498: %USB_CONSOLE-6-CONFIG_DISALLOW: Console media-type USB is
disallowed by system configuration, media-type remains RJ45. (switch-stk-2)
```

在这个示例中，我们对之前的配置进行了逆向操作，因此用户连接的 USB console 线缆马上就被激活了。

```
Device# configure terminal
Device(config)# line console 0
Device(config-line)# no media-type rj45
```

配置 USB 静默超时：示例

下面这个示例将静默超时时间配置为了 30 分钟：

```
Device# configure terminal
Device(config)# line console 0
Device(config-line)# usb-inactivity-timeout 30
```

要想禁用配置，需要使用下列命令：

```
Device# configure terminal
Device(config)# line console 0
Device(config-line)# no usb-inactivity-timeout
```

如果 USB console 端口在用户配置的时长之内没有活动（输入信息），那么用户设置的静默超时时间就会引用到 RJ-45 端口，这时系统会显示一条日志：

```
*Mar 1 00:47:25.625: %USB_CONSOLE-6-INACTIVITY_DISABLE: Console media-type USB
disabled due to inactivity, media-type reverted to RJ45.
```

此时，要想重新激活 USB console 端口，唯一的方法就是断开线缆再重新连接。

当用户断开重新连接与交换机之间的 USB 线缆之后，系统会显示一条类型的日志：

```
*Mar 1 00:48:28.640: %USB_CONSOLE-6-MEDIA_USB: Console media-type is USB.
```

接口特征特性的其他参考资料

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com

标准与 RFC

标准/RFC	标题
无	--

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档	http://www.icntnetworks.com

和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	

配置接口特征的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

配置 Auto-MDIX

Auto-MDIX 的前提条件

如果接口处于三层模式下，而用户又要配置二层参数，那必须输入接口配置命令 **switchport**（不添加任何参数）来将这个接口设置为二层模式。在输入这条命令之后，接口会先关闭再重新打开，此时接口可能会向其连接的设备发送消息。在用户将三层模式的接口切换为二层模式时，之前对这个接口所作的配置有可能会丢失，这个接口会回到默认配置的状态。
下表显示了以太网接口的默认配置，其中包括一些只应用于二层接口的特性。
自动媒体相关接口交叉（auto-MDIX）在默认情况下就会启用。
所有 10/100/1000-Mb/s 和 10/100/1000BASE-TX SFP（小型可插拔）模块接口都可以支持 auto-MDIX 特性，但 1000BASE-SX 或-LX SFP 模块接口则不支持这项特性。

Auto 的限制条件

设备有可能不支持预先定义的用电设备（如不能完全支持 IEEE 802.3af 的 Inspur IP 电话和接入点），如果该用电设备是通过交叉线与交换机相连。这一点无论交换机端口上是否启用了 auto-MDIX 都是一样的。

关于配置 Auto-MDIX 的信息

接口上的 Auto-MDIX

当接口上启用了 auto-MDIX 时，接口就会自动检测相关线缆的连接类型（是直通线还是交叉线），并进行对应的配置。如果连接的设备没有 auto-MDIX 特性，那么用户就必须使用直通线来连接诸如服务器、工作在或路由器，用交叉线来连接其它交换机或者中继器（repeater）。如果启用了 auto-MDIX，那么用户用什么类型的线缆连接其它设备都不受限制，接口会自动纠正错误的线缆类型。要想了解关于线缆需求的详细信息，可以查看硬件安装指南。
下表总结了不同 auto-MDIX 设置与线缆连接方式组合，所对应的链路状态。

表 9：链路条件与 auto-MDIX 的设置

本地端 auto-MDIX	远端端 auto-MDIX	线缆连接正确	线缆连接错误
启用	启用	链路 up	链路 up
启用	禁用	链路 up	链路 up
禁用	启用	链路 up	链路 up
禁用	禁用	链路 up	链路 down

配置 Auto-MDIX

在接口上配置 Auto-MDIX

总步骤

1. enable
2. configure terminal
3. interface *interface-id*
4. speed auto
5. duplex auto
6. end
7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id*	指定要配置的物理接口，并且进入该接口的接口配置模式

	示例： `Device(config)#` **interface gigabitethernet1/0/1**	
步骤 4	**speed auto** 示例： `Device(config-if)#` **speed auto**	配置接口使其与直连设备自动协商速率
步骤 5	**duplex auto** 示例： `Device(config-if)#` **duplex auto**	配置接口使其与直连设备自动协商双工模式
步骤 6	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 Auto-MDIX 的示例

这个示例显示了如何在端口上启用 auto-MDIX：

```
Device# configure terminal
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# speed auto
Device(config-if)# duplex auto
Device(config-if)# mdix auto
Device(config-if)# end
```

接口特征特性的其他参考资料

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用	http://www.icntnetworks.com

户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	

配置接口特征的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

配置以太网管理端口

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

以太网管理端口的前提条件

在将 PC 连接到以太网管理端口时，必须为其分配一个 IP 地址。

关于以太网管理端口的信息

以太网管理端口也称为 Gi0/0 或 GigabitEthernet0/0 端口，这是一种可以用来连接 PC 的 VRF（VPN 路由/转发）接口。用户可以使用以太网管理端口代替设备 console 端口来对设备实施管理。在管理设备堆栈时，用户可以将 PC 机连接到堆栈成员设备的以太网管理端口，对整个堆栈实施管理。

以太网管理端口直接与设备相连

下图显示了如何在独立设备环境中[①]，将以太网管理端口与一台 PC 相连。

① 即不是堆栈环境。——译者注

图 5：将交换机与 PC 相连

Switch	交换机
Ethernet Management port	以太网管理端口
Network ports	网络端口
Network cloud	网络云

使用集线器将以太网管理端口连接到一个设备堆栈

在一个只有堆栈设备的堆栈环境中，每一个堆栈成员的以太网管理端口都会连接到一台与 PC 机相连的集线器上。主用交换机上的以太网管理端口所连接的有效线路会通过集线器连接到 PC。如果主用设备出现了故障，而集群选取出了新的主用设备，那么新主用设备上的管理端口就会与 PC 之间建立有效的链路。

下图显示了 PC 如何通过集线器连接一个设备堆栈。

图 6：将设备堆栈连接到 PC

以太网管理端口与路由转发

以太网管理端口在默认情况下就是启用的。设备不能将以太网管理端口接收到的数据包路由给网络端口，反之亦然。尽管以太网管理端口并不支持路由，但用户有可能需要在这个端口上启用路由协议。

当 PC 有与设备之间相隔多跳，因此数据包必须穿越多台三层设备才能到达 PC 时，我们就需要在以太网管理端口上启用路由协议。

图 7：需要启用路由协议的网络示例

Switch	交换机
Ethernet Management port	以太网管理端口
Network ports	网络端口
Network cloud	网络云
Network cloud	网络云

在上图中，如果以太网管理端口与网络端口都添加到了同一个路由进程中，那么路由条目就会按照下面的方式进行转发：

- 以太网管理端口接收到的路由会通过网络端口转发到网络中。
- 网络端口接收到的路由会通过以太网管理端口转发到网络中。

由于以太网管理端口和网络端口之间是不支持路由转发的，所以要在这两类端口之间往返的流量，这些端口既不会接收，也不会发送。如果设备转发了这类流量，端口之间就会形成数据包环路，影响网络和设备的正常运转。为了防止环路，用户可以配置过滤技术来避免设备为以太网管理端口和网络端口之间转发流量。

以太网管理端口支持的特性

以太网管理端口支持下列特性：

- 快速安装（Express Setup）（仅适用于交换机堆栈）；
- 网络助手（Network Assistant）
- 使用密码认证的 Telnet
- TFTP
- SSH（安全外壳协议）
- 使用 DHCP 协议进行配置
- SNMP（仅 ENTITY-MIB 和 IF-MIB）
- IP ping
- 接口特性：
- 速率：10Mb/s、100Mb/s 及自动协商
- 双工模式：全双工、半双工及自动协商
- 环路检测
- 思科发现协议（CDP）
- DHCP 中继代理
- IPv4 访问控制列表（ACL）
- 路由协议

注意： 在以太网管理端口启用特性之前，请确认该端口支持这种特性。如果在以太网端口上配置了其不支持的特性，这项特性有可能出现工作异常的情况，进而导致设备出现故障。

如何配置以太网管理端口

以太网管理端口的启用与禁用

总步骤

1. configure terminal

2. interface gigabitethernet0/0

3. shutdown

4. no shutdown

5. exit

6. show interfaces gigabitethernet0/0

具体步骤

	命令或操作	目的

步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**interface gigabitethernet0/0** 示例： `Device(config)#` **interface gigabitethernet0/0**	在 CLI 界面中设置以太网管理端口
步骤 3	**shutdown** 示例： `Device(config-if)#` **shutdown**	禁用以太网管理端口
步骤 4	**no shutdown** 示例： `Device(config-if)#` **no shutdow**	启用以太网管理端口
步骤 5	**exit** 示例： `Device(config-if)#` **exi**	离开接口配置模式
步骤 6	**show interfaces gigabitethernet0/0** 示例： `Device#` **show interfaces gigabitethernet0/0**	禁用链路状态。 要想查看 PC 的链路状态，可以观察以太网管理端口的 LED 等。如果 LED 灯为绿色，表示链路状态正常；如果 LED 灯熄灭，表示链路断开；如果 LED 灯为橙色，表示链路中出现了 POST 错误①

① POST 全称为 Quick Power On Self Test，即开机自检

在开始前

要想继续了解通过以太网管理端口管理和配置交换机的方法。可以参考《网络管理配置指南（Inspur 6650 交换机）》

其他参考资料

相关文档

相关主题	文档名
引导加载程序配置	《系统管理配置指南（Inspur 6650 交换机）》
引导加载程序命令	《系统管理配置指南（Inspur 6650 交换机）》

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误	http://www.icntnetworks.com

消息，可以使用错误消息解码器这项工具	

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

关于以太网管理端口的特性信息

版本	修改
Inspur INOS 11.3.1	引入该特性

配置 LLDP、LLDP-MED 及有线位置服务

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

LLDP、LLDP-MED 及有线位置服务概述

LLDP

思科发现协议（CDP）是一种所有 Inspur 制造的设备平台都支持的二层（数据链路层）设备发现协议，包括路由器、网桥、访问服务器、交换机和控制器。CDP 可以让网络管理应用自动发现和学习其它与网络相连的 Inspur 设备。

要想支持非 Inspur 设备，实现与其它设备之间的互操作，这些设备需要支持 IEEE 802.1AB 链路层发现协议（LLDP）。LLDP 是一种网络设备使用的邻居发现协议，这种协议会将关于自己的信息通告给网络中的其它设备。这种协议同样运行在数据链路层上，它可以让运行不同网络层协议的系统之间学习到关于对方的信息。

LLDP 支持的 TLV

LLDP 支持一系列用于发现邻居设备的属性。这些属性包含类型、长度和参数描述，这些属性都称为 TLV。支持 LLDP 的设备可以使用 TLV 来接收来自于其它设备的信息，并且向其它设备发送信息。这种协议可以通告包括诸如配置信息、设备性能和设备身份在内的详细信息。交换机支持下列基本的管理 TLV。这些都是必需的 LLDP TLV。

- 端口描述 TLV
- 系统名 TLV
- 系统描述 TLV
- 系统性能 TLV
- 管理地址 TLV

设备也会通告下列这些特定组织机构定义的 LLDP TLV，以实现对 LLDP-MED 的支持：

- 端口 VLAN ID TLV（这是 IEEE 802.1 组织指定的 TLV）
- MAC/PHY 配置/状态 TLV（IEEE 802.3 组织指定的 TLV）

LLDP 与 Inspur 设备堆栈

一个由多台设备组成的堆栈在网络中相当于一台设备。因此，LLDP 发现的也是设备堆栈，而不是堆栈中的成员设备。

LLDP-MED

媒体端点设备（MED，Media Endpoint Devices）LLDP（LLDP-MED）是 LLDP 的一项扩展协议，这种协议运行在端点设备（如 IP 电话）与网络设备（如交换机）之间。这项协议会专门对 VoIP 应用提供支持，并且为功能发现、网络策略、以太网供电、产品清单管理和位置信息提供额外的 TLV。在默认情况下，所有 LLDP-MED TLV 都是启用的。

LLDP-MED 支持的 TLV

- LLDP-MED 支持下列 TLV：
- LLDP-MED 功能 TLV

  可以让 LLDP-MED 端点判断出直连设备①所支持的功能，及其启用的功能。

① 鉴于 LLDP-MED 运行在端点设备与网络设备之间，而本文档为交换机配置指南。因此，用户应注意在 LLDP-MED 这一部分，凡原文中提到“设备”一词，指的都是交换机。语音设备则一概用“端点”一词表示。——译者注

- 网络策略 TLV

可以让网络设备和端点通告 VLAN 配置，以及该端口上使用的某项应用所对应的二层和三层属性。例如，交换机可以向一台电话通告其应该使用哪个 VLAN ID。这台电话可以与任何设备相连，获取自己的 VLAN ID，然后通过呼叫控制来启动通信。

用户可以通过定义网络策略配置文件（profile）TLV 的方式，给语音和语音信令创建一个配置文件，在其中设置 VLAN 值、服务类型（CoS）、差分服务代码点（DSCP）和标记模式。接下来，这些配置文件属性会由交换机进行集中维护，然后再发送给电话。

- 电源管理 TLV

  可以在 LLEP-MED 端点与网络设备之间启用高级电源管理，让设备和电话能够描述电源信息，如设备的供电方式、电源优先级以及设备需要的电量。

  LLDP-MED 也支持通过一种扩展的电源 TLV 来通告准确的电源需求、端点电源优先级，以及端点和网络设备的电源状态。LLDP 启用时，端口会获得供电，电源 TLV 可以指定端点设备的实际电源需求，让设备可以根据这种需求来为端点设备分配功率。设备会处理请求消息，并且根据当前的功率分配情况来判断是批准还是拒绝自己接收到的请求。如果批准请求，那么交换机就会更新自己的功率分配。如果请求被拒绝，那么设备就会关闭这个端口的供电，生成一个系统日志消息，同时更新自己的功率分配情况。如果禁用了 LLDP-MED 或者端点根本不支持 LLDP-MED，那么在整个连接建立的过程中，设备都会使用最初分配的数值。

  用户可以通过输入接口配置命令 **power inline** {**auto** [**max** *max-wattage*] | **never** | **static** [**max** *max-wattage*]}来修改电源的设置。在默认情况下，PoE 接口的模式为 **auto**。如果用户不指定任何参数，那么设备可以为其分配最大功率（30W）。

- 产品清单管理 TLV

  让端点可以将自己详细的产品清单信息发送给交换机，其中包括硬件修订版本、固件版本、软件版本、序列号、制造商、型号和资产 ID TLV。

- 位置 TLV

  从设备向端点设备提供位置信息。位置 TLV 可以发送下列信息：

  - 公民位置信息

    提供公民地址信息和邮政地址信息。所谓公民位置信息为包括其所在的街道地址、道路名和小区名的邮政信息。

  - ELIN 位置信息

    提供呼叫者的位置信息。这个位置是通过紧急位置标识符（ELIN，Emergency Location Identifier Number）判断出来的，所谓 ELIN 是一个电话号码，可以将紧急呼叫路由到本地公共安全接听点（PSAP），而 PSAP 则可以使用这个电话号码回叫呼叫方。

  - 地址位置信息

    提供关于交换机位置的地址信息，如交换机的经度、纬度和海拔高度。

  - 客户位置

    提供自定义的名称与交换机的位置参数。

有线位置服务

设备可以使用位置服务特性来将直连设备的位置与连接追踪信息发送给 Inspur 移动服务引擎（MSE，Mobility Services Engine）。被追踪设备既可以是有线端点，也可以是有线设备或者控制器。设备会使用网络移动性服务协议（NMSP，Network Mobility Services Protocol）的位置与连接通告，来向 MSE 通告设备链路状态变更事件。

MSE 会向设备发起 NMSP 连接，这会打开一个服务器端口。当 MSE 连接到设备之后，双方

会首先通过一系列的消息交换来建立版本兼容性并交互服务信息，然后它们才会开始同步位置信息。在连接结束之后，设备会周期性地向 MSE 发送位置与连接通告。在一个间隔时间之内发生的一切链路状态变化，都会在这个时间间隔结束之前，以汇总的形式发送出去。

当设备在链路开启或关闭事件中，检测出了链路中的某台设备时，它也就获得了关于这个客户端的很多信息，包括设备的 MAC 地址、IP 地址和用户名。如果客户端支持 LLDP-MED 或者 CDP，那么设备还可以通过 LLDP-MED 位置 TLV 或者通过 CDP 获得这台设备的序列号和 UDI。

根据设备功能的不同，设备可以在链路处于开启状态获时取到下列关于客户端的信息：

- 端口连接中描述的插槽与端口
- 客户端 MAC 地址中描述的 MAC 地址
- 端口连接中描述的 IP 地址
- 802.1x 用户名（如适用）
- 设备分类会被描述为*有线工作站*（*wired station*）
- 状态会被描述为*新*（*new*）
- 序列号、UDI
- 设备型号
- 设备检测到这个关联后经历的时间（单位为秒）

根据设备功能的不同，设备可以在链路处于关闭状态时获取到下列关于客户端的信息：

- 断开连接的插槽与端口
- MAC 地址
- IP 地址
- 802.1x 用户名（如适用）
- 设备分类会被指定为*有线工作站*（*wired station*）
- 状态会被指定为*删除*（*delete*）
- 序列号、UDI
- 设备检测到这个关联断开后经历的时间（单位为秒）

当设备关闭时，它会在关闭与 MSE 的 NMSP 连接之前发送一条连接通告，其中包含 *delete* 这种状态，和 IP 地址。MSE 会认为这个通告表示，所有与这台设备相关的有线客户端都会与其断开关联。

如果用户在这台设备上修改位置地址，那么设备就会发送一条 NMSP 位置通告消息，标识出与此相关的端口以及修改后的地址信息。

默认的 LLDP 配置

表 10：默认的 LLDP 配置

特性	**默认设置**
LLDP 全局状态	禁用
LLDP 保持时间（丢弃前）	120 秒
LLDP 计时器（数据包更新频率）	30 秒
LLDP 重新启动的延迟	2 秒
LLDP tlv-select	禁止发送和接收所有 TLV
LLDP 接口状态	禁用
LLDP 接收	禁用

LLDP 过渡	禁用
LLDP med-tlv-select	禁用发送所有 LLDP-MED TLV。如果在全局启用了 LLDP，那么 LLDP-MED-TLV 也会启用

LLDP 的限制条件

- 如果用户将一个接口配置为了隧道端口，那么 LLDP 就会自动被禁用；
- 如果用户首先在接口上配置了一个网络策略配置文件，那么这个接口上就不能再应用 **switchport voice vlan** 这条命令了。但如果用户已经在接口上配置了 **switchport voice vlan** *vlan-id* 这条命令，那么这个接口上可以应用网络策略配置文件。通过这种方式，用户可以给接口分配一个语音 VLAN 或者语音信令 VLAN，同时在这个接口上应用网络策略配置文件；
- 用户无法在配置了网络策略配置文件的接口上配置静态安全 MAC 地址。

如何配置 LLDP、LLDP-MED 及有线位置服务

启用 LLDP

总步骤

1. enable

2. configure terminal

3. lldp run

4. interface *interface-id*

5. lldp transmit

6. lldp receive

7. end

8. show lldp

9. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**lldp run** 示例： `Device (config)# lldp run`	在设备全局启用 LLDP

步骤 4	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet2/0/1**	指定要启用 LLDP 的物理接口，并且进入该接口的接口配置模式
步骤 5	**lldp transmit** 示例： `Device(config-if)#` **lldp transmit**	启用接口发送 LLDP 数据包的操作
步骤 6	**lldp receive** 示例： `Device(config-if)#` **lldp receive**	启用接口接收 LLDP 数据包的操作
步骤 7	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 8	**show lldp** 示例： `Device#` **show lldp**	验证前面所作的配置
步骤 9	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 LLDP 特征

用户可以对 LLDP 的更新频率、丢弃信息之前保留信息的总时长以及启用 LLDP 的延迟时间。用户还可以选择可以发送和接收哪些 LLDP 和 LLDP-MED TLV。

注释： 从第 2 步到第 5 步不需要按照具体步骤的顺序操作来执行配置。

总步骤

1. **enable**
2. **configure terminal**
3. **lldp holdtime** *seconds*
4. **lldp reinit** *delay*
5. **lldp timer** *rate*
6. **lldp tlv-select**
7. **interface** *interface-id*

8. lldp med-tlv-select

9. end

10. show lldp

11. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**lldp holdtime** *seconds* 示例： `Device(config)#` **lldp holdtime 120**	（可选）指定接收方设备应该丢弃信息之前，在这台设备上保留该信息的总时长。 时间范围是 0 到 65535 秒；默认时长为 120 秒
步骤 4	**lldp reinit** *delay* 示例： `Device(config)#` **lldp reinit 2**	（可选）指定 LLDP 在一个接口上启动时的延迟时间。 时间范围为 2 到 5 秒；默认时长为 2 秒。
步骤 5	**lldp timer** *rate* 示例： `Device(config)#` **lldp timer 30**	（可选）设置 LLDP 更新的发送频率（单位为秒）。 范围为 5 到 65534 秒；默认时间为 30 秒。
步骤 6	**lldp tlv-select** 示例： `Device(config)#` **tlv-select**	（可选）指定可以发送或接收的 LLDP TLV
步骤 7	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet2/0/1**	指定要启用 LLDP 的物理接口，并且进入该接口的接口配置模式
步骤 8	**lldp med-tlv-select** 示例： `Device (config-if)#` **lldp med-tlv-select inventory management**	（可选）指定可以发送或接收的 LLDP-MED TLV

步骤 9	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 10	**show lldp** 示例： `Device#` **show lldp**	验证前面所作的配置
步骤 11	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 LLDP-MED TLV

在默认情况下，设备只会在它从终端设备那里接收到 LLDP-MED 数据包时才会发送 LLDP 数据包。接下来，设备也会发送带有 LLDP-MED 的 LLDP 数据包。当 LLDP-MED 条目过期之后，这台设备会再次回到只发送 LLDP 数据包的操作。

用户可以在接口配置模式下使用命令 **lldp** 让这个接口不要发送下表中的 TLV。

表 11：LLDP-MED-TLV

LLDP-MED-TLV	**描述**
inventory-management	LLDP-MED 产品清单管理 TLV
location	LLDP-MED 位置 TLV
network-policy	LLDP-MED 网络策略 TLV
power-management	LLDP-MED 电源管理 TLV

用户可以按照下面的步骤在接口上启用 TLV

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. lldp med-tlv-select

5. end

6. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device#` **configure terminal**	
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet2/0/1**	指定要启用 LLDP 的物理接口，并且进入该接口的接口配置模式
步骤 4	**lldp med-tlv-select** 示例： `Device(config-if)#` **lldp med-tlv-select inventory management**	指定要启用的 TLV
步骤 5	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 6	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置网络策略 TLV

总步骤

1. enable

2. configure terminal

3. network-policy profile *profile number*

4. {**voice** | **voice-signaling**} **vlan** [*vlan-id* {**cos** *cvalue* | **dscp** *dvalue*}] | [[**dot1p** {**cos** *cvalue* | **dscp** *dvalue*}] | **none** | **untagged**]

5. exit

6. interface *interface-id*

7. network-policy *profile number*

8. lldp med-tlv-select network-policy

9. end

10. show network-policy profile

11. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密码

	示例： `Device>` **enable**	
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**network-policy profile** *profile number* 示例： `Device(config)#` **network-policy profile 1**	指定网络策略配置文件的编号，并且进入网络策略配置模式。编号范围是从 1 到 4294967295。
步骤 4	{**voice** | **voice-signaling**} **vlan** [*vlan-id* {**cos** *cvalue* | **dscp** *dvalue*}] | [[**dot1p** {**cos** *cvalue* | **dscp** *dvalue*}] | **none** | **untagged**] 示例： `Device(config-network-policy)#` **voice vlan 100 cos 4**	配置策略属性： • **voice**：指定语音应用类型； • **voice-signaling**：指定语音信令应用类型； • **vlan**：指定传输语音流量的本征 VLAN； • *vlan-id*：（可选）指定传输语音流量的 VLAN。取值范围是从 1 到 4094； • **cos** *cvalue*：（可选）指定所配置 VLAN 的二层优先级服务类型（CoS）。取值范围是 0 到 7，默认值为 5； • **dscp** *dvalue*：（可选）指定所配置 VLAN 的差分服务代码点（DSCP）值。取值范围是 0 到 63，默认值为 46； • **dot1p**：（可选）让电话使用 IEEE 802.1p 优先级标记并使用 VLAN 0（即本章 VLAN）； • **none**：（可选）不告诉 IP 电话语音 VLAN 的编号。此时电话会使用用户从电话键盘中输入的配置； • **untagged**：（可选）让电话发送未打标的语音流量。这是电话的默认操作方式。
步骤 5	**exit** 示例： `Device(config)#` **exit**	返回全局配置模式
步骤 6	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet2/0/1**	指定要配置网络策略配置文件的物理接口，并且进入该接口的接口配置模式
步骤 7	**network-policy** *profile number* 示例：	指定网络策略配置文件的编号

	`Device(config-if)#` **network-policy 1**	
步骤 8	**lldp med-tlv-select network-policy** 示例： `Device(config-if)#` **lldp med-tlv-select network-policy**	指定网络策略 TLV
步骤 9	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 10	**show network-policy profile** 示例： `Device#` **show network-policy profile**	验证前面所作的配置
步骤 11	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置位置 TLV 和有线位置服务

用户可以从特权 EXEC 模式开始按照下面的步骤来给端点配置位置信息，并且将配置的信息应用到接口上。

总步骤

1. configure terminal

2. location {**admin-tag** *string* | **civic-location identifier** {*id* | **host**} | **elin-location** *string* **identifier** *id* | **custom-location identifier** {*id* | **host**} | **geo-location identifier** {*id* | **host**}}

3. exit

4. interface *interface-id*

5. location {**additional-location-information** *word* | **civic-location-id** {*id* | **host**} | **elin-location-id** *id* | **custom-location-id** {*id* | **host**} | **geo-location-id** {*id* | **host**} }

6. end

7. 使用下列命令：

• **show location admin-tag** *string*

• **show location civic-location identifier** *id*

• **show location elin-location identifier** *id*

8. copy running-config startup-config

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**location** {**admin-tag** *string* | **civic-location identifier** {*id* | **host**} | **elin-location** *string* **identifier** *id* | **custom-location identifier** {*id* | **host**} | **geo-location identifier** {*id* | **host**}} 示例： `Device(config)#` **location civic-location identifier 1** `Device(config-civic)#` **number 3550** `Device(config-civic)#` **primary-road-name "Inspur Way"** `Device(config-civic)#` **city "San Jose"** `Device(config-civic)#` **state CA** `Device(config-civic)#` **building 19** `Device(config-civic)#` **room C6** `Device(config-civic)#` **county "Santa Clara"** `Device(config-civic)#` **country US**	给一个端点设置位置信息： • **admin-tag**：设置管理标记或站点信息； • **civic-location**：设置公民位置信息； • **elin-location**：设置紧急位置信息（ELIN）； • **custom-location**：设置客户位置信息； • **geo-location**：设置地理空间位置信息； • **identifier** *id*：设置公民、ELIN、客户或地理位置的 ID； • **host**：设置主机的公民、客户或地理位置； • *string*：用文字描述的形式置站点或位置信息。
步骤 3	**exit** 示例： `Device(config)#` **exit**	返回全局配置模式
步骤 4	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet2/0/1**	指定要配置位置信息的物理接口，并且进入该接口的接口配置模式
步骤 5	**location** {**additional-location-information** *word* |	在这个接口中输入位置信息： • **additional-location-information**：设置关于位置

	civic-location-id {*id* | **host**} | **elin-location-id** *id* | **custom-location-id** {*id* | **host**} | **geo-location-id** {*id* | **host**} } 示例： `Device(config-if)#` **location elin-location-id 1**	或地点的额外信息 • **civic-location-id**：设置这个接口的全局公民位置信息 • **elin-location-id**：设置这个接口的紧急位置信息 • **custom-location-id**：设置这个接口的客户位置信息 • **geo-location-id**：设置这个接口的地理空间位置信息 • **host**：设置主机的位置标识符 • *word*：设置一段与位置信息有关的文字 • *id*：设置公民、ELIN、客户或地理位置的 ID。ID 取值范围是 1 到 4095。
步骤 6	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 7	使用下面命令： • **show location admin-tag** *string* • **show location civic-location identifier** *id* • **show location elin-location identifier** *id* 示例： `Device#` **show location admin-tag** 或 `Device#` **show location civic-location identifier** 或 `Device#` **show location elin-location identifier**	验证前面所作的配置
步骤 8	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

在设备上启用有线位置服务

在开始前

要想让有线位置服务正常工作，用户必须输入全局配置命令 **ip device tracking**。

总步骤

1. enable

2. configure terminal

3. nmsp notification interval {**attachment** | **location**} *interval-seconds*

4. end

5. show network-policy profile

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**nmsp notification interval** {**attachment** | **location**} *interval-seconds* 示例： `Device(config)#` **nmsp notification interval location 10**	设置 NMSP 的通告间隔。 • **attachment**：设置连接通告间隔； • **location**：设置位置通告间隔； • *interval-seconds*：设置设备向 MSE 发送通告更新或者连接更新之前等待的秒数。范围为从 1 到 30 秒；默认为 30 秒。
步骤 4	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 5	**show network-policy profile** 示例： `Device#` **show network-policy profile**	验证前面所作的配置
步骤 6	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

LLDP、LLDP-MED 与有线位置服务的配置示例

配置网络策略 TLV：示例

这个示例显示了如何通过配置让携带 CoS 的语音应用在 VLAN 100 中传输，同时在接口上启用网络策略配置文件和网络策略 TLV：

```
# configure terminal
(config)# network-policy 1
(config-network-policy)# voice vlan 100 cos 4
(config-network-policy)# exit
(config)# interface gigabitethernet1/0/1
(config-if)# network-policy profile 1
(config-if)# lldp med-tlv-select network-policy
```

这个示例显示了如何让携带优先级标记的语音应用通过本征 VLAN 进行传输：

```
config-network-policy)# voice vlan dot1p cos 4
config-network-policy)# voice vlan dot1p dscp 34
```

LLDP、LLDP-MED 和有线位置服务的监控与维护

监控与维护 LLDP、LLDP-MED 和有线位置服务的命令可以参考下表：

命令	描述
clear lldp counters	将流量计数器重置为 0
clear lldp table	删除 LLDP 邻居信息表
clear nmsp statistics	清除 NMSP 统计数据计数器
show lldp	显示全局信息，如传输频率、被发送数据包保存时间、LLDP 在一个接口上启动的延迟时间
show lldp entry *entry-name*	显示关于特定邻居的信息。 用户可以输入星号（*）来显示所有邻居，也可以输入具体的邻居名
show lldp interface [*interface-id*]	显示与启用了 LLDP 的接口有关的信息。 可以让系统仅仅显示某个接口的信息
show lldp neighbors [*interface-id*] [**detail**]	显示关于邻居的信息，其中包括设备类型、接口类型与编号、保存时间的设置、功能与端口 ID。 可以让系统仅显示某个接口的邻居，也可以让系统显示更加具体的信息
show lldp traffic	显示 LLDP 计时器，包括收发的数据包数量、丢弃的数据包数量以及未识别的 TLV 数量
show location admin-tag *string*	显示特定管理标记或站点的位置信息
show location civic-location identifier *id*	显示一个特定全局公民位置的位置信息

show location elin-location identifier *id*	显示一个紧急位置的位置信息
show network-policy profile	显示用户配置的网络策略配置文件
show nmsp	显示 NMSP 信息

其他关于 LLDP、LLDP-MED 与有线位置服务的参考资料

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

关于 LLDP、LLDP-MED 与有线位置服务的特性信息

版本	修改
Inspur INOS 11.3.1	引入该特性

配置系统 MTU

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于 MTU 的信息

所有设备接口默认收发的数据帧最大传输单元（MTU）为 1500 字节。

系统 MTU 的限制条件

用户在配置系统 MTU 值时，可以参考下面的指导方针：

- 设备不支持给不同接口分别配置 MTU；
- 如果用户在全局配置模式下输入命令 **system mtu** *bytes*，这条命令并不会在设备上生效。这条命令只会作用于交换机快速以太网端口的系统 MTU 设置。

系统 MTU 值的应用

在一个交换机堆栈中，应用于成员交换机的 MTU 值取决于堆栈的配置。用户可以进行下面这些堆栈的配置：
根据交换机或交换机堆栈的配置，并参考当前应用的系统 MTU 或系统巨型 MTU 值，来设置 IP 或 IPv6 MTU 值的上限。要了解更多关于设置 MTU 值的信息，可以查看这个版本命令参考手册中的全局配置命令 **system mtu**。

如何配置 MTU 值

配置系统 MTU

用户可以按照下列步骤来修改交换与路由数据包的 MTU 值：
总步骤
1. enable
2. configure terminal
3. system mtu *bytes*
4. end

5. copy running-config startup-config

6. reload

7. show system mtu

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**system mtu** *bytes* 示例： `Device(config)#` **system mtu 1900**	（可选）给所有 GigabitEthernet 和 10-GigabitEthernet 接口修改 MTU 值
步骤 4	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 5	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中
步骤 6	**reload** 示例： `Device#` **reload**	重启操作系统
步骤 7	**show system mtu** 示例： `Device#` **show system mtu**	验证前面所作的设置

配置特定协议的 MTU

用户可以从特权 EXEC 模式，按照下列步骤来修改路由端口的 MTU 值：

总步骤

1. configure terminal

2. interface *interface*

3. ip mtu *bytes*

4. ipv6 mtu *bytes*

5. end

6. copy running-config startup-config

7. reload

8. show system mtu

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *interface* 示例： `Device(config)#` **interface gigabitethernet0/0**	进入接口配置模式
步骤 3	**ip mtu** *bytes* 示例： `Device(config-if)#` **ip mtu 68**	修改 IPv4 MTU 值
步骤 4	**ipv6 mtu** *bytes* 示例： `Device(config-if)#` **ipv6 mtu 1280**	（可选）修改 IPv6 MTU 值
步骤 5	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 6	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中
步骤 7	**reload** 示例： `Device#` **reload**	重启操作系统
步骤 8	**show system mtu** 示例：	验证前面所作的设置

	Device# **show system mtu**	

系统 MTU 的配置示例

这个示例显示了如何将 GigabitEthernet 端口的最大数据包设置为 7500 字节：

```
Device(config)# system mtu 7500system mtu 1900
Device(config)#
Device(config)# exit
```

如果用户输入的数值超出了这类接口许可的范围，设备就不会接受这条命令。这个示例显示了当用户尝试给 GigabitEthernet 端口设置一个超出范围的参数时，系统作出的响应：

```
Device(config)# system mtu 25000
^
% Invalid input detected at '^' marker.
```

这个示例显示了命令 **show system mtu** 的输出信息：

```
Device# show system mtu
Global Ethernet MTU is 1500 bytes.
```

其他关于系统 MTU 的参考资料

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

关于系统 MTU 的特性信息

版本	修改
Inspur INOS XE 3.3SE	引入该特性

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于 MTU 的信息

所有设备接口默认收发的数据帧最大传输单元（MTU）为 1500 字节。

系统 MTU 的限制条件

用户在配置系统 MTU 值时，可以参考下面的指导方针：

- 设备不支持给不同接口分别配置 MTU；
- 如果用户在全局配置模式下输入命令 **system mtu** *bytes*，这条命令并不会在设备上生效。这条命令只会作用于交换机快速以太网端口的系统 MTU 设置。

系统 MTU 值的应用

在一个交换机堆栈中，应用于成员交换机的 MTU 值取决于堆栈的配置。用户可以进行下面这些堆栈的配置：
根据交换机或交换机堆栈的配置，并参考当前应用的系统 MTU 或系统巨型 MTU 值，来设置 IP 或 IPv6 MTU 值的上限。要了解更多关于设置 MTU 值的信息，可以查看这个版本命令参考手册中的全局配置命令 **system mtu**。

如何配置 MTU 值

配置系统 MTU

用户可以按照下列步骤来修改交换与路由数据包的 MTU 值：
总步骤
1. enable
2. configure terminal
3. system mtu *bytes*
4. end

5. copy running-config startup-config

6. reload

7. show system mtu

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**system mtu** *bytes* 示例： `Device(config)#` **system mtu 1900**	（可选）给所有 GigabitEthernet 和 10-GigabitEthernet 接口修改 MTU 值
步骤 4	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 5	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中
步骤 6	**reload** 示例： `Device#` **reload**	重启操作系统
步骤 7	**show system mtu** 示例： `Device#` **show system mtu**	验证前面所作的设置

配置特定协议的 MTU

用户可以从特权 EXEC 模式，按照下列步骤来修改路由端口的 MTU 值：

总步骤

1. configure terminal

2. interface *interface*

3. ip mtu *bytes*

4. ipv6 mtu *bytes*

5. end

6. copy running-config startup-config

7. reload

8. show system mtu

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *interface* 示例： `Device(config)#` **interface gigabitethernet0/0**	进入接口配置模式
步骤 3	**ip mtu** *bytes* 示例： `Device(config-if)#` **ip mtu 68**	修改 IPv4 MTU 值
步骤 4	**ipv6 mtu** *bytes* 示例： `Device(config-if)#` **ipv6 mtu 1280**	（可选）修改 IPv6 MTU 值
步骤 5	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 6	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中
步骤 7	**reload** 示例： `Device#` **reload**	重启操作系统
步骤 8	**show system mtu** 示例：	验证前面所作的设置

	Device# **show system mtu**	

系统 MTU 的配置示例

这个示例显示了如何将 GigabitEthernet 端口的最大数据包设置为 7500 字节：

```
Device(config)# system mtu 7500system mtu 1900
Device(config)#
Device(config)# exit
```

如果用户输入的数值超出了这类接口许可的范围，设备就不会接受这条命令。这个示例显示了当用户尝试给 GigabitEthernet 端口设置一个超出范围的参数时，系统作出的响应：

```
Device(config)# system mtu 25000
^
% Invalid input detected at '^' marker.
```

这个示例显示了命令 **show system mtu** 的输出信息：

```
Device# show system mtu
Global Ethernet MTU is 1500 bytes.
```

其他关于系统 MTU 的参考资料

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

关于系统 MTU 的特性信息

版本	修改
Inspur INOS 11.3.1	引入该特性

配置内部电源

关于内部电源的信息

用户可以阅读设备安装指南，来了解关于电源的信息

如何配置内部电源

配置内部电源

用户可以使用 EXEC 命令 **power supply** 开对设备上的内部电源进行配置和管理。设备不支持 **no power supply** 这条 EXEC 命令。

总步骤

1. power supply *switch_number* **slot**{**A** | **B**} { **off** | **on** }

2. show environment power

具体步骤

	命令或操作	目的
步骤 1	**power supply** *switch_number* **slot**{**A** | **B**} { **off** | **on** } 示例： `Device#` **power supply 1 slot A on**	通过下面的关键字来将所选电源设置为 off 或 on： • **A**：选择槽位 A 的电源； • **B**：选择槽位 B 的电源； **注释**：槽位 B 的电源与设备外缘距离最近。 • **off**：将电源设置为关闭； • **on**：将电源设置为开启。 设备电源的默认状态为 **on**。
步骤 2	**show environment power** 示例： `Device#` **show environment power**	验证前面所作的设置

内部电源的监控

表 12：查看电源情况的 show 命令

命令	目的
show environment power [**all** | **switch** *switch_number*]	（可选）显示堆栈中每台设备或者某台设备的内部供电状态。取值范围为 1 到 9，具体取值取决于堆栈中的设备成员编号。 只有在有堆栈功能的设备上才可以使用设

	备关键字（**switch**）。

内部电源的配置示例

这个示例显示了如何将槽位 A 中的电源设置为关闭：

```
Device# power supply 1 slot A off
Disabling Power supply A may result in a power loss to PoE devices and/or switches ...
Continue? (yes/[no]): yes
Device#
Jun 10 04:52:54.389: %PLATFORM_ENV-6-FRU_PS_OIR: FRU Power Supply 1 powered off
Jun 10 04:52:56.717: %PLATFORM_ENV-1-FAN_NOT_PRESENT: Fan is not present
Device#
```

这个示例显示了如何将槽位 A 中的电源设置为打开：

```
Device# power supply 1 slot A on
Jun 10 04:54:39.600: %PLATFORM_ENV-6-FRU_PS_OIR: FRU Power Supply 1 powered on
```

这个示例显示了命令 **show env power** 的输出信息：

```
Device# show env power
SW PID Serial# Status Sys Pwr PoE Pwr Watts
--- ------------------
1A PWR-C2-640WAC
1B Not Present
---------- ---------------
DCB1705B05B OK
-------
Good
-------
Good
-----
640
Device#
```

表 13：命令 **show env power** 的状态描述

列	描述
OK	有电源且电源工作正常
Not Present	这个槽位没有安装电源
No Input Power	槽位安装了电源但电源没有供电
Disabled	有电源、电源也有供电，但电源被 CLI 命令关闭
Not Responding	电源无法识别或故障
Failure-Fan	电源风扇故障

其他关于系统 MTU 的参考资料

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com

配置 Inspure 可扩展电源系统（XPS）2200

这部分文档包含下面几部分内容：
配置 XPS 2200 的限制条件
关于配置 XPS 2200 的信息
如何配置 XPS 2200
XPS 2200 的监控与维护
其他参考资料

配置 XPS 2200 的限制条件

- 当用户在使用 RPS 模式的 XPS 供电来为交换机电源提供备份时，XPS 中的最小供电必须大于 RPS 模式下 XPS 端口所连交换机的最大供电；
- 在 RPS 模式下，每个 XPS 电源都可以为一个，也只能为一个交换机电源提供分配，无论其是大是小；
- 如果用户要从电源堆叠（交换机或 XPS）中删除电源，那么请务必确保移除这个电源不会因为可用电源不足而导致低压减载（load shedding）。

关于配置 XPS 2200 的信息

Inspur 可扩展电源系统（XPS）2200 概述

Inspur 可扩展电源系统（XPS）2200 是一种独立的电源系统，用户可以将这个系统连接到 Inspur 交换机。XPS 2200 可以在所连设备的电源出现故障时为设备提供备份电源，也可以在交换机电源堆叠中为电源堆叠提供额外的供电。
XPS 2200 电源端口和内部电源可以工作在冗余电源（RPS）模式或者堆叠电源（SP）模式下。堆叠电源模式只能用于有堆栈功能的交换机的电源堆叠当中。如果没有使用 XPS，那么电源堆叠会采用环形拓扑的方式来操作，堆栈中最大可以部署 4 台交换机。如果将两个堆栈融合在一起，融合后的交换机总数也不能超过 4 台。如果在电源堆叠中应用了 XPS，那么堆栈加 XPS 最大可以连接 9 台交换机，它们可以采取一种类似于堆叠电源环形拓扑的操作方式来为

电源堆叠中的成员提供供电。
所有在 SP 端口与 XPS 相连的 Inspur 交换机都是同一个电源堆叠的成员，所有来自于 XPS 和交换机的电能都会由堆栈中的所有交换机共享。电源共享是默认的模式，但 XPS 也支持环形拓扑中所采用的那种堆叠电源模式（严格与非严格电源共享或冗余模式）。
在使用两个电源时，系统可以工作在混合模式下，其中一个电源工作在 RPS 模式下，另一个电源则工作在 SP 模式下。用户可以按照自己希望使用 XPS 2200 的方式来对端口和电源进行配置。
XPS 2000 有 9 个电源端口，这些端口可以充当 RPS 角色，也可以充当自动堆叠电源角色（此为默认模式），操作模式是由连接端口的交换机类型所决定的。用户也可以使用 CLI 来强制可堆叠交换机工作在 RPS 模式下。

- 当一台运行 LAN base 镜像的 Inspur 交换机，或者一台 Inspur（不可堆叠）交换机连接到一个端口时，其模式为 RPS 模式，而 XPS 2200 会在交换机电源出现故障时充当备份电源；
- 当一台运行 IP base 或 IP services 许可证的 Inspur（可堆叠）交换机连接到一个端口时，其模式为 SP，此时这台交换机会成为堆叠电源系统中的一部分。

用户可以在每一台与电源端口相连的交换机上对 XPS 进行配置。用户可以使用任何 XPS 端口来进行配置，也可以从任意一台与 XPS 相连的交换机上对任意端口进行配置。如果用户进入了多台交换机的配置模式，那么最终生效的配置是用户最后应用的配置。
虽然用户可以通过一台交换机完成所有的 XPS 配置，但 XPS 2200 也运行有自己的软件。用户可以通过 XPS 服务端口来对这个软件进行升级。

XPS 2200 电源模式

XPS 有两个电源可以运行在 RPS 模式或 SP 模式下。
在 SP 模式下，XPS 上的所有 SP 端口都属于同一个电源堆叠。当电源堆叠中包含 XPS 时，堆叠拓扑就是一个星型拓扑，其中包含最多 9 台成员设备再加上 XPS 2200。工作在 SP 模式下的 XPS 电源可以理解为是在进行功率分配。如果两个 XPS 电源同时处于 RPS 模式，那么电源堆叠就会完全由那些与 SP 模式 XPS 端口相连的交换机组成，而功率分配方式是由这些交换机上的电源来决定的。
如果电源角色不匹配，例如有一个 XPS 端口被配置为了 RPS 模式，但两个电源都工作在 SP 模式下，那么 XPS 就会检测到这个模式不匹配的情况，因此它就会发送错误消息。

RPS 模式

如果两个 XPS 电源都处于 RPS 模式下，那么 RPS 就可以给同等数量或者数量较少的交换机电源提供电源备份。XPS 中最小的电源必须大于 RPS 模式下 XPS 端口所连交换机中的最大电源。
如果只有一个电源处于 RPS 模式，那么 XPS 就只能给一个电源提供备份，即使故障电源的功率要小得多。例如，如果有一个 XPS 1100W 电源处于 RPS 模式，同时有两个 350W 交换机电源出现了故障，XPS 还是只能给其中的一个交换机电源提供备份。
当一个处于 RPS 模式下的 XPS 电源正在为一个交换机电源提供备份时，另一个交换机电源也发生了故障，那么系统就会弹出一个消息，显示 XPS 备份不可用。当故障电源重新启动时，XPS 就能够给另一个电源提供备份了。
如果 XPS 正在为一台交换机上的两个故障电源提供备份（这两个电源都工作在 RPS 模式下），那么这个 XPS 在两个失效电源全部恢复正常或者被新电源替换之前，就无法再为其他交换机的电源再提供备份了，

在混合模式下，一个电源工作在 RPS 模式下，另一个电源工作在 SP 模式下，如果一台交换机上的两个电源都发生了故障，那么由于 XPS 只能为其中一个电源提供备份，因此它不会为两个电源攻击电量，因此交换机就会关闭。只有在混合电源模式下才有可能发生这种情况。如果一台交换机与 RPS 模式的端口相连，但这台交换机上并没有工作在 RPS 模式下的电源，那么系统就会拒绝接受 RPS 端口的配置，而 XPS 则会尝试将这台交换机添加到电源堆叠中。如果这台交换机无法工作在 SP 模式下（也即这是一台不可堆叠的交换机），那么这个端口就会被禁用。

工作在 RPS 模式的端口有一个可以进行配置的优先级。默认优先级是根据 XPS 端口编号而定的，端口 1 是默认优先级最高的端口。端口优先级越高，越优先得到备份。如果一台连接到高优先级值端口的交换机上出现了电源故障的问题，同时一台连接到的优先级端口的交换机正在接受 XPS 的电源备份，那么此时 XPS 就会停止为低优先级的端口提供电源，并转而为高优先级端口供电。

堆叠电源模式

只有电源堆叠中的 Inspur 交换机才可以使用堆叠电源模式。如果没有使用 XPS，那么电源堆叠会采用环形拓扑的方式来操作，堆栈中最多可以部署 4 台交换机。如果在电源堆叠中应用了 XPS，那么堆栈加 XPS 最大可以连接 9 台交换机，它们可以采取一种类似于堆叠电源环形拓扑的操作方式来为电源堆叠中的成员提供供电。

所有在 SP 端口与 XPS 相连的 Inspur 交换机都是同一个电源堆叠的成员，所有来自于 XPS 和交换机的电能都会由堆栈中的所有交换机共享。电源共享是默认的模式，但 XPS 也支持环形拓扑中所采用的那种堆叠电源模式（严格与非严格电源共享或冗余模式）。

XPS 会采用邻居发现的方式来创建电源堆叠。当它在未配置的端口上发现了一台 Inspur 交换机时，它会将这个端口标记为 SP 端口，而这台交换机也会加入这个电源堆叠。XPS 会向交换机发送通告，会启动功率分配进程，并且会根据电源堆叠中每台交换机的需求、优先级、当前的电源分配情况、以及堆叠汇聚电源功能来给每台交换机分配功率。

XPS 会向每台交换机分配功率。如果 XPS 没有足够的功率来给每台交换机提供它们各自所需的最大电量，那么 XPS 就会根据优先级来分配功率。拥有最高优先级的交换机会首先获得自己所需的电源，接下来 XPS 会按照优先级来给每一台用电设备分配电源。剩余的电源则会在堆栈中平均进行分配。

RPS 端口优先级（取值范围是从 1 到 9）并不会影响堆叠电源优先级。每台参与堆叠电源的交换机都有自己的系统优先级，连接到其端口的设备也有或高或低的优先级。堆叠电源会按照环形拓扑中的方式来使用这些优先级。用户可以在交换机堆叠电源配置模式下，使用命令 **power-priority switch**、**power-priority high** 和命令 **power-priority low**，来给系统和高低优先级端口配置堆叠电源优先级。如果一个系统，或者一系列用电设备都在使用默认优先级，那么 XPS 会自动分配优先级（取值范围为 1 到 27），MAC 地址越小的设备获得的优先级越高。有 4 中电源堆叠模式：电源共享模式、严格电源共享模式、冗余模式和严格冗余模式。用户可以在电源堆叠配置模式下使用命令 **mode** {**power-sharing** | **redundant**} [**strict**]来配置电源的堆叠模式。在 **power-sharing**（电源共享）或 **redundant**（冗余）之间所作的选择会影响堆栈的功率分配，是否使用关键字 **strict**（严格）则会在功率缩减没有导致低压减载的情形下，影响 PoE 应用的操作。

- 在电源共享模式下（无论是否为严格电源共享模式），堆叠电源功率是堆栈中所有电源的功率之和（减去 30W 的保留电源）这是默认的模式。
- 在冗余模式下（无论是否为严格冗余模式），堆叠电源功率是电源堆叠中功率最大的电源被减去之后的总可用电源（减 30W）。冗余模式可以确保在某个电源出现故障时，没有交换机或用电设备会掉电，或者出现低压减载的情况，但如果有多个电源同时出现故

障，就会出现低压减载的情况。

- 在严格模式下，如果因输入电源掉电导致总功率减少，但并没有让任何硬件出现低压减载，那么 XPS 会开始按照优先级自低到高的顺序拒绝为低优先级用电设备供电，直至总分配电源少于等于总可用 PoE 电源。
- 在非严格模式下，如果电源减少，系统会允许总分配电源降低到设备请求的功率之下，让设备低功率运转。

例如，有一个系统的总 PoE 可用功率为 400W，可以为用电设备分配 390W 功率。设备的分配功率为设备所需的最大总功率。一组设备的实际功率消耗往往与分配功率并不相等。此时，实际功率有可能大概为 200W。如果堆栈中出现了电量丢失的情况，导致可用功率降低为 210W，那么这个功率还是足够维持用电设备实际消耗的功率，但是却小于设备在最坏情况下需要的电量，这就会让系统*低功率运转*。在严格模式下，当可以分配的功率称为 210W 甚至更低时，堆栈会立刻拒绝向用电设备提供电能。而在非严格模式下，堆栈不会采取任何行为，堆栈会放任这种状态持续下去。在非严格模式下，如果实际电源消耗大于了 210W，那么用电设备就会出现低压减载的情况，这会导致所有低优先级的用电设备或交换机都出现缺电。

混合模式

XPS 2200 也可以工作在混合模式下。在这种模式下，有些连接到交换机的端口为 RPS 模式端口，其他端口则工作在 SP 模式下。在这种配置方案中，至少有一个电源必须是 RPS 电源。XPS 中的电源只能给一个交换机电源提供备份，而 XPS 功率必须大于 RPS 模式下 XPS 端口所连交换机中的最大功率。

连接到 SP 端口的交换机都属于同一个电源堆叠。如果 SP 交换机有足够大的功率，那么 XPS 上就不需要 SP 电源。当用户配置了 XPS 电源时，其供应的电源就会被添加到电源堆叠的电源池中。

XPS 2200 系统默认参数

端口的默认角色为 Auto-SP，此时电源模式会由连接到该端口的交换机来进行判断。（运行 LAN base 镜像的 Inspur 交换机会设置为 RPS 模式，而运行 IP base 或 IP services 镜像的 Inspur 交换机会设置为 SP 模式）

XPS 电源 A（PS1）的默认模式为 RPS 模式。电源 B（PS2）的默认模式为 SP 模式。

所有端口和电源的默认模式都是启用的。

对于配置为 RPS 模式的端口来说，其默认优先级与端口编号相同。

如何配置 XPS 2200

用户可以从任何连接到 XPS 端口的交换机上对 XPS 进行配置。如果用户在多台交换机上都输入了 XPS 配置命令，那么最终生效的命令是用户最后应用的配置命令。只有交换机和端口名会保存在交换机配置文件中。

配置系统名

总步骤

1. enable

2. configure terminal

3. power xps *switch-number* **name** {*name* | **serialnumber**}

4. power xps *switch-number* **port** {*name* | **hostname** | **serialnumber**}

5. end

6. show env xps system

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式
步骤 3	**power xps** *switch-number* **name** {*name* | **serialnumber**}	**注释**：只有 Inspur 交换机上会出现 *switch-number* 这个参数，其表示交换机在这个数据堆栈中的编号，取值范围是从 1 到 9。 给 XPS 2200 系统配置一个系统名。 • *name*：给 XPS 2000 端口输入一个名称。这个名称最多可以由 20 个字符组成。 • **serialnumber**：将 XPS 2200 的序列号作为系统名。
步骤 4	**power xps** *switch-number* **port** {*name* | **hostname** | **serialnumber**}	**注释**：只有 Inspur 交换机上会出现 *switch-number* 这个参数，其表示交换机在这个数据堆栈中的编号，取值范围是从 1 到 9。 给 XPS 2200 系统配置一个系统名。 • *name*：给 XPS 2000 端口输入一个名称； • **serialnumber**：使用连接到端口的设备的序列号； • **hostname**：使用连接到端口的设备的主机名。
步骤 5	**end**	返回特权 EXEC 模式
步骤 6	**show env xps system**	验证前面配置的系统和端口名称
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 XPS 端口

总步骤

1. enable

2. power xps *switch-number* **port** {*number* | **connected**} **mode** {**disable** | **enable**}

3. power xps *switch-number* **port** {*number* | **connected**} **role** {**auto** | **rps**}

4. power xps *switch-number* **port** {*number* | **connected**} **priority** *port-priority*

5. show env xps port

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**power xps** *switch-number* **port** {*number*	**connected**} **mode** {**disable**	**enable**}	**注释**：只有 Inspur 交换机上会出现 *switch-number* 这个参数，其表示交换机在这个数据堆栈中的编号，取值范围是从 1 到 9。 设置是启用还是禁用这个端口。 • *number*：输入 XPS 2200 端口号，取值范围是从 1 到 9； • **connected**：如果不知道交换机连接的端口号，则应输入这个关键字； • **mode disable**：禁用（即关闭）这个 XPS 端口； **注释**：禁用 XPS 端口如同直接移除这条线缆，show 命令提供的输出信息也与直接断开线缆无异。不过如果连接了物理线缆，用户可以使用 enable 这个关键字来启用这个额端口。 • **mode enable**：启用这个 XPS 端口。这是默认的设置。
步骤 3	**power xps** *switch-number* **port** {*number*	**connected**} **role** {**auto**	**rps**}	**注释**：只有数据堆栈中的 Inspur 交换机上才会出现 *switch-number* 这个参数，其取值范围是从 1 到 9。 设置 XPS 端口的角色。 • **role auto**：端口模式由连接该端口的交换机来进行判断，这是默认的设置； • **role RPS**：当交换机电源出现故障时，XPS 充当备份电源。对于这个配置，至少要有一个 RPS 电源要处于 RPS 模式下。
步骤 4	**power xps** *switch-number* **port** {*number*	**connected**} **priority** *port-priority* 示例： `Device`	**注释**：只有数据堆栈中的 Inspur 交换机上才会出现 *switch-number* 这个参数，其取值范围是从 1 到 9。 设置这个端口的 RPS 优先级，如果多个电源同时出现故障，高优先级端口优先获得备用电源。只有在端口模式为 RPS 时，这条命令才能生效。如果端口模式为堆叠电源，用户就需要使用堆叠电源命令来设置优先级。 • **priority** *port-priority*：设置端口的 RPS 优先级。取值范围是从 1 到 9。1 是最高优先级。默认优先级为 XPS 端口号。
步骤 5	**show env xps port**	验证端口的 XPS 配置

配置 XPS 供电

总步骤

1. enable

2. power xps *switch-number* **supply** {**A** | **B**} **mode** {**rps** | **sp**}

3. power xps *switch-number* **supply** {**A** | **B**} {**on** | **off**}

4. end

5. show env xps power

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**power xps** *switch-number* **supply** {**A**	**B**} **mode** {**rps**	**sp**}	**注释：**只有数据堆栈中的 Inspur 交换机上才会出现 *switch-number* 这个参数，其取值范围是从 1 到 9。 设置 XPS 端口的角色。 设置 XPS 电源模式。 • **supply** {**A**	**B**}：选择要配置的电源。电源 A 是左侧（标记 PS1）的电源，而电源 B 是右侧（的 PS2）； • **mode rps**：将电源模式设置为 RPS，为连接的交换机提供备份。这是电源 A（PS1）的默认设置； • **mode sp**：将电源模式设置为堆叠电源（SP），让其参与到堆叠电源中。这是电源 B（PS2）的默认设置。
步骤 3	**power xps** *switch-number* **supply** {**A**	**B**} {**on**	**off**}	**注释：**只有数据堆栈中的 Inspur 交换机上才会出现 *switch-number* 这个参数，其取值范围是从 1 到 9。 设置 XPS 电源的开关状态。默认状态为两个电源都开启。
步骤 4	**end**	返回特权 EXEC 模式
步骤 5	**show env xps port**	验证 XPS 电源的状态

XPS 2200 的监控与维护

命令	目的
show env xps system	查看系统与端口的配置名称
show env xps port	查看端口的 XPS 配置
show env xps power	显示 XPS 供电的状态

其他参考资料

在这一节中，我们会提供关于交换机管理的参考资料。

XPS 2200 的特性历史与信息

表 16：XPS 2200 的特性信息

版本	修改
Inspur INOS XE 15.2.(3)E1	交换机引入 XPS 2200 特性

相关文档

相关主题	文档名
配置堆叠电源	Inspur INOS XE 3.7E 及后续版本的合并平台配置指南（Inspur 6850 交换机）

标准

标准	文档名
这个特性没有新标准，也没有修订的标准。关于这个特性，对当前标准的支持没有变化	—

RFC

RFC	文档名
这个特性没有新的 RFC，也没有修订的 RFC。关于这个特性，对当前标准的 RFC 没有变化	—

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

第 4 部分 IPv6

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于配置 IPv6 MLD Snooping 的信息

注释： 要想使用 IPv6 MLS Snooping，交换机必须运行 LAN Base 镜像。
用户可以使用 MLD（组播侦听者发现协议）Snooping，在交换机上向交换网络中的客户端与路由器高效地分发 IPv6（IP 协议第 6 版）组播数据。如无特别说明，交换机一词在这里指代的既可以是独立交换机，也可以是交换机堆栈。
注释： 运行 LAN Base 镜像的 Inspur 2960-X 交换机可以支持堆栈。
注释： 要想使用 IPv6，用户必须在交换机上配置双栈 IPv6 和 IPv6 交换数据库管理（SDM）模版。
运行 LAN Base 镜像的交换机特性集不支持使用路由模版。
注释： 用户若想获取本章所述全部命令的语法与使用信息，可以参阅这个版本的命令参考手册，或者 Inspur INOS 文档中的相关资料。

理解 MLD Snooping

在 IP 第 4 版中，二层交换机可以使用 IGMP（互联网组播管理协议）snooping 动态配置二层交换机，让组播流量只转发给相关的 IP 组播设备所在的接口，以此限制组播流量的泛洪。在 IPv6 中，MLD snooping 也可以执行类似的功能。通过 MLD snooping，IPv6 组播数据可以有选择地发送给那些希望接收到这些数据的端口，而不会在这个 VLAN 的所有端口进行泛洪。这个列表是交换机通过窥探 IPv6 组播控制数据包组建起来的。
MLD 是 IPv6 组播路由器使用的一项协议，其作用是在与路由器直连的发现链路上发现组播

侦听设备（也就是那些希望接收到 IPv6 组播数据包的节点），同时发现各个邻居节点分别对哪些组播数据包感兴趣。MLD 来自于 IGMP。MLD 版本 1（MLDv1）是 ICMP（互联网控制消息协议）v6 的一个子协议，MLD 消息是 ICMPv6 消息的一个子集。在 IPv6 数据包中，下一个头部（Next Header）字段的取值为 58 的消息即为 MLD 消息。

交换机支持下面两个版本的 MLD snooping：

- MLDv1 Snooping 可以监测 MLDv1 控制数据包，并且根据 IPv6 目的组播地址来建立流量的桥接关系；
- MLDv2 Basic Snooping（MBSS，MLDv2 snooping 基础版）使用 MLDv2 控制数据包来根据 IPv6 目的组播地址建立流量转发对应关系。

交换机可以对 MLDv1 和 MLDv2 协议的数据包进行窥探，并且根据目的 IPv6 组播地址来桥接 IPv6 组播数据。

注释： 交换机不支持 MLDv2 enhanced snooping（MLDv2 snooping 增强版），这种特性可以根据 IPv6 源和目的组播地址来设置转发方式。

MLD snooping 既可以在全局启用或者禁用，也可以以 VLAN 为单位启用和禁用。在启用了 MLD Snooping 后，交换机会在软件和硬件中都建立起一个以 VLAN 为单位的 IPv6 组播地址表。接下来，交换机会在硬件中执行基于 IPv6 组播地址的桥接。

根据 IPv6 组播的标准，交换机会对交换机 MAC 地址最低 4 个八位二进制数，与 33:33:00:00:00:00 这个 MAC 地址执行逻辑或（OR）运算，通过这种方式提取出组播 MAC 地址。例如，FF02:DEAD:BEEF:1:3 这个 IPv6 MAC 地址对应的以太网 MAC 地址就是 33:33:00:01:00:03。

如果目的 IPv6 地址不匹配目的 MAC 地址，则组播数据包就是不匹配的。交换机会根据 MAC 地址表，对不匹配的数据包执行硬件转发。如果交换机的 MAC 地址表中没有这个目的 MAC 地址，那么交换机就会以同一个 VLAN 的所有端口作为接收方端口，来泛洪这个数据包。

MLD 消息

MLDv1 支持三种类型的消息：

- 侦听者查询消息（Listener Query）相当于 IGMPv2 查询消息，它可以执行总的查询，也可以针对某个特定的 MAC 地址进行查询；
- 组播侦听者报告（Multicast Listener Report）相当于 IGMPv2 报告；
- 组播侦听者完成（Multicast Listener Done）消息相当于 IGMPv2 离开消息。

MLDv2 支持 MLDv2 查询和报告，和 MLDv1 报告和完成消息。

消息计时器和因为消息收发而导致的状态过渡，都与 IGMPv2 消息相同。MLD 路由器和交换机会忽略那些没有有效链路本地 IPv6 源地址的 MLD 消息。

MLD 查询

交换机会发送 MLD 查询消息，建立 IPv6 组播地址数据库，并且会使用 MLD 特性组和 MLD 组与特定源查询消息来响应 MLD 完成消息。交换机还支持报告抑制（report suppression）、报告代理（report proxying）、直接离开（Immediate-Leave）功能和静态 IPv6 组播组地址配置。

在禁用了 MLD snooping 之后，所有 MLD 查询消息都会在消息的入站 VLAN 中进行泛洪。

在启用了 MLD snooping 之后，交换机会将接收到的 MLD 查询消息在入站 VLAN 中进行泛洪，同时将查询消息发送给 CPU 进行处理。MLD snooping 会通过接收到的查询消息来建立 IPv6 组播地址数据库。它会检测组播路由器的端口、维护计时器、设置报告响应时间、学习这个 VLAN 中的查询方源 IP 地址、学习这个 VLAN 中的查询方端口，监控组播地址时间的老化情况。

~~**注释：** 当这台 IPv6 组播路由器是一台 Inspur 6500 交换机，而且用户使用的又是扩展 VLAN（即范围在 1006 到 4094 之间的 VLAN）时，那么用户就必须在 Inspur 6500 交换机的扩展~~

~~VLAN 上启用 IPv6 MLD snooping，这是为了让 Inspur 2960、2960S、2960C、2960X 或 2960CX 交换机能够在这个 VLAN 上接收到查询消息。如果使用的是正常范围 VLAN（即范围在 1 到 1005 之间的 VLAN），用户可以不必在 Inspur 6500 交换机的这些 VLAN 上启用 IPv6 MLD。~~

当 MLD snooping 数据库中存在一个组时，交换机就可以通过发送 MLDv1 报告来对针对该组的查询作出响应。如果这个组是未知的，交换机就会把针对这个组的查询在整个入站 VLAN 当中进行泛洪。

当一台主机想要离开一个组播组时，它可以发送一条 MLD 完成消息（相当于 IGMP 离开消息）。交换机在接收到这个 MLDv1 完成消息之后，如果没有启用直接离开（Immediate-Leave）特性，那么交换机就会向接收到这个消息的端口发送一条 MASQ 消息，以判断这个端口是否连接了其他希望继续保持在这个组播组中的设备。

组播客户端老化的稳健性

用户可以对查询的数量进行配置，让不达标的端口成员离开其对应的组播地址。只有当对某个地址的报告数量达不到用户配置的查询数量时，这个端口才会被交换机从响应的组播组地址中移除出去。默认的查询数量为 2。

组播路由器发现

MLD snooping 也和 IGMP snooping 一样执行组播路由器发现，组播路由器发现拥有下列特征：

- 用户所配置的端口永不老化；
- 通过 MLDv1 snooping 查询和 IPv6 PIMv2 数据包实现动态端口学习；
- 如果同一个二层接口连接了多台路由器，MLD snooping 只会在该端口追踪一台组播路由器（追踪的是最近发送路由器控制数据包的那台路由器）；
- 动态组播路由器端口老化默认的计时器时间为 5 分钟；如果端口连续 5 分钟没有接收到控制数据包，那么这台组播路由器就会路由器端口列表中删除；
- 只有在交换机上启用了 MLD snooping 的情况下，IPv6 组播路由器发现才会生效；
- 接收到的 IPv6 组播路由器控制数据包一定会在入站 VLAN 中进行泛洪，这与交换机上是否启用 MLD snooping 无关；
- 在设备发现了第一个 IPv6 组播路由器端口之后，它就会开始仅向发现的路由器端口转发 IPv6 组播数据。（而在此之前，所有 IPv6 组播数据都会在入站 VLAN 中进行泛洪）

MLD 报告

MLDv1 加入消息的处理方式与 IGMPv2 基本相同。当一个 VLAN 中没有检测到 IPv6 组播路由器时，交换机就不会处理或转发报告。而当设备检测到了 IPv6 组播路由器，并且接收到了 MLDv1 报告之后，它就会在 VLAN MLD 数据库中输入一个 IPv6 组播组地址。接下来，所有在这个 VLAN 中去往这个组的 IPv6 组播流量都会用这个地址进行转发。如果设备禁用了 MLD snooping，那么报告就会在入站 VLAN 中进行泛洪。

如果启用了 MLD snooping 特性，那么 MLD 报告抑制（称为侦听者消息抑制）也会自动启用。通过报告抑制特性，交换机就会将第一个组接收到的 MLDv1 报告转发给 IPv6 组播路由器；但它不会再将后续的组报告发送给路由器。如果禁用了 MLD snooping，那么报告抑制特性也会被禁用，因此所有 MLDv1 报告都会在入站 VLAN 中进行泛洪。

交换机也支持 MLDv1 代理报告功能。当交换机接收到一个 MLDv1 MASQ 消息时，如果交换机的另一个端口存在这个组，并且查询消息到达的那个端口不是该地址的最后一个成员端口的话，那么交换机就会向查询消息到达的那个地址发送 MLDv1 报告，以响应 MLDv1 MASQ 响应。

MLD 完成消息与直接离开特性

如果交换机上启用了直接离开（Immediate-Leave）特性，并且一台主机发送了一个 MLDv1 消

息（相当于 IGMP 离开消息），那么交换机会立刻将接收到完成消息的那个端口从组中删除。用户可以以 VLAN 为单位启用直接离开特性，此时用户应该只在那些 VLAN 成员端口都只连接了一台主机的 VLAN 中启用这项特性（这一点和 IGMP snooping 相同）。如果这个端口是一个组成员的最后一个端口，那么这个组也会一并被交换机删除，同时交换机还会将离开信息转发给被删除的那台 IPv6 组播路由器。

如果一个 VLAN 中没有启用直接离开特性（当某个组中，存在有多个客户端连接在同一个端口上的情形时，就不应该在启用该特性），而该 VLAN 中有端口接收到一个完成消息，那么这个端口就会生成一个 MASQ。用户可以根据接收到的 MASQ 数量，来控制何时移除对某个地址移除一个端口的成员身份。在端口接收到的查询次数达到了用户配置的数值，但该端口并没有去往对应地址的 MLDv1 报告时，交换机就会删除这个端口在该地址的成员身份。

用户可以使用全局配置命令 **ipv6 mld snooping last-listener-query count** 来配置生成的 MASQ 数量。默认的数量为 2。

交换机会将 MASQ 发送给完成消息的目的地址。如果在交换机最大响应时间之内，没有报告消息发送给 MASQ 中指定的 IPv6 组播地址，交换机就会将发送 MASQ 的端口从 IPv6 组播地址数据库中删除。用户可以通过全局配置命令 **ipv6 mld snooping last-listener-query-interval** 来配置最大响应时间。如果交换机删除的端口是组播地址的最后一个成员端口，那么交换机也会同时删除这个组播地址，同时交换机会向所有被删除的组播路由器发送一个地址离开信息。

在没有启用直接离开特性的情况下，如果某个端口接收到了一条 MLD 完成消息，那么交换机就会在这个端口上创建 MASQ，并且将这些消息发送给发送完成消息的那个 IPv6 组播地址。用户可以对发送多少 MASQ 进行配置，也可以配置交换机在从组播组中删除端口之前，等待响应消息的时长。

如果启用了 MLDv1 直接离开特性，那么当交换机在某个端口上检测到了一个 MLD 完成消息时，它就会立刻将这个端口从组播组中移除。只有在 VLAN 中每个端口都只连接了一台接收方设备时，用户才可以考虑在这个 VLAN 上使用直接离开特性。如果同一个端口上连接了某个组播组的多台客户端，那就不要在这个端口所在的 VLAN 启用直接离开特性。

拓扑变化通告处理

在用户使用全局配置命令 **ipv6 mld snooping tcn query solicit** 启用了通告（TCN）请求（solicitation）特性之后，MLDv1 snooping 就会对 VLAN 泛洪自己配置数量的 MLDv1 查询消息，然后再将组播数据发送给所选的端口。用户可以使用全局配置命令 **ipv6 mld snooping tcn flood query count** 来设置这个数值。默认值为发送 2 条查询消息。交换机也会在交换机成为这个 VLAN 中的 STP 根，或者用户将其配置为这个 VLAN 的 STP 根时，生成 MLDv1 的全局完成消息，这种做法与 IGMP snooping 相同。

如何配置 IPv6 MLD Snooping

默认的 MLD Snooping 配置

特性	默认设置
MLD snooping（全局）	禁用
MLD snooping（各个 VLAN）	启用。不过 MLD snooping 必须首先在全局启用，各个 VLAN 的 MLD snooping 才能生效
IPv6 组播地址	未配置

IPv6 组播路由器端口	未配置
MLD snooping 直接离开（Immediate Leave）特性	禁用
MLD snooping 稳健性（robustness）变量	全局：2；各个 VLAN：0 **注释**：各个 VLAN 的参数优于全局设置。当 VLAN 值为 0 时，VLAN 才会使用全局值
最后的侦听者查询数	全局：2；各个 VLAN：0 **注释**：各个 VLAN 的参数优于全局设置。当 VLAN 值为 0 时，VLAN 才会使用全局值
最后的侦听者查询间隔	全局：1000（即 1 秒）；各个 VLAN：0 **注释**：各个 VLAN 的参数优于全局设置。当 VLAN 值为 0 时，VLAN 才会使用全局值
TCN 查询请求	禁用
TCP 查询数	2
MLD 侦听者抑制	禁用

MLD Snooping 配置指南

在配置 MLD snooping 时，可以考虑下面的指导方针：

- 用户可以随时配置 MLD snooping 特征，但必须使用全局配置命令 **ipv6 mld snooping** 在全局启用 MLD snooping 才能让配置生效；
- ~~当这台 IPv6 组播路由器是一台 Inspur 6500 交换机，而且用户使用的又是扩展 VLAN（即范围在 1006 到 4094 之间的 VLAN）时，那么用户就必须在 Inspur 6500 交换机的扩展 VLAN 上启用 IPv6 MLD snooping，这是为了让交换机能够在这个 VLAN 上接收到查询消息。如果使用的是正常范围 VLAN（即范围在 1 到 1005 之间的 VLAN），用户可以不必在 Inspur 6500 交换机的这些 VLAN 上启用 IPv6 MLD；~~
- MLD snooping 与 IGMP snooping 是相互独立工作的。用户可以在交换机上同时启用这两个特性；
- 交换机或交换机堆栈上允许的最大组播条目数量是由用户配置的 SDM 模板来决定的；
- 交换机或交换机堆栈上允许的最大地址条目数量为 4000 条。

在交换机上启用或禁用 MLD Snooping（CLI 界面配置方法）

在默认情况下，IPv6 MLD snooping 在交换机上是全局禁用的，但同时是在所有 VLAN 上启用的。当 MLD snooping 在全局禁用时，它也不会真的在所有 VLAN 上生效。而当用户全局启用 MLD snooping 时，各个 VLAN 的配置就会覆盖全局的配置。也就是说，在默认状态下，MLD snooping 只有在 VLAN 接口上是启用的。

对于一个范围内的 VLAN，用户可以针对各个 VLAN 分别启用和禁用 MLD snooping，但如果用户在全局禁用了 MLD snooping，那么所有 VLAN 上配置的 MLD snooping 也会被禁用。如果在全局启用了 snooping，那么用户也就可以给各个 VLAN 设置是否启用 snooping。

用户可以从特权 EXEC 模式中，按照下面的步骤在交换机上全局启用 MLD snooping：

具体步骤

	命令或操作	**目的**

步骤 1	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 2	**ipv6 mld snooping** 示例： `Device(config)# ipv6 mld snooping`	在交换机上启用 MLD snooping
步骤 3	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 4	**copy running-config startup-config** 示例： `Device(config)# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中
步骤 5	**reload** 示例： `Device(config)# reload`	重启操作系统

对一个 VLAN 启用或禁用 MLD Snooping（CLI 界面配置方法）

用户可以从特权 EXEC 模式中，按照下面的步骤在一个 VLAN 上启用 MLD snooping：

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 2	**ipv6 mld snooping** 示例： `Device(config)# ipv6 mld snooping`	在交换机上启用 MLD snooping
步骤 3	**ipv6 mld snooping vlan** *vlan-id* 示例： `Device(config)# ipv6 mld snooping vlan 1`	在一个 VLAN 上启用 MLD snooping。VLAN ID 的取值范围是从 1 到 1001，以及从 1006 到 4094. **注释**：必须在全局启用 MLD snooping， VLAN snooping 才能生效

步骤 4	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式

配置一个静态组播组（CLI 界面配置方法）

主机或二层端口一般会动态加入组播组，但用户也可以给一个 VLAN 静态配置 IPv6 组播地址和成员端口。

用户可以从特权 EXEC 模式中，按照下面的步骤将一个二层端口添加为组播组的成员：

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**ipv6 mld snooping vlan** *vlan-id* **static** *ipv6_multicast_address* **interface** *interface-id* 示例： `Device(config)#` **ipv6 mld snooping vlan 1 static FF12::3 interface gigabitethernet 0/1**	将一个二层端口配置为一个组播组的成员端口： • *vlan-id* 是组播组的 VLAN ID。VLAN ID 的范围是从 1 到 1001，以及从 1006 到 4094； • *ipv6_multicast_address* 是 128 位组 IPv6 地址。这个地址必须按照 RFC 2373 的格式来进行配置； • *interfaces-id* 是成员端口。这个端口既可以是物理接口，也可以是 port channel（编号 1 到 48）。
步骤 3	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 4	使用下面两条命令之一： • **show ipv6 mld snooping address** • **show ipv6 mld snooping address vlan** *vlan-id* 示例： `Device#` **show ipv6 mld snooping address** 或 `Device#` **show ipv6 mld snooping vlan 1**	验证静态成员端口和 IPv6 地址

配置一个组播路由器端口（CLI 界面配置方法）

注释： 只有交换机端口支持与组播路由器之间的静态连接

用户可以从特权 EXEC 模式中，按照下面的步骤向一个 VLAN 中添加一个组播路由器端口：

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**ipv6 mld snooping vlan** *vlan-id* **static** *ipv6_multicast_address* **interface** *interface-id* 示例： `Device(config)#` **ipv6 mld snooping vlan 1 static FF12::3 interface gigabitethernet 0/1**	设置组播路由器 VLAN ID 并将这个接口指定给组播路由器： • VLAN ID 的范围是从 1 到 1001，以及从 1006 到 4094； • 这个端口既可以是物理接口，也可以是 port channel。后者的编号范围是从 1 到 48。
步骤 3	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 4	使用下面两条命令之一： **show ipv6 mld snooping mrouter** [**vlan** *vlan-id*] 示例： `Device#` **show ipv6 mld snooping mrouter vlan 1**	验证该 VLAN 接口已经启用了 IPv6 MLD Snooping

启用 MLD 直接离开特性（CLI 界面配置方法）

用户可以从特权 EXEC 模式中，按照下面的步骤启用 MLDv1 直接离开特性：

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**ipv6 mld snooping vlan** *vlan-id* **immediate-leave**	在一个 VLAN 接口上启用 MLD 直接离开特性

	示例： `Device(config)#` **ipv6 mld snooping vlan 1 immediate-leave**	
步骤 3	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 4	**show ipv6 mld snooping mrouter** [**vlan** *vlan-id*] 示例： `Device#` **show ipv6 mld snooping mrouter vlan 1**	验证该 VLAN 接口已经启用了直接离开特性

配置 MLD Snooping 查询（CLI 界面配置方法）

用户可以从特权 EXEC 模式中，按照下面的步骤为交换机或 VLAN 配置 MLD Snooping 查询的相关功能：

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**ipv6 mld snooping robustness-variable** *value* 示例： `Device(config)#` **ipv6 mld snooping robustness-variable 3**	（可选）设置交换机在将一个没有对一般性查询作出响应的侦听设备（端口）删除之前，会发送多少条查询消息。这个参数的取值范围是从 1 到 3，默认设置为 2
步骤 3	**ipv6 mld snooping vlan** *vlan-id* **robustness-variable** *value* 示例： `Device(config)#` **ipv6 mld snooping vlan 1 robustness-variable 3**	（可选）给各个 VLAN 分别设置稳健性变量，这个变量的目的是指明 MLD snooping 特性在因没有收到 MLD 报告响应消息而让一个组播地址老化之前，会发送多少条一般性查询消息。这个参数的取值范围是从 1 到 3，默认设置为 2。如果设置为 0，则针对该 VLAN 应用全局的稳健性变量参数
步骤 4	**ipv6 mld snooping last-listener-query-count** *count* 示例：	（可选）设置交换机让一个 MLD 客户端老化之前，会发送多少条 MASQ。这个参数的取值范围是从 1 到 7，默认设置为 2。查询消息每隔 1 秒发送一次

	Device(config)# **ipv6 mld snooping last-listener-query-count 7**	
步骤 5	**ipv6 mld snooping vlan** *vlan-id* **last-listener-query-count** *count* 示例： Device(config)# **ipv6 mld snooping vlan 1 last-listener-query-count 7**	（可选）给各个 VLAN 设置最后侦听者查询计数。这个值会覆盖全局配置的参数。这个参数的取值范围是从 1 到 7，默认设置为 0。如果设置为 0，则针对该 VLAN 应用全局的计数值。查询消息每隔 1 秒发送一次
步骤 6	**ipv6 mld snooping last-listener-query-interval** *interval* 示例： Device(config)# **ipv6 mld snooping last-listener-query-interval 2000**	（可选）设置交换机在发送 MASQ 之后，会等待的最大响应时间，经过这段时间没有得到响应，交换机才会将端口从这个组播组中删除。这个参数的取值范围是从 100 到 32768 毫妙，默认设置为 1000 毫妙（即 1 秒）。
步骤 7	**ipv6 mld snooping vlan** *vlan-id* **last-listener-query-interval** *interval* 示例： Device(config)# **ipv6 mld snooping vlan 1 last-listener-query-interval 2000**	（可选）给各个 VLAN 设置最后侦听者查询间隔时间。这个值会覆盖全局配置的参数。这个参数的取值范围是从从 0 到 32768 毫妙，默认设置为 0。如果设置为 0，则针对该 VLAN 应用全局的最后侦听者查询间隔时间
步骤 8	**ipv6 mld snooping tcn query solicit** 示例： Device(config)# **ipv6 mld snooping tcn query solicit**	（可选）启用拓扑变更通告（TCN）请求，即 VLAN 会对所有 IPv6 组播流量泛洪用户指定数量的请求消息，然后才会将组播数据专门发送给那些请求接收这些数据的端口。TCN 默认是禁用的。
步骤 9	**ipv6 mld snooping tcn flood query count** *count* 示例： Device(config)# **ipv6 mld snooping tcn flood query count 5**	（可选）在启用了 TCN 之后，用户需要设置发送 TCN 请求的次数。这个参数的取值范围是从从 1 到 10，默认设置为 2。
步骤 10	**end** 示例： Device(config)# **end**	返回特权 EXEC 模式
步骤 11	**show ipv6 mld snooping**	（可选）验证为交换机或 VLAN 配置的 MLD

	querier [**vlan** *vlan-id*] 示例： `Device(config)#` **show ipv6 mld snooping querier vlan 1**	snooping 查询方信息

禁用 MLD 侦听者消息抑制（CLI 界面配置方法）

MLD snooping 侦听者消息抑制在默认状态下就是启用的。在启用这个特性之后，交换机只会针对每个组播路由器查询消息转发一个 MLD 报告消息。在禁用了这个消息抑制特性之后，交换机可以向组播路由器转发多个 MLD 报告消息。
用户可以从特权 EXEC 模式中，按照下面的步骤禁用 MLD 侦听者消息抑制特性：
具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**no ipv6 mld snooping listener-message-suppression** 示例： `Device(config)#` **no ipv6 mld snooping listener-message-suppression**	禁用 MLD 消息抑制特性
步骤 3	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 4	**show ipv6 mld snooping** 示例： `Device#` **show ipv6 mld snooping**	验证 IPv6 MLD snooping 报告抑制特性已经禁用

查看 MLD Snooping 的信息

用户可以查看路由器端口或 VLAN 接口那些动态学习或静态配置的 MLD snooping 信息，也可以查看针对 MLD snooping 给一个 VLAN 配置的 IPv6 组地址组播条目。
表 18：显示 MLD snooping 信息的命令

命令	目的
show ipv6 mld snooping [**vlan** *vlan-id*]	显示这台交换机上所有 VLAN 或某些特性 VLAN 的 MLD snooping 配置信息。 （可选）输入 **vlan** *vlan-id* 可以让系统显示一

	个 VLAN 的信息。VLAN ID 的取值范围是从 1 到 1001，从 1006 到 4094
show ipv6 mld snooping mrouter [**vlan** *vlan-id*]	显示动态学习和手动配置的组播路由器接口信息。如果启用了 MLD snooping，那么交换机就会自动学习组播路由器连接的端口。这些就是动态学习的端口。 （可选）输入 **vlan** *vlan-id* 可以让系统显示一个 VLAN 的信息。VLAN ID 的取值范围是从 1 到 1001，从 1006 到 4094
show ipv6 mld snooping querier [**vlan** *vlan-id*]	显示 VLAN 中最近接收到的 MLD 查询消息的 IPv6 地址和入站端口。 （可选）输入 **vlan** *vlan-id* 可以让系统显示一个 VLAN 的信息。VLAN ID 的取值范围是从 1 到 1001，从 1006 到 4094
show ipv6 mld snooping address [**vlan** *vlan-id*] [**count** | **dynamic** | **user**]	显示交换机或一个 VLAN 的所有 IPv6 组播地址信息或特定 IPv6 组播地址信息。 • 输入 **count** 则系统会显示交换机或一个 VLAN 中的组数量； • 输入 **dynamic** 则系统会显示交换机或一个 VLAN 通过 MLD snooping 学习到的组信息； • 输入 **user** 则系统会显示用户给交换机或一个 VLAN 配置的组信息。
show ipv6 mld snooping address vlan *vlan-id* [*ipv6-multicast-address*]	显示特定 VLAN 和 IPv6 组播地址的 MLD snooping 信息

配置 MLD Snooping 的示例

配置静态组播组：示例

这个示例显示了静态配置 IPv6 组播组的方法：

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 2 static FF12::3 interface gigabitethernet 1/0/1
Device(config)# end
```

配置组播路由器端口：示例

这个示例显示了向 VLAN 200 中添加组播路由器端口的方法：

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 200 mrouter interface gigabitethernet 0/2
Device(config)# exit
```

启用 MLD 直接离开（Immediate-Leave）特性：示例

这个示例显示了如何在 VLAN 130 上启用 MLD 直接离开特性：

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 130 immediate-leave
Device(config)# exit
```

配置 MLD Snooping 查询：示例

这个示例显示了如何将 MLD snooping 全局稳健性变量设置为 3：

```
Device# configure terminal
Device(config)# ipv6 mld snooping robustness-variable 3
Device(config)# exit
```

这个示例显示如何将一个 VLAN 的 MLD snooping 最后侦听者查询数量设置为 3：

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 200 last-listener-query-count 3
Device(config)# exit
```

这个示例显示如何将 MLD snooping 最后侦听者查询时间间隔（最大响应时间）设置为 2000（即两秒）：

```
Device# configure terminal
Device(config)# ipv6 mld snooping last-listener-query-interval 2000
Device(config)# exit
```

配置 IPv6 单播路由

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导

航系统。

关于配置 IPv6 单播路由的信息

本章会描述如何在交换机上配置 IPv6 单播路由。

注释： 要想使用这一章介绍的所有 IPv6 特性，交换机或堆栈主交换机必须运行 IP services 特性集。运行 IP base 特性集的交换机支持 IPv6 静态路由、IPv6 RIP 和 IPv6 OSPF。运行 LAN base 特性集的交换机则只支持 IPv6 主机功能。

理解 IPv6

IPv4 用户如果迁移到 IPv6 环境中，就可以得到诸如端到端的安全、服务质量（QoS）和全局唯一地址等服务。IPv6 庞大的地址空间降低了人们对私有地址空间的需求，位于网络边缘的边界路由器不必处理大量的网络地址转换（NAT）操作。

要想进一步了解 Inspur 系统实施 IPv6 的相关信息，可以访问：

http://www.icntnetworks.com

要想进一步了解本章中介绍的 IPv6 和其他相关特性，可以参阅《INOS IPv6 配置库》。

用户可以通过 icntnetworks.com 中的搜索栏来查找 Inspur INOS 软件的技术文档。例如，如果用户希望了解关于静态路由的信息，可以在搜索栏中输入“实施 IPv6 静态路由”来了解关于静态路由的信息。

IPv6 地址

交换机只支持 IPv6 单播地址。它不支持站点本地单播地址或者任意播地址。

IPv6 的 128 位地址是通过一系列 8 个用英文冒号分割的十六进制数来表示的，其中每个十六进制数的长度为 16 位二进制数，其格式为 n:n:n:n:n:n:n:n。下面是一个 IPv6 地址的示例：

2031:0000:130F:0000:0000:09C0:080F:130B

为简化期间，每部分中的前导 0 可以省略。IPv6 地址每一段中有没有前导 0 都是一样的，所以上面的 IPv6 可以简化为：

2031:0:130F:0:0:9C0:80F:130B

我们也可以使用两个英文冒号（::）来代表连续多个全 0 的十六进制字段，但这种双冒号的简化表示法每个地址中只能使用一次：

2031:0:130F::09C0:080F:130B

要想进一步了解关于 IPv6 地址格式、地址类型和 IPv6 数据包头部的信息，可以参阅 icntnetworks.com 中，“Inspur INOS IPv6 配置库”一文中的“实施 IPv6 编址和基本的连通性”一章。

在“关于实施 IPv6 基本连通性的信息”一章中，下面几节的内容适用于交换机环境：

- IPv6 地址格式
- IPv6 地址类型：单播
- IPv6 地址类型：组播
- IPv6 地址输出显示
- 简化的 IPv6 数据包头部

支持的 IPv6 单播路由特性

在这一部分中，我们会介绍交换机支持的 IPv6 协议特性：

交换机可以通过 IPv6 版 RIP（路由信息协议）、和 OSPF（开放式最短路径优先）协议第 3 版

来提供 IPv6 路由功能。交换机支持最多 16 条等价路由，也可以用线速并行转发 IPv4 数据帧和 IPv6 数据帧。

128 位宽单播地址

交换机支持可汇总的全局单播地址和链路本地单播地址。交换机不支持站点本地单播地址。可汇总全局单播地址是包含可汇总全局单播前缀的 IPv6 地址。这种地址结构可以实现对路由前缀的严格汇总，因此可以限制全局路由表中路由条目的数量。这类地址用于可以对整个机构的地址进行汇总，或者最终连接到互联网服务提供商的链路上。

这类地址是通过全局路由前缀、子网 ID 和接口 ID 进行定义的。当前的全局单播地址是从二进制值 001（2000::/3）开始的地址范围进行分配的。前缀为 2000::/3（001）到 E000::/3（111）的地址必须包含 EUI（扩展唯一标识符）-64 格式的 64 位接口标识符。

链路本地单播地址可以在任何接口上自动进行配置，这类地址由链路本地前缀 FE80::/80（1111 1110 10）和修改的 EUI 格式的接口标识符构成。链路本地地址会用于邻居发现协议（NDP）和无状态地址自动配置的进程。本地链路上的节点会使用链路本地地址，它们不需要配置全局唯一地址就可以实现通信。IPv6 路由器不会将以链路本地地址作为源或目的的数据包转发到其他链路上。

要想进一步了解相关信息，可以参阅 icntnetworks.com 中，“Inspur INOS IPv6 配置库”一文中的“实施 IPv6 编址和基本的连通性”一章。

IPv6 DNS

IPv6 支持 DNS 域名-地址和地址-域名查询进程中的 DNS（域名系统）记录类型。DNS AAAA 资源记录类型支持 IPv6 地址，它相当于 IPv4 中的 A 地址类型。交换机支持针对 IPv4 和 IPv6 地址执行 DNS 解析。

IPv6 单播的路径 MTU 发现

交换机支持向 IPv6 节点通告系统最大传输单元（MTU），也支持路径 MTU 发现。路径 MTU 发现可以让主机动态发现一条给定数据路径中各个链路的 MTU 值，并且以此对 MTU 进行调整。在 IPv6 环境中，如果路径中一条链路的 MTU 小于数据包的大小，那么数据包的源设备就会对数据进行分片。

ICMPv6

IPv6 版的互联网控制消息协议（ICMP）可以在进行处理和执行错误诊断时，创建各类错误消息（譬如 ICMP 目的地址不可达消息）来报告网络中的错误。在 IPv6 环境中，ICMP 数据包也会用于邻居发现协议和路径 MTU 发现功能。

邻居发现

交换机支持 IPv6 的 NDP（这是一项运行在 ICMPv6 之上的协议），也支持给那些不支持 NDP 的 IPv6 工作站静态配置邻居条目。IPv6 邻居发现进程会使用 ICMP 消息和请求节点组播地址来判断同一个网络（本地链路）中邻居设备的链路层地址，以便测试邻居的可达性，并且追踪邻居路由器。

交换机支持对那些掩码长度少于 64 位的路由执行 ICMPv6 重定向，但不支持对那些掩码长度大于 64 位的主机路由或汇总路由进行 ICMP 重定向。

邻居发现压制（throttling）可以确保交换机在处理下一跳转发信息，以路由 IPv6 数据包时，CPU 不至于出现过载的情况。当下一跳就是交换机正在主动尝试解析的邻居时，交换机就会丢弃 IPv6 数据包。这可以避免进一步增加 CPU 的负担。

默认路由器优先级

交换机支持 IPv6 默认路由器优先级（DRP），这是路由器通告消息的一个扩展部分。DRP 可以提升主机的功能，让主机能够选择合适的路由器，这种技术特别适合用于那些多宿主的主机，且路由器处于不同链路的情形。交换机不支持 RFC 4191 中定义的路由信息可选项。

IPv6 主机会维护一个默认路由器列表，主机会从列表中选择一台路由器来向其转发去往链路外目的地址的流量。被选中为某个目的地址转发流量的路由器会被缓存到目的地址缓存当中。IPv6 NDP 规定，那些可达的或者很可能可达的路由器要优于那些可达性未知或者可达性成疑的路由器。对于可达或者很可能可达的路由器，NDP 既可以每次都选择相同的路由器，也可以在路由器列表中循环选用。通过 DRP，用户可以对一台 IPv6 主机进行配置，让某一台路由器的优先级比另一台路由器优先级高，当然前提是这两台路由器都要是可达或者很可能可达的路由器。

要想进一步了解 IPv6 DRP，可以参阅 icntnetworks.com 中的“Inspur INOS IPv6 配置库”一文。

IPv6 无状态自动配置与重复地址检测

交换机会使用无状态自动配置来管理链路、子网和站点的地址变更，这包括对主机和移动 IP 地址的管理。主机会自动配置自己的链路本地地址，而启动节点会发送路由器请求消息来请求路由器通告，以配置接口地址。

要想进一步了解关于自动配置与重复地址检测的信息，可以参阅 icntnetworks.com 中，“Inspur INOS IPv6 配置库”一文中的“实施 IPv6 编址和基本的连通性”一章。

IPv6 应用

交换机支持下列 IPv6 应用：

Ping、traceroute、Telnet 和 TFTP；

通过 IPv6 传输来实现 SSH；

通过 IPv6 传输来访问 HTTP 服务器；

通过 IPv4 传输来对 AAAA 执行 DNS 解析；

思科发现协议（CDP）支持 IPv6 地址

要想进一步了解关于管理这些应用的信息，可以参阅 icntnetworks.com 中的“Inspur INOS IPv6 配置库”一文。

使用 DHCP 来分配 IPv6 地址

DHCPv6 可以通过 DHCP 服务器来向 IPv6 客户端传输配置参数，其中包括 IPv6 网络地址。地址分配特性会根据主机所连接的网络，用正确的前缀来分配不重复的地址。分配的地址可以来自于一个或多个地址池。还有一些其他的可选项（如默认域和 DNS 名称服务器地址）也可以由 DHCP 服务器传输给客户端。地址池中的地址可以分配给一个特定的接口或者多个接口，DHCP 服务器也可以自动找到正确的地址池。

要想进一步了解关于管理这些应用的信息，可以参阅《Inspur INOS IPv6 配置指南》。

本文仅仅描述了 DHCPv6 的地址分配方式。要想进一步了解关于配置 DHCPv6 客户端、服务器或中继代理功能的信息，可以访问 icntnetworks.com，参阅“Inspur INOS IPv6 配置库”中的“实施 IPv6 DHCP”一章。

IPv6 静态路由

静态路由是手动进行配置的、用来明确定义两台网络设备之间路径的路由条目。静态路由适用于那些只有一条路径通往外部网络的小型网络环境，也可以在大型网络中的某些流量类型提供安全性保护。

要想进一步了解关于静态路由的信息，可以访问 icntnetworks.com，参阅“Inspur INOS IPv6 配置库”中的“实施 IPv6 静态路由”一章。

IPv6 RIP

IPv6 版的路由信息协议（RIP）是一种距离矢量型协议，这种协议会使用跳数作为路由度量值。这种路由协议可以支持 IPv6 地址和前缀，它会用所有 RIP 路由器组播组地址 FF02::9 作为 RIP 更新消息的目的地址。

要想进一步了解关于 IPv6 RIP 的信息，可以访问 icntnetworks.com，参阅“Inspur INOS IPv6 配

置库”中的“实施 IPv6 RIP”一章。

IPv6 OSPF

运行 IP Base 镜像特性集的交换机支持 IPv6 版的最短路径优先（OSPF）协议，这是一种链路状态型 IP 协议。要想进一步了解相关的信息，可以访问 icntnetworks.com，参阅“Inspur INOS IPv6 配置库”。

配置 IPv6 HSRP

HSRP 可以为路由 IPv6 流量提供路由冗余，让流量不再依赖于某一台路由器进行转发。IPv6 主机会通过 IPv6 邻居发现路由器通告消息来学习网络中可用的路由器。这些信息是路由设备组播周期性进行发送的，或者由主机请求发送的。

每个 HSRP IPv6 组都有一个虚拟的 MAC 地址（这个 MAC 地址取自于 HSRP 组的编号）和一个虚拟的 IPv6 链路本地地址（在默认情况下取自于 HSRP 虚拟 MAC 地址）。当 HSRP 组处于活跃状态时，设备就会使用 HSRP 虚拟 IPv6 链路本地地址来周期性地发送消息。而当这个 HSRP 组不再处于活跃状态时，在最后一个消息更结束之后，设备就不会再发送周期性的消息了。

注释： 在配置 IPv6 HSRP 时，必须在接口上启用 HSRP 第 2 版（HSRPv2）。

IPv6 EIGRP

交换机支持 IPv6 版的增强型内部网关路由协议（EIGRP）。用户可以在要运行这个协议的接口上配置该协议，而不必配置全局 IPv6 地址。运行 IP Lite 镜像的交换机只支持 EIGRPv6 末节路由。

在运行之前，IPv6 版 EIGRP 需要获得一个路由器 ID。隐式的路由器 ID 是从本地 IPv6 地址中提取出来的，所以往往每个 IPv6 节点都可以获得一个可用的路由器 ID。但 IPv6 版的 EIGRP 有可能会在一个只有 IPv6 节点的网络中运行，因此 IPv6 节点可能没有可用的 IPv6 路由器 ID。

要想进一步了解关于 IPv6 EIGRP 的信息，可以访问 icntnetworks.com，参阅“Inspur INOS IPv6 配置库”中的“实施 IPv6 EIGRP”一章。

EIGRPv6 末节路由

EIGRPv6 末节路由特性可以将被路由的流量移动到距离终端用户更近的位置，以此介绍对网络资源的占用。

在使用 EIGRPv6 末节路由的网络中，唯一允许向用户转发 IPv6 流量的路由就是那些穿越配置了 EIGRPv6 末节路由的交换机，发往终端用户的路由条目。交换机会将被路由流量发送给那些配置为用户接口的端口，或者那些连接到其他设备的端口。

在使用 EIGRPv6 末节路由时，用户需要通过配置让转发路由器和远程路由器使用 EIGRPv6，同时只将这台交换机配置为末节设备。只有用户指定的路由才可以由交换机通告给其他设备。交换机会对所有针对汇总路由、直连路由和路由更新的查询消息作出响应。

当邻居设备接收到一个向它通告末节状态的数据包时，它不会向末节路由器查询任何路由信息，而连接有末节对等体的路由器也不会向这台对等体设备发送查询。末节路由器会依靠转发路由器将正确的更新消息转发给所有的对等体路由器。

在下图中，我们将交换机 B 配置为了一台 EIGRPv6 末节路由器。交换机 A 和交换机 C 都与 WAN 相连。交换机 B 会将直连路由、静态路由、重分布路由和汇总路由通告给交换机 A 和交换机 C。交换机 B 并不会将任何通过交换机 A 学习到的路由通告出去（反之亦然）。

图 8：EIGRP 末节路由器配置

Routed to WAN	路由到 WAN
Switch A	交换机 A
Switch B	交换机 B

Switch C	交换机 C
Host A	主机 A
Host B	主机 B
Host C	主机 C

要想进一步了解关于 IPv6 末节路由的信息，可以参阅“Inspur INOS IP 配置指南 卷 2：路由协议，第 12.4 版”中的“实施 IPv6 EIGRP”一章。

基于 IPv6 的 SNMP 和系统日志

要想让网络同时支持 IPv4 和 IPv6 流量，IPv6 网络管理需要能够同时使用 IPv6 和 IPv4 实现流量传输。基于 IPv6 的系统日志支持对地址数据类型进行传输。

基于 IPv6 的 SNMP 和系统日志可以提供下列特性：

- 同时支持 IPv4 和 IPv6；
- 可以通过 IPv6 传输 SNMP 流量，并且对 SNMP 代理进行修改，以支持向 IPv6 主机发送 trap；
- 可以让与 SNMP 和系统日志相关的 MIB 支持 IPv6 的编址方式；
- 可以将 IPv6 配置为 trap 的接收方。

为了能够支持通过 IPv6 传输流量，SNMP 对当前的 IP 传输映射关系进行修改，使其能够同时支持 IPv4 和 IPv6。下列 SNMP 操作支持 IPv6 传输管理：

- 开放默认设置的用户数据报协议（UDP）SNMP 套接字；
- 提供一种新的、称为 *SR_IPv6_传输*的传输机制；
- 通过 IPv6 发送 SNMP 通告消息；
- 支持针对 IPv6 传输使用 SNMP 命名的访问控制列表；
- 支持使用 IPv6 传输的 SNMP 代理转发；
- 查看 SNMP 管理器特性可以兼容 IPv6 传输

要想进一步了解关于基于 IPv6 的 SNMP，包括其配置步骤，可以访问 icntnetworks.com，参阅“Inspur INOS IPv6 配置库”中的“管理基于 IPv6 的 Inspur INOS 应用”一章。

要想进一步了解关于基于 IPv6 的系统日志，包括其配置步骤，可以访问 icntnetworks.com，参阅“Inspur INOS IPv6 配置库”中的“实施 IPv6 编址和基本的连通性”一章。

基于 IPv6 的 HTTP

HTTP 客户端会向 IPv4 和 IPv6 HTTP 服务器发送请求消息，而服务器则会响应 IPv4 和 IPv6 HTTP 客户端发送的请求消息。包含 IPv6 地址的 URL 必须用每 16 位二进制就用冒号分隔的十六进制表示法表示。

Accept 套接字在调用时会选择 IPv4 或 IPv6 地址族。这里的 accept 套接字可以是一个 IPv4 套接字或者 IPv6 套接字。监听套接字会继续监听指示连接的 IPv4 和 IPv6 信号。IPv6 监听套接字会绑定一个 IPv6 通配符地址。

底层的 TCP/IP 协议栈支持双栈环境。HTTP 依靠 HTTP 协议栈和套接字来处理网路层的交互信息。

在建立 HTTP 连接之前，客户端和服务器之间必须建立了基本的网络连接（**ping**）。

要想进一步了解相关信息，可以访问 icntnetworks.com，参阅“Inspur INOS IPv6 配置库”中的“管理基于 IPv6 的 Inspur INOS 应用”一章。

不支持的 IPv6 单播路由特性

交换机不支持下列 IPv6 特性：

- IPv6 虚拟专用网络（VPN）路由器与转发（VRP）表；
- 转发以站点本地地址为目的的 IPv6 数据包；
- 隧道协议，如 IPv4-IPv6 隧道，或 IPv6-IPv4 隧道

- 用交换机充当 IPv4-IPv6 隧道，或 IPv6-IPv4 隧道协议的隧道端点；
- IPv6 单播逆向路径转发；
- IPv6 Web 缓存通信协议（WCCP）

IPv6 特性的限制

鉴于 IPv6 是在交换机硬件中实施的，硬件内存中的 IPv6 压缩地址存在一些限制。这些硬件限制导致有些特性的功能会受到影响，因此这些特性的使用也会受到限制：

这些限制包括：

- 交换机不能在硬件中转发 SNMP 封装的 IPv6 数据包。这类数据包只能在软件中转发；
- 交换机不能在硬件中对根据源地址路由的 IPv6 数据包应用 QoS 分类（classification）。

IPv6 与交换机堆栈

交换机支持以堆栈的形式执行 IPv6 转发，堆栈的主交换机也支持 IPv6 主机功能。堆栈主交换机可以运行 IPv6 单播路由协议，并且计算路由表。它们会接收路由表，并且创建硬件的 IPv6 路由以便对数据包进行转发。堆栈主交换机也可以运行所有的 IPv6 应用。

注释： 要在堆栈中路由 IPv6 数据包，那么堆栈中的所有交换机都要运行 IP Base 特性集。

如果一台新的交换机成为了堆栈的主设备，那么它就会重新计算 IPv6 路由表，并且将路由表分发给其他成员交换机。在堆栈选举出了新的主设备并且重启时，交换机堆栈并不会转发 IPv6 数据包。此时堆栈的 MAC 地址会发送变化，这也会导致 IPv6 地址出现变化。在通过接口配置命令 **ipv6 address** *ipv6-prefix/prefix length* **eui-64** 用扩展为标识符（EUI）设置堆栈 IPv6 地址时，这个接口的 IPv6 地址就是基于接口 MAC 地址生成的，详见的配置 IPv6 地址与启用 IPv6 路由。

如果用户在堆栈上配置了永久 MAC 地址特性，此时堆栈主设备发生了变化，那么堆栈 MAC 地址会有大约 4 分钟的时间不会变更。

下面是 IPv6 堆栈主设备和成员设备的功能：

- 堆栈主设备：
  - 运行 IPv6 路由协议；
  - 生成路由表；
  - 将路由表分发给使用 dCEFv6 的堆栈成员设备
  - 运行 IPv6 主机功能和 IPv6 应用。
- 堆栈成员设备（必须运行 IP Services 特性集）：
  - 接收堆栈主设备发来的 CEFv6 路由表；
  - 将路由条目写入硬件当中；

    注释： 在堆栈中，在数据包上没有携带 IPv6 扩展头部可选项，同时堆栈中的交换机也没有全部耗尽硬件资源的情况下，IPv6 数据包是在硬件中跨交换机进行路由转发的。
  - 在重新选举主设备时清空 CEFv6 表。

默认的 IPv6 配置

表 19：默认的 IPv6 配置

特性	**默认设置**
SDM 模版	高级桌面。默认为高级模版
IPv6 路由	全局禁用但在所有接口上启用
CEFv6 或 dCEFv6	禁用（IPv4 CEF 和 dCEF 默认是启用的） **注释：**在用户启用 IPv6 路由时，CEFv6 和

	dCEFv6 也会自动启用
IPv6 地址	未配置

配置 IPv6 地址与启用 IPv6 路由（CLI 界面配置方法）

在这一节中，我们会描述如何向一个三层接口分配 IPv6 地址，以及如何在交换机上全局转发 IPv6 流量。

用户在交换机上配置 IPv6 之前，应该考虑下面这些指导方针：

- 并不是本章中介绍的所有特性交换机都可以提供支持。详见不支持的 IPv6 单播路由特性；
- 在接口配置命令 **ipv6 address** 中，用户必须使用冒号分隔的十六进制这种格式来输入 *ipv6-address*（IPv6 地址）和 *ipv6-prefix*（IPv6 前缀）这两个变量。另外，*prefix-length*（前缀长度）这个变量（斜线/后面的参数）是一个十进制数，这个数代表了前缀是由前多少位连续的地址所组成的（也就是说，地址的网络位占多少位）。

要让一个接口转发 IPv6 流量，用户必须在这个接口上配置一个全局 IPv6 地址。在接口上配置 IPv6 地址之后，接口会自动配置上一个链路本地地址，同时这个接口会启用 IPv6 协议。

用户配置的这个接口会自动加入这条链路上必须加入的那些组播组，其中包括：

- 每个分配给该接口的单播地址，所对应的请求节点组播组 FF02:0:0:0:0:1:ff00::/104；
- 全节点链路本地组播组 FF02::1；
- 全路由器链路本地组播组 FF02::2。

要删除一个接口的 IPv6 地址，需要执行接口配置命令 **no ipv6 address** *ipv6-prefix/prefix length* **eui-64** 或 **no ipv6 address** *ipv6-address* **link-local**。要移除接口上所有手动配置的地址，可以直接在该接口的配置模式下输入命令 **no ipv6 address**，不带任何参数。要让一个尚未手动配置 IPv6 地址的接口停止处理 IPv6 流量，可以在接口配置模式下输入命令 **no ipv6 enable**。要在全局禁用 IPv6 路由，可以在全局配置模式下输入命令 **no ipv6 unicast-routing**。

要想进一步了解关于配置 IPv6 路由的信息，可以参阅 icntnetworks.com 中，“Inspur INOS IPv6 配置库”一文中的“实施 IPv6 编址和基本的连通性”一章。

用户可以从特权 EXEC 模式中，按照下面的步骤为一个三层接口分配 IPv6 地址，并启用 IPv6 路由转发：

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**sdm prefer dual-ipv4-and-ipv6** { **advanced** | **vlan**} 示例： `Device(config)#` **sdm prefer dual-ipv4-and-ipv6 default**	选择一个支持 IPv4 和 IPv6 的 SDM 模版。 • **advanced**：设置交换机，让其使用默认模版来平衡系统资源； • **vlan**：在不支持用硬件执行路由转发的交换机上最大化 VLAN 的配置。 **注释**：所有许可证版本都支持使用 advanced 这个参数。但只有 LAN Base 许可证的交换机支持使用 VLAN 模版这个参数。

步骤 3	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 4	**reload** 示例： `Device#` **reload**	重启操作系统
步骤 5	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 6	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	进入接口配置模式，指定要进行配置的三层接口。这个接口既可以是物理接口，也可以是交换机虚拟接口（SVI），或者三层 EtherChannel 接口
步骤 7	**no switchport** 示例： `Device(config-if)#` **no switchport**	清除这个接口的二层配置模式（如果这是一个物理接口的话）
步骤 8	配置下列命令之一： • **ipv6 address** *ipv6-prefix/prefix length* **eui-64** • **ipv6 address** *ipv6-address/prefix length* • **ipv6 address** *ipv6-address* **link-local** • **ipv6 enable** • **ipv6 address***WORD* • **ipv6 address***autoconfig* • **ipv6 address***dhcp* 示例： `Device(config-if)#` **ipv6 address 2001:0DB8:c18:1::/64 eui 64** `Device(config-if)#` **ipv6 address 2001:0DB8:c18:1::/64** `Device(config-if)#` **ipv6 address 2001:0DB8:c18:1:: link-local**	• 设置一个低 64 位为扩展唯一标识符（EUI）的全局 IPv6 地址。仅设置网络前缀；交换机会自动从自己的 MAC 地址中计算出后面的 64 位地址。 设置该地址会让该接口开始处理 IPv6 流量； • 在接口上设置一个链路本地地址，以替代在该接口启用 IPv6 时，接口自动配置的那个链路本地地址。执行这一设置会让该接口开始处理 IPv6 流量； • 在接口上自动配置 IPv6 链路本地地址，并让该接口开始处理 IPv6 流量。链路本地地址只能用来与连接在同一条链路上的节点进行通信。

	`Device(config-if)#` **ipv6 enable**	
步骤 9	**exit** 示例： `Device(config-if)#` **exit**	返回全局配置模式
步骤 10	**ip routing** 示例： `Device(config)#` **ip routing**	在这台交换机上启用 IP 路由转发功能
步骤 11	**ipv6 unicast-routing** 示例： `Device(config)#` **ipv6 unicast-routing**	启用对 IPv6 单播数据包的转发
步骤 12	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 13	**show ipv6 interface** *interface-id* 示例： `Device#` **show ipv6 interface gigabitethernet 1/0/1**	验证前面所作的设置
步骤 14	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 IPv4 与 IPv6 协议栈（CLI 界面配置方法）

用户可以从特权 EXEC 模式中，按照下面的步骤对一个三层接口进行配置，让该接口同时支持 IPv4 协议和 IPv6 协议，并且启用 IPv6 路由转发：

注释： 要让一个尚未手动配置 IPv6 地址的接口停止处理 IPv6 流量，可以在接口配置模式下输入命令 **no ipv6 enable**。

总步骤

1. configure terminal

2. ip routing

3. ipv6 unicast-routing

4. interface *interface-id*

5. no switchport

6. ip address *ip-address mask* [**secondary**]

7. Use one of the following:

• **ipv6 address** *ipv6-prefix/prefix length* **eui-64**

• **ipv6 address** *ipv6-address/prefix length*

• **ipv6 address** *ipv6-address* **link-local**

• **ipv6 enable**

• **ipv6 address***WORD*

• **ipv6 address***autoconfig*

• **ipv6 address***dhcp*

8. end

9. 配置下列命令之一：

• **show interface** *interface-id*

• **show ip interface** *interface-id*

• **show ipv6 interface** *interface-id*

10. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** 示例： `Switch#` **configure terminal**	进入全局配置模式
步骤 2	**ip routing** 示例： `Switch(config)#` **ip routing**	在这台交换机上启用 IP 路由转发功能
步骤 3	**ipv6 unicast-routing** 示例： `Switch(config)#` **ipv6 unicast-routing**	启用对 IPv6 单播数据包的转发
步骤 4	**interface** *interface-id* 示例： `Switch(config)#` **interface gigabitethernet 1/0/1**	进入接口配置模式，指定要进行配置的三层接口。
步骤 5	**no switchport** 示例： `Switch(config-if)#` **no switchport**	清除这个接口的二层配置模式（如果这是一个物理接口的话）
步骤 6	**ip address** *ip-address mask* [**secondary**] 示例：	为该接口设置主用或辅助 IPv4 地址

	`Switch(config-if)#` **ip address 10.1.2.3 255.255.255**	
步骤 7	配置下列命令之一： • **ipv6 address** *ipv6-prefix/prefix length* **eui-64** • **ipv6 address** *ipv6-address/prefix length* • **ipv6 address** *ipv6-address* **link-local** • **ipv6 enable** • **ipv6 address***WORD* • **ipv6 address***autoconfig* • **ipv6 address***dhcp* 示例： `Device(config-if)#` **ipv6 address 2001:0DB8:c18:1::/64 eui 64** `Device(config-if)#` **ipv6 address 2001:0DB8:c18:1::/64** `Device(config-if)#` **ipv6 address 2001:0DB8:c18:1:: link-local** `Device(config-if)#` **ipv6 enable**	• 设置一个低 64 位为扩展唯一标识符（EUI）的全局 IPv6 地址。仅设置网络前缀；交换机会自动从自己的 MAC 地址中计算出后面的 64 位地址； • 在接口上设置一个链路本地地址，以替代在该接口启用 IPv6 时，接口自动配置的那个链路本地地址； • 在接口上自动配置 IPv6 链路本地地址，并让该接口开始处理 IPv6 流量。链路本地地址只能用来与连接在同一条链路上的节点进行通信。 **注释**：要移除接口上所有手动配置的地址，可以直接在该接口的配置模式下输入命令 **no ipv6 address**，不带任何参数。
步骤 8	**end** 示例： `Switch(config)#` **end**	返回特权 EXEC 模式
步骤 9	使用下列命令之一： • **show interface** *interface-id* • **show ip interface** *interface-id* • **show ipv6 interface** *interface-id*	验证前面所作的设置
步骤 10	**copy running-config startup-config** 示例： `Switch#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置默认路由器优先级（CLI 界面配置方法）

在设备发送路由器通告消息中，消息中会携带用户通过接口配置命令 **ipv6 nd router-preference** 所配置的默认路由器优先级（DRP）。
当同一条链路上的两条路由器提供目的地相同而又不等价的路由时，DRP 可以告诉主机应该选择其中的哪台路由器。
要想进一步了解 IPv6 DRP，可以参阅 icntnetworks.com 中，“Inspur INOS IPv6 配置库”一文中的“实施 IPv6 编址和基本的连通性”一章。
用户可以从特权 EXEC 模式中，按照下面的步骤给路由器在一个接口上配置 DRP：
具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**ipv6 icmp error-interval** *interval* [*bucketsize*] 示例： `Device(config)#` **ipv6 icmp error-interval 50 20**	配置 IPv6 ICMP 错误消息间隔时间和令牌桶大小： • *interval*：即向令牌桶中添加令牌的时间间隔（单位为毫秒）。这个参数的取值范围是从 0 到 2147483647 毫秒。 • *bucketsize*：（可选配置）即桶中可以储存的最大令牌数。这个参数的取值范围是从 0 到 200。
步骤 3	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 4	**show ipv6 interface** [*interface-id*] 示例： `Device#` **show ipv6 interface gigabitethernet 1/0/1**	验证前面所作的设置
步骤 5	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 IPv6 ICMP 速率限制（CLI 界面配置方法）

设备默认就会启用 ICMP 速率限制特性，此时设备采用的错误消息默认间隔时间为 100 毫秒，

令牌桶大小（即桶中最多可以储存多少个令牌）为 10。
用户可以从特权 EXEC 模式中，按照下面的步骤来修改 ICMP 速率限制参数：
具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	进入接口配置模式，指定要设置 DRP 的三层接口。
步骤 3	**ipv6 nd router-preference** {**high** | **medium** | **low**} 示例： `Device(config-if)#` **ipv6 nd router-preference medium**	在这个交换机接口上设置 DRP
步骤 4	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 5	**show ipv6 interface** 示例： `Device#` **show ipv6 interface**	验证前面所作的设置
步骤 6	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 IPv6 CEF 和 dCEF

Inspur 快速转发是一种三层 IP 交换技术，其作用是可以增强网络的性能。CEF 会实施一种高级 IP 查找和转发算法，来提供最强大的三层交换性能。这种技术比快速交换路由缓存占用的 CPU 资源更是，可以让 CPU 有更多的处理资源为数据包提供转发。在交换机堆栈中，硬件会在堆栈中使用分布式 CEF（dCEF）。IPv4 CEF 和 dCEF 在默认状态下就是启用的。但 IPv6 CEF 和 dCEF 则默认是禁用的，但是在用户配置 IPv6 路由时就会自动启用。
当 IPv6 路由配置被删除时，IPv6 CEF 和 dCEF 也会被自动禁用。IPv6 CEF 和 dCEF 不能通过配置来禁用。用户可以在特权 EXEC 模式下输入命令 **show ipv6 cef** 来查看 IPv6 的状态。

要想让交换机路由 IPv6 单播数据包，用户必须使用全局配置命令 **ipv6 unicast-routing** 让路由器对 IPv6 单播数据包提供转发，此外用户也必须在接口配置模式下使用命令 **ipv6 address** 来给接口配置 IPv6 地址并且让接口启用 IPv6 流量处理功能。
要想进一步了解关于配置 CEF 和 dCEF 的信息，可以参阅 icntnetworks.com 中的“Inspur INOS IPv6 配置库”一文。

配置静态 IPv6 路由（CLI 界面配置方法）

在配置 IPv6 静态路由之前，用户必须首先使用全局配置命令 **ip routing** 来启用路由功能，使用全局配置命令 **ipv6 unicast-routing** 来启用 IPv6 数据包转发功能，并且至少给一个三层接口配置 IPv6 地址。
要想进一步了解关于配置静态 IPv6 路由的信息，可以参阅 icntnetworks.com 中，“Inspur INOS IPv6 配置库”一文中的“实施 IPv6 静态路由”一章。

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**ipv6 route** *ipv6-prefix/prefix length* {*ipv6-address* | *interface-id* [*ipv6-address*]} [*administrative distance*] 示例： `Device(config)#` **ipv6 route 2001:0DB8::/32 gigabitethernet2/0/1 130**	配置一条静态 IPv6 路由： • *ipv6-prefix*：静态路由的目的 IPv6 网络。如果配置的是静态主机路由，那么也可以在这里配置主机名； • */prefix length*：IPv6 前缀的长度。这个变量是一个十进制数，代表前缀是由前多少位连续的地址所组成的（也就是说，地址的网络位占多少位）。前缀长度前面必须要加斜线； • *ipv6-address*：可以到达指定网络的下一跳 IPv6 地址。下一跳 IPv6 地址不一定要是直连地址；设备可以通过递归的方式查找到直连的 IPv6 地址。在这里，地址必须按照 RFC 2373 定义的格式来输入，及输入用英文冒号分隔的十六进制数，每组长度为 16 位二进制； • *interface-id*：指定去往静态路由目的网络的点到点接口和广播接口。如果在这里指定的是点到点接口，那就可以不必配置下一跳 IPv6 地址。如果在这里指定的是广播接口，那就一定要指定下一跳 IPv6 地址，或者当目的网络就在这条链路上时，将链路本地地址指定为下一跳。用户可以将接收这个数据包的那个接口的地址设置为下一

		跳 IPv6 地址。 **注释：**在用链路本地地址作为下一跳时，用户必须指定一个 *interface-id*。 • *administrative distance*：（可选）即管理距离。取值范围为 1 到 254，默认值为 1，静态路由优先级优于直连路由之外的各类路由。要配置浮动静态路由，可以在这里配置一个大于动态路由协议的参数。
步骤 3	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 4	配置下列命令之一： • **show ipv6 static** [*ipv6-address* | *ipv6-prefix/prefix length*] [**interface** *interface-id*] [**detail**]][**recursive**] [**detail**] • **show ipv6 route static** [*updated*] 示例： `Device#` **show ipv6 static 2001:0DB8::/32 interface gigabitethernet2/0/1** 或 `Device#` **show ipv6 route static**	通过查看 IPv6 路由表中的条目来验证前面所作的设置： • **interface** *interface-id*：（可选）显示以用户所指定的接口作为出站接口的那部分静态路由； • **recursive**：（可选）仅显示递归静态路由。在这条命令中，关键字 **recursive** 和关键字 **interface** 是互斥的，但无论命令语法中是否包含了 IPv6 前缀，用户都可以使用 recursive 这个关键字； • **detail**：（可选）显示下列信息： ▪ 对于有效的递归路由，显示路径集和最大解析深度； ▪ 对于失效的递归路由，显示这条路由失效的原因。
步骤 5	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 IPv6 RIP（CLI 界面配置方法）

在配置交换机，让其运行 IPv6 RIP 之前，用户必须首先使用全局配置命令 **ip routing** 来启用路由功能，使用全局配置命令 **ipv6 unicast-routing** 来启用 IPv6 数据包转发功能，并且给要运行 RIP 的那个三层接口配置 IPv6 地址。

要想进一步了解关于配置 IPv6 RIP 路由的信息，可以参阅 icntnetworks.com 中，“Inspur INOS IPv6 配置库”一文中的“实施 IPv6 RIP”一章。

具体步骤

	命令或操作	**目的**

步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**ipv6 router rip** *name* 示例： `Device(config)#` **ipv6 router rip inspur**	配置一个 IPv6 RIP 进程，并进入该进程的路由器配置模式
步骤 3	**maximum-paths** *number-paths* 示例： `Device(config-router)#` **maximum-paths 6**	（可选）定义 IPv6 RIP 可以支持的最大等价路由数量。这个参数的取值范围是从 1 到 32，默认值为 16
步骤 4	**exit** 示例： `Device(config-router)#` **exit**	返回全局配置模式。
步骤 5	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	进入接口配置模式，指定要进行配置的三层接口
步骤 6	**ipv6 rip** *name* **enable** 示例： `Device(config-if)#` **ipv6 rip inspur enable**	在这个接口下启用并设置 IPv6 RIP 路由进程
步骤 7	**ipv6 rip** *name* **default-information** {**only** | **originate**} 示例： `Device(config-if)#` **ipv6 rip inspur default-information only**	（可选）让该接口在发送 RIP 路由进程更新时一并发送 IPv6 默认路由（::/0）。 **注释：**为了避免接口因发布 IPv6 默认路由（::/0）而产生路由环路，路由进程会忽略所有接口接收到的全部默认路由。 • **only**：选择加入默认路由，同时抑制该接口更新消息中发送的所有其他路由； • **originate**：在该接口发送的更新消息中加入默认路由与其他路由一起发送。
步骤 8	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 9	配置下列命令之一： • **show ipv6 rip** [*name*]	• 显示关于当前 IPv6 RIP 进程的信息； • 显示 IPv6 路由表中当前包含的内容。

	[**interface***interface-id*] [**database**] [**next-hops**] • **show ipv6 rip** 示例： `Device#` **show ipv6 rip inspur interface gigabitethernet2/0/1** 或 `Device#` **show ipv6 rip**	
步骤 10	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 IPv6 OSPF（CLI 界面配置方法）

用户可以自定义自己网络中的 IPv6 OSPF。但 IPv6 OSPF 的默认设置可以满足大多数客户和特性的使用需求。

用户可以参考下面的指导方针：

- 在修改 IPv6 命令的默认设置时务请谨慎。修改默认设置可能会对 OSPF 在 IPv6 网络中的工作产生不良影响；
- 在配置交换机，让其运行 IPv6 OSPF 之前，用户必须首先使用全局配置命令 **ip routing** 来启用路由功能，使用全局配置命令 **ipv6 unicast-routing** 来启用 IPv6 数据包转发功能，并且给要运行 OSPF 的那个三层接口配置 IPv6 地址。

要想进一步了解关于配置 IPv6 OSPF 路由的信息，可以参阅 icntnetworks.com 中，“Inspur INOS IPv6 配置库”一文中的“实施 IPv6 OSPF”一章。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**ipv6 router ospf** *process-id* 示例： `Device(config)#` **ipv6 router ospf 21**	启用这个进程的 OSPF 路由器配置模式。进程 ID 是管理员在启用 IPv6 OSPF 路由进程时分配的一个参数。这个参数只具有本地意义，用户可以选择 1 到 65535 之间的任意正整数
步骤 3	**area** *area-id* **range** {*ipv6-prefix/prefix length*} [**advertise** | **not-advertise**] [**cost** *cost*]	（可选）在区域边界合并和汇总路由。 • *area-id*：区域标识符，指定哪个区域的路由要进行汇总。这个值既可以设置为一个十进制数，也可以设置为一个 IPv6 前缀；

	示例： `Device(config)#` **area .3 range 2001:0DB8::/32 not-advertise**	• *ipv6-prefix/prefix length*：这个变量是一个十进制数，代表前缀是由前多少位连续的地址所组成的（也就是说，地址的网络位占多少位）。前缀长度前面必须要加斜线（/）； • **advertise**：（可选）设置地址范围状态，让设备通告并生成一个类型 3 汇总 LSA（链路状态通告） • **not-advertise**：（可选）将地址范围状态设置为 DoNotAdvertise（不要通告）。设备会抑制类型 3 汇总 LSA，这个网络会对外部网络隐藏自己的详细信息。 • **cost** *cost*：（可选）设置这条汇总路由开销值或代价值，OSPF 在执行 SPF 计算时会用这个数值来判断去往目的网络的最短路径。这个值的取值范围是从 0 到 16777215。
步骤 4	**maximum-paths** *number-paths* 示例： `Device(config)#` **maximum paths 16**	（可选）定义 IPv6 OSPF 可以支持的最大等价路由数量。这个参数的取值范围是从 1 到 32，默认值为 16
步骤 5	**exit** 示例： `Device(config-router)#` **exit**	返回全局配置模式。
步骤 6	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	进入接口配置模式，指定要进行配置的三层接口
步骤 7	**ipv6 ospf** *process-id* **area** *area-id* [**instance** *instance-id*] 示例： `Device(config-if)#` **ipv6 ospf 21 area .3**	在这个接口上启用 IPv6 OSPF。 **instance** *instance*：（可选）实例的标识符
步骤 8	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 9	配置下列命令之一： • **show ipv6 ospf** [*process-id*] [*area-id*] **interface** [*interface-id*]	• 显示关于 OSPF 接口的信息； • 显示关于 OSPF 路由进程的总的信息。

	• **show ipv6 ospf** [*process-id*] [*area-id*] 示例： `Device#` **show ipv6 ospf 21 interface gigabitethernet2/0/1** 或 `Device#` **show ipv6 ospf 21**	
步骤 10	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 IPv6 EIGRP

在配置交换机，让其运行 IPv6 EIGRP 之前，首先要输入命令 **ip routing global configuration** 来启用路由功能，使用配置命令 **ipv6 unicast-routing global** 来启用 IPv6 数据包转发功能，并且在要运行 IPv6 EIGRP 的那个三层接口上启用 IPv6 协议。

要手动设置路由器 ID，可以使用命令来查看当前配置的路由器 ID，然后使用命令 **router-id** 进行配置。

用户可以使用 EIGRPv6 来设置 EIGRP IPv6 接口，并且将其中一些接口设置为被动接口。用户可以使用命令 **passive-interface** 将一个接口设置为被动接口，然后在根据需要在一些接口上使用命令 **no passive-interface** 让这些接口恢复为主动接口。用户不必在被动接口上配置 EIGRP IPv6。

要想进一步了解关于配置 IPv6 RIP 路由的信息，可以参阅 icntnetworks.com 中，“Inspur INOS IPv6 配置库”一文中的“实施 IPv6 EIGRP”一章。

配置 IPv6 单播逆向路由转发

单播逆向路径转发（单播 RPF）特性可以缓解因攻击者将伪造了 IP 源地址的（欺骗）数据包注入到网络当中，导致网络丢弃拥有正确 IP 源地址数据包的情形。例如，很多类型的拒绝服务（DoS）攻击——包括 Smurf 和 Tribal Flood Network（TFN）发起的 DoS 攻击都会利用伪造的或不停变化的源 IP 地址，来防止人们找出并过滤掉它们发起的攻击数据。对于为公众提供网络接入服务的互联网服务提供商来说，单播 RPF 特性可以让设备只转发那些源地址有效且与 IP 路由表中一致的数据包，以此削弱这类攻击造成的危害。这种方法可以对 ISP、客户端和互联网的其他区域构成保护。

注释：

- 只有 IP Services 镜像支持单播 RPF；
- 如果交换机被部署在了一个混合的硬件堆栈中，这个堆栈由多种交换机型号组成，那么请不要配置单播 RPF。

要想进一步了解关于配置 IP 单播 RPF 的信息，可以参阅“Inspur INOS 安全配置指南 第 12.4

版”中的“其他安全特性”一章。

查看 IPv6

要想了解这些命令的完整语法结构及用法，可以参阅 Inspur INOS 命令参考手册。

表 20：监控 IPv6 的相关命令

命令	目的
show ipv6 access-list	显示汇总的访问控制列表
show ipv6 cef	显示 IPv6 Inspur 快速转发
show ipv6 interface*interface-id*	显示 IPv6 接口状态与配置
show ipv6 mtu	显示各个目的缓存的 IPv6 MTU 值
show ipv6 neighbors	显示 IPv6 邻居缓存条目
show ipv6 ospf	显示 IPv6 OSPF 信息
show ipv6 prefix-list	显示 IPv6 前缀列表
show ipv6 protocols	显示这台交换机上使用的 IPv6 路由协议列表
show ipv6 rip	显示 IPv6 RIP 路由协议的状态
show ipv6 route	显示 IPv6 路由表条目
show ipv6 routers	显示本地 IPv6 路由器
show ipv6 static	显示 IPv6 静态路由
show ipv6 traffic	显示 IPv6 流量的统计数据

表 21：显示 EIGRP IPv6 信息的命令

命令	目的
show ipv6 eigrp [*as-number*] *interface*	显示配置了 IPv6 EIGRP 的接口的相关信息
show ipv6 eigrp [*as-number*] *neighbor*	显示通过 IPv6 EIGRP 发现的邻居
show ipv6 interface[*as-number*] *traffic*	显示发送和接收的 IPv6 EIGRP 数据包数量
show ipv6 eigrptopology [*as-number* | *ipv6-address*][**active** | **all-links** | **detail-links** | **pending** | **summary** | **zero-successors** | **Base**]	显示 IPv6 拓扑表中的 EIGRP 条目

配置 DHCP 来分配 IPv6 地址

DHCPv6 地址分配的默认配置

在默认情况下，交换机上没有配置任何 DHCPv6 特性。

DHCPv6 地址分配的配置指导方针

在配置 DHCPv6 地址分配时，可以考虑下面的指导方针：

- 在这个配置过程中，用户指定的接口必须是下列三层地址之一：
  - 必须在一个三层接口上启用 DHCPv6 IP 路由；

- SVI：使用命令 **interface vlan** *vlan_id* 创建的 VLAN 接口；
- 三层模式下的 EtherChannel 接口：使用命令 **interface port-channel port-channel-number** 创建的 port-channel 逻辑接口；

- 交换机可以充当 DHCPv6 客户端、服务器或者中继代理。一个接口不能同时充当 DHCPv6 客户端、服务器或中继代理；
- 只有主交换机可以运行 DHCPv6 客户端、服务器或中继代理。如果堆栈重新选举主设备，那么新的主交换机就会继任 DHCPv6 的配置，但它无法集成那些保存在本地 RAM 中的 DHCP 服务器数据库租期信息。

启用 DHCPv6 服务器功能（CLI 界面配置方法）

用户可以在 DHCP 地址池配置模式下使用 **no** 形式的命令来修改 DHCPv6 地址池特征。要想在一个接口上禁用 DHCPv6 服务器功能，可以使用接口配置命令 **no ipv6 dhcp server** 来实现。用户可以从特权 EXEC 模式中，按照下面的步骤在接口上启用 DHCPv6 服务器功能：

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 2	**ipv6 dhcp pool** *poolname* 示例： `Device(config)# ipv6 dhcp pool 7`	进入 DHCP 地址池配置模式，并且给这个 IPv6 DHCP 池定义一个名称。地址池的名称既可以设置为字符串（如 Engineering），也可以设置为一个整数（如 0）。
步骤 3	**address prefix** *IPv6-prefix* {**lifetime**} {*t1 t1* | **infinite**} 示例： `Device(config-dhcpv6)# address prefix 2001:1000::0/64 lifetime 3600`	（可选）设置可供分配的地址前缀。 这个地址必须用冒号分隔的十六进制这种格式进行配置。 **lifetime** *t1 t1*：设置 IPv6 地址前缀保存在有效状态下的时间间隔（单位为秒）。这个参数的取值范围是从 5 到 4294967295 秒。若设置 **infinite** 则表示没有时间间隔。
步骤 4	**link-address** *IPv6-prefix* 示例： `Device(config-dhcpv6)# link-address 2001:1002::0/64`	（可选）设置一个链路地址 IPv6 前缀。 当入站接口的地址或者数据包的链路地址与用户设置的 IPv6 前缀相匹配，服务器就会使用配置的信息池。 这个地址必须用冒号分隔的十六进制这种格式进行配置。
步骤 5	**vendor-specific** *vendor-id* 示例： `Device(config-dhcpv6)#`	（可选）进入特定厂商的配置模式，并且给指定一个特定厂商标识数。这个数字是厂商的 IANA 私有企业编号。取值范围是从 1 到 4294967295。

	vendor-specific 9	
步骤 6	**suboption** *number* {**address** *IPv6-address* | **ascii** *ASCII-string* | **hex** *hex-string*} 示例： `Device(config-dhcpv6-vs)#` **suboption 1 address 1000:235D::**	（可选）输入厂商的自选项编号。取值范围是从 1 到 65535。用户可以在这里输入一个 IPv6 地址、一个 ASCII 文本，或者一个由子选项参数定义的十六进制字符串
步骤 7	**exit** 示例： `Device(config-dhcpv6-vs)#` **exit**	返回 DHCP 地址池配置模式。
步骤 8	**exit** 示例： `Device(config-dhcpv6)#` **exit**	返回全局配置模式。
步骤 9	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	进入接口配置模式，指定要进行配置的接口
步骤 10	**ipv6 dhcp server** [*poolname* | **automatic**] [**rapid-commit**] [**preference** *value*] [**allow-hint**] 示例： `Device(config-if)#` **ipv6 dhcp server automatic**	在接口上启用 DHCPv6 服务器功能： • *poolname*：（可选）用户给 IPv6 DHCP 定义的名称。地址池名称既可以是一个字符串（如 Engineering），也可以是一个整数（如 0）； • **automatic**：（可选）让系统自动判断在为客户端分配地址时使用哪个地址池； • **rapid-commit**：（可选）使用两次消息交互的方式； • **preference** *value*：（可选）配置服务器发送的通告消息中，优先级可选项中携带的优先级值。优先级值的取值范围是从 0 到 255，默认值为 0； • **allow-hint**：（可选）指定是否让服务器考虑客户端在 SOLICIT 消息中提出的建议。在默认情况下，服务器会忽略客户端的这种暗示。
步骤 11	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 12	配置下列命令之一：	• 显示 DHCPv6 地址池的配置；

	• **show ipv6 dhcp pool** • **show ipv6 dhcp interface** 示例： `Device#` **show ipv6 dhcp pool** 或 `Device#` **show ipv6 dhcp interface**	• 查看接口上上是否启用了 DHCPv6 服务器功能
步骤 13	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

启用 DHCPv6 客户端功能（CLI 界面配置方法）

我们接下来介绍如何在一个接口上启用 DHCPv6 客户端。

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	进入接口配置模式，指定要进行配置的接口
步骤 3	**ipv6 address dhcp** [**rapid-commit**] 示例： `Device(config-if)#` **ipv6 address dhcp rapid-commit**	让接口从 DHCPv6 服务器那里获取 IPv6 地址 **rapid-commit**：（可选）使用两次消息交互的方式分配地址
步骤 4	**ipv6 dhcp client request** [**vendor-specific**] 示例： `Device(config-if)#` **ipv6 dhcp client request vendor-specific**	（可选）让接口请求特定厂商的可选项
步骤 5	**end**	返回特权 EXEC 模式

	示例： `Device(config)#` **end**	
步骤 6	**show ipv6 dhcp interface** 示例： `Device#` **show ipv6 dhcp interface**	查看接口上上是否启用了 DHCPv6 客户端功能

配置 IPv6 单播路由的示例

配置 IPv6 地址并启用 IPv6 路由转发：示例

这个示例显示了如何对链路本地地址和基于 IPv6 前缀 2001:0DB8:c18:1::/64 的全局地址启用 IPv6。其中 EUI-64 为接口 ID 会用来作为这两个地址的后 64 位地址。在这个示例中，我们也提供了 EXEC 命令 **show ipv6 interface** 的输出信息，以说明接口 ID（20B:46FF:FE2F:D940）是如何添加在接口链路本地前缀（FE80::/64）后面的。

```
Device(config)# ipv6 unicast-routing
Device(config)# interface gigabitethernet1/0/11
Device(config-if)# no switchport
Device(config-if)# ipv6 address 2001:0DB8:c18:1::/64 eui 64
Device(config-if)# end
Device# show ipv6 interface gigabitethernet1/0/11
GigabitEthernet1/0/11 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::20B:46FF:FE2F:D940
Global unicast address(es):
2001:0DB8:c18:1:20B:46FF:FE2F:D940, subnet is 2001:0DB8:c18:1::/64 [EUI]
Joined group address(es):
FF02::1
FF02::2
FF02::1:FF2F:D940
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
```

配置默认路由器优先级：示例

这个示例显示了如何在一个接口上，给一台路由器配置一个高（*high*）DRP。

```
Device# configure terminal
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ipv6 nd router-preference high
Device(config-if)# end
```

配置 IPv4 与 IPv6 协议栈：示例

这个示例显示了如何在一个接口上启用 IPv4 和 IPv6 路由。

```
Device(config)# ip routing
Device(config)# ipv6 unicast-routing
Device(config)# interface fastethernet1/0/11
Device(config-if)# no switchport
Device(config-if)# ip address 192.168.99.1 255.255.255.0
Device(config-if)# ipv6 address 2001:0DB8:c18:1::/64 eui 64
Device(config-if)# end
```

启用 DHCPv6 服务器功能：示例

这个示例显示了如何用 IPv6 地址前缀配置一个名为 *engineering* 的池：

```
Device# configure terminal
Device(config)# ipv6 dhcp pool engineering
Device(config-dhcpv6)#address prefix 2001:1000::0/64
Device(config-dhcpv6)# end
```

这个示例显示了如何用 3 个链路地址和 1 个 IPv6 地址前缀来配置一个名为 *testgroup* 的池：

```
Device# configure terminal
Device(config)# ipv6 dhcp pool testgroup
Device(config-dhcpv6)# link-address 2001:1001::0/64
Device(config-dhcpv6)# link-address 2001:1002::0/64
Device(config-dhcpv6)# link-address 2001:2000::0/48
Device(config-dhcpv6)# address prefix 2001:1003::0/64
Device(config-dhcpv6)# end
```

这个示例显示了如何用厂商特定可选项配置一个名为 *350* 的池：

```
Device# configure terminal
Device(config)# ipv6 dhcp pool 350
Device(config-dhcpv6)# address prefix 2001:1005::0/48
Device(config-dhcpv6)# vendor-specific 9
Device(config-dhcpv6-vs)# suboption 1 address 1000:235D::1
Device(config-dhcpv6-vs)# suboption 2 ascii "IP-Phone"
Device(config-dhcpv6-vs)# end
```

启用 DHCPv6 客户端功能：示例

这个示例显示了如何让接口获取 IPv6 地址并启用 rapid-commit 可选项：

```
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# ipv6 address dhcp rapid-commit
```

配置 IPv6 ICMP 速率限制：示例

这个示例显示了如何将 IPv6 ICMP 错误消息显示的时间间隔设置为 50 毫秒，将令牌桶大小设置为 20 个令牌。

```
Device(config)#ipv6 icmp error-interval 50 20
```

配置 IPv6 静态路由：示例

这个示例显示了如何用出站接口配置一条浮动静态路由，并且将其管理距离设置为 130：

```
Device(config)# ipv6 route 2001:0DB8::/32 gigabitethernet2/0/1 130
```

配置 IPv6 RIP：示例

这个示例显示了如何启用 RIP 路由进程（将进程名设置为 *inspur*），并且将最大等价路由数量设置为 8 条，同时在接口上启用 RIP：

```
Device(config)# ipv6 router rip inspur
Device(config-router)# maximum-paths 8
Device(config)# exit
Device(config)# interface gigabitethernet2/0/11
Device(config-if)# ipv6 rip inspur enable
```

查看 IPv6：示例

这个示例显示了特权 EXEC 命令 **show ipv6 interface** 的输出信息：

```
Device# show ipv6 interface
Vlan1 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::20B:46FF:FE2F:D940
Global unicast address(es):
3FFE:C000:0:1:20B:46FF:FE2F:D940, subnet is 3FFE:C000:0:1::/64 [EUI]
Joined group address(es):
FF02::1
FF02::2
FF02::1:FF2F:D940
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
```

```
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
<output truncated>
```

实施 IPv6 组播

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于实施 IPv6 组播路由的信息

本章会描述如何在一台交换机上实施 IPv6 组播路由。
传统 IP 通信环境支持一台主机向另一台主机发送（单播传输的）数据包，或者向所有主机发送（广播传输的）数据包。IPv6 组播提供了第 3 种途径，它让一台主机可以将一组数据流同时发送一个所有主机中的一个子集（即传输给一个组）。

IPv6 组播概述

每个 IPv6 组播组都是一个由希望接收某组数据流的设备所组成的任意组。这种组没有物理或地理层面的边界，也就是说接收方可以位于互联网的任何位置，也可以位于任何私有网络当中。如果接收方对接收某个组的数据流感兴趣，它就必须向本地交换机发送信令，以加入

到这个组中。这个信令消息是通过 MLD 协议传输的。
交换机会使用 MLD 协议来学习其直连的子网中是否有组的成员设备。主机会通过发送 MLD 报告消息的方式加入组播组当中。接下来，交换机会在每个子网中发送组播数据的一个副本，以此将数据发送给大量的接收方。希望接收流量的 IPv6 主机称为组成员。
发送给组成员的数据包都会用同一个组播组地址进行标识。网络会采用尽力而为的传输方式，将组播数据包发送给组。这种方式与网络传输 IPv6 单播数据包的方式相同。
组播环境由发送方和接收方组成。任何主机都可以向一个组发送消息，无论该主机是不是组的成员，但只有组成员可以侦听并接收消息。
组播地址是为组播组的接收方选择的。发送方会用这个地址作为数据报的目的地址，将数据报发送给组的所有成员。
组播组的成员身份是动态的；主机可以随时加入和离开组播组。组播组对于成员的位置和数量是没有限制的。一台主机可以同时充当多个组播组的成员。
组播组的活动状态、它的存在时间和成员身份，这些都因组而异，因时而异。拥有成员的组播组也有可能没有活动。

IPv6 组播路由的实施

Inspur INOS 软件支持下列协议实施 IPv6 组播路由：

- IPv6 交换机会使用 MLD 在直连链路上发现组播侦听设备（即那些希望接收发往某个组播地址的数据帧的节点）。MLD 有 2 个版本：MLD 第 1 版是一个基于 IPv4 版 IGMP（互联网组管理协议）的协议，而 MLD 第 2 版则是基于 IPv4 的 IGMP 第 3 版开发的。Inspur INOS 软件会同时使用 MLD 第 2 版和第 1 版来实现 IPv6 组播。MLD 第 2 版完全可以向后兼容 MLD 第 1 版（定义在 RFC 2710 中）。因此，只支持 MLD 第 1 版的主机可以与运行 MLD 第 2 版的交换机之间实现互操作。同理，交换机也能够支持那些同时连接了 MLD 第 1 版主机和 MLD 第 2 版主机的混合 VLAN 环境。
- 交换机之间会使用 PIM-SM 来追踪需要将哪些组播数据包发送给对端及自己直连的局域网；
- 特定源组播模式的 PIM（PIM-SSM）与 PIM-SM 类似，只是 PIM-SSM 可以报告自己对接收从特定源地址（或者除某个特定源地址之外的所有其他地址）发往某个 IP 组播地址的数据包感兴趣。

IPv6 组播侦听者发现协议

要在园区网中实施组播传输，必须首先找出谁是组播的接收方。IPv6 交换机会使用 MLD 协议来发现直连链路上的组播接收方（比如那些希望接收到组播数据包的节点），同时发现这些邻居节点都对哪些组播地址感兴趣。这种协议的作用是发现本地组，以及特定源组的成员身份。
MLD 协议可以通过使用特殊组播查询方和主机，提供一种可以动态控制和限制网络中组播数据流的方式。

组播查询方与主机

组播查询方是指那些通过发送查询消息来发现哪些网络设备是某个组播组成员的网络设备（比如交换机）。
组播主机是组播的接收方（也包括交换机），这些交换机会通过发送报告消息来向查询方通告主机的成员身份。
从同一个源那里接收组播数据流的一系列查询方和主机就称为一个组播组。查询方和主机会使用 MLD 报告来加入和离开组播组，并开始接收组流量。

MLD 会用 ICMP（互联网控制消息协议）来携带自己的消息。所有 MLD 消息都是链路本地消息，这类消息的跳数限制都会被设置为 1，它们都设置了交换机告警可选项（switch alert option）。交换机告警可选项会显示出哪些消息实施了逐跳可选项（hop-by-hop option）头部。

MLD 访问组

在 Inspur INOS IPv6 组播交换机上，MLD 访问组可以为接收方提供访问控制功能。这种特性能够限制一台接收方设备可以加入的组播组列表，也可以选择允许或拒绝加入 SSM 信道所使用的源。

接收方显式追踪

显示追踪特性可以让一台交换机追踪 IPv6 网络中主机的行为。这个特性也可以让 MLD 第 2 版的主机报告能够运用快速离开机制。

协议独立组播

交换机之间会使用协议独立组播（PIM）来追踪要将哪些组播数据包转发给对方，要将哪些组播数据包转发到它们直连的局域网当中。PIM 是独立于单播路由协议工作的，但它也会像其它协议那样负责处理组播路由更新的收发。无论局域网中使用了哪种单播路由协议来建立单播路由表，Inspur INOS PIM 都会使用当前单播表中的内容来执行逆向路径转发（RPF）校验，而不会建立和维护自己独立的路由表。

用户可以通过配置 IPv6 组播，来使用 PIM-SM 或 PIM-SSM 两种操作方式之一，也可以在网络中同时使用 PIM-SM 和 PIM-SSM。

PIM 稀疏模式

IPv6 组播使用 PIM-SM 为域内组播路由提供支持。PIM-SM 会使用单播路由来提供建立组播树所需的逆向路径信息，但这并不依赖任何特定的单播路由协议。

PIM-SM 适用于那种每个组播组所参与的交换机数量不多，且这些交换机（在没有设备明确请求流量时）不会为一个组转发组播数据包的环境。PIM-SM 会在共享树中转发数据包，以此分发关于活动源的信息。PIM-SM 最初会使用共享树，而共享树中需要使用 RP。

请求消息是通过 PIM 加入消息一起发送的，而 PIM 加入消息是逐跳向着树的根节点进行发送的。在 PIM-SM 模式中，如果是共享树，那么树的根节点就是 RP；如果是最短路径树（SPT），那么树的跟节点就是与组播源直连的那台第一跳交换机。RP 会追踪组播组，而发送组播数据包的主机会在 RP 中进行注册。

当 PIM 加入消息到达树时，沿途的交换机就会建立组播转发状态，让被请求的组播流量能够沿着树向反方向转发。在不需要传输组播流量时，交换机就会沿着树向根节点发送一条 PIM 修剪消息，以修剪（移除）掉那些不必要的流量。在 PIM 修剪消息沿着树逐跳传输时，沿途的每台交换机都会对自己的转发状态进行相应地更新。最终，与某个组播组或源相关的转发状态也会被删除。

组播数据的发送方会发送去往一个组播组的数据。发送方的指定交换机（DR）在接收到这些数据时，会用单播的形式封装它们，然后直接将这些数据发送给 RP。RP 在接收到封装后的数据之后，会对数据进行解封装，然后将它们转发到共享树中。接下来，这些数据包就会沿着交换机中的(*, G)组播树状态，被复制到 RP 树的所有设备上，并且最终到达这个组播组的中每一个接收方。封装数据包并发送给 RP 的过程称为注册，封装的数据包则称为 PIM 注册数据包。

IPv6 BSR：配置 RP 映射

一个域中的 PIM 交换机必须能够将各个组播组映射到正确的 RP 地址。PIM-SM 的 BSR 协议为在域中快速分发组与 RP 的映射关系，提供了一种动态的、可自适应的机制。如果 RP 变为

不可达，那么通过 IPv6 BSR 特性，交换机可以检测到这个事件，此时设备会修改映射表，以便不再使用那个不可达的 RP，并且将新的表迅速分发到整个域中。

每个 PIM-SM 组播组都需要关联 RP 的 IP 或 IPv6 地址。当新的组播发送方开始发送数据时，其本地 DR 会将这些数据包封装在 PIM 注册消息中，并且将它们转发给那个组播组的 RP。当新的组播接收方设备加入时，其本地 DR 会向这个组播组的 RP 发送一条 PIM 加入消息。任何一台 PIM 交换机发送了一条(*, G)加入消息时，PIM 交换机都需要知道哪台交换机是 RP 方向的下一台交换机，于是 G（组）就可以向交换机发送一条消息。同样，当 PIM 交换机在使用(*, G)状态转发数据包时，PIM 交换机也需要知道哪个接口是去往 G 的数据包应该入站的接口，因为它也需要拒绝从其他接口进入的交换机的数据包。

域中的一小部分交换机会被配置为候选的引导程序交换机（C-BSR），域只会在其中选择一台作为 BSR。此外，域中也有一小部分交换机要配置为候选 RP（C-RP）；一般来说，这些交换机也就是配置为 C-BSR 的那些交换机。候选 RP 会周期性地向域中的 BSR 发送单播候选 RP 通告消息（C-RP-Adv），表示自己希望称为一台 RP。C-RP-Adv 消息中会包含通告 C-RP 的地址，和一个组地址与掩码长度字段的可选列表，显示向其通告了成员身份的组前缀。接下来，BSR 会在周期生成的引导程序消息中包含一系列这样的 C-RP，及它们对应的组前缀。BSM 会逐跳在域中分发。

双向 BSR 支持在 C-RP 消息中通告双向 RP，以及在 BSM 中通告双向范围。系统中的所有交换机都必须能够使用 BSM 中的双向范围；否则，双向 RP 特性就无法工作。

PIM-特定源组播

PIM-SSM 是一种支持 SSM 的路由协议，这个路由协议来自于 PIM-SM。不过，PIM-SSM 并不像 PIM-SM 那样，在有 PIM 加入消息时，会发送来自所有源的数据。SSM 特性只会将那些从接收方已经明确表示（显示）加入的组播源那里接收到数据流量发送给接收方，因此 SSM 可以优化带宽的利用率，防止恶意的互联网组播流量。此外，除了使用 RP 和共享树之外，SSM 还会使用通过组播组源地址中发现的信息。接收方会通过源地址，在 MLD 成员关系报告中，向最后一跳交换机提供相关信息，并以此建立去往源的最短路径树。

在 SSM 中，数据是基于(S, G)信道进行传输的。(S, G)信号的流量源地址为 IPv6 单播源地址 S，而其目的 IPv6 地址则是组播组地址 G。系统会通过加入(S, G)信道的成员来接收这些流量。这个过程中不需要使用信令，但接收方必须订阅或取消订阅(S, G)信道的方式，来接收或不再接收发送给某个源的流量。

要想让 SSM 正常工作，需要使用 MLD 第 2 版。MLD 可以让主机提供源信息。在 SSM 能够与 MLD 一起运行之前， Inspur INOS IPv6 交换机、运行应用的主机和应用本身都必须支持 SSM，SSM 才能正常工作。

可路由地址 Hello 选项

在使用 IPv6 内部网关协议来建立单播路由表时，交换机负责检测上游交换机地址的那个流程会假定 PIM 邻居的地址永远与下一跳交换机的地址相同，只要它们指的是同一台交换机。然而，如果交换机在一条链路上有多个地址时，情形往往并非如此。

在 IPv6 环境中，有两种典型的情形可能会导致这种情况。第一种情形是当单播路由表是由非 IPv6 内部网关协议（例如组播 BGP）建立的情形。第二种情形是 RP 的地址与下游交换机共享了一个子网前缀（要注意，RP 交换机地址必须是整个域范围内的，因此不能是链路本地地址）。

可路由地址 Hello 选项可以避免 PIM 协议出现这类情形，它会添加一个 PIM Hello 消息选项，其中包含了通告 PIM Hello 消息的接口上所有的地址。当 PIM 交换机为一些地址找到了一台上游交换机时，它就会将 RPF 的计算结果与选项中的地址和 PIM 邻居的地址进行比较。由于选项中包含了 PIM 交换机在这条链路上的所有地址，所以如果它指向的那台 PIM 交换机

也支持这个选项的话，那么这里面也会包含 RPF 计算的结果。
由于对 PIM 消息大小的限制，而且可路由地址 hello 选项必须携带在单个 PIM hello 消息中，所以接口上最多只能配置个 16 地址。

PIM IPv6 末节路由

PIM 末节路由特性可以将被路由的流量移动到距离终端用户更近的位置，以此介绍对网络资源的占用。
在使用 PIM 末节路由的网络中，唯一允许向用户转发 IPv6 流量的路由就是那些穿越配置了 PIM 末节路由的交换机，发往终端用户的路由条目。PIM 被动接口会连接到（像 VLAN 这样的）二层接入域或者连接其他二层设备的接口。只有直连的组播接收方和组播源可以处于二层接入域中。PIM 被动接口不会对接收到的 PIM 控制数据包进行发送或处理。
在使用 PIM 末节路由时，用户需要通过配置让转发路由器和远程路由器使用 IPv6 组播路由，同时只将这台交换机配置为 PIM 末节路由器。交换机不会路由转发穿越路由器的流量。用户也需要在交换机上配置一个路由模式的上行链路端口。这个交换机上行链路端口不能使用 SVI。
当用户在交换机上配置 PIM 末节路由时，也必须配置 EIGRP 末节路由。要想进一步了解相关信息，可以参阅 “EIGRPv6 末节路由”。
拓扑中不支持部署冗余 PIM 末节路由器。如果有多台 PIM 路由器向一个接入域转发组播流量，就表示网络中存在冗余拓扑。在 PIM 被动接口上，PIM 消息会被阻塞，PIM 被动接口不支持 PIM 断言报文（PIM Assert）和指定路由选举机制。PIM 末节特性只支持非冗余接入路由器拓扑。在非冗余拓扑中，PIM 被动接口会假定这个接入域中只有自己这个接口和指定路由器。
在下图中，路由器连接了交换机 A 的 25 号上行链路端口， VLAN 100 接口和主机 3 上启用了 PIM 末节路由。这种配置方法可以让直连主机能够从组播源那里接收流量。用户可以参阅 “配置 PIM IPv6 末节路由”来了解更多信息。

图 9：PIM 末节路由器配置

Source	源
Router	路由器
Switch A	交换机 A
Host 1	主机 1
Host 2	主机 2
Host 3	主机 3
Port 25	端口 25
Port 20	端口 20

静态组播路由（Mroute）

IPv6 静态组播路由的工作方式与用于 RPF 校验的 IPv4 静态组播路由别无二致。IPv6 静态组播路由会与 IPv6 静态路由共享同一个数据库，它对静态路由支持 RPF 校验的操作进行了扩展。静态组播路由支持等价多路径组播路由，也支持纯单播静态路由。

MRIB

组播路由信息库（MRIB）是一种独立于协议的组播路由条目数据库，这个数据库是通过组播

路由协议（路由客户端）实现的。这个数据库的主要功能是在路由协议和组播转发信息库（MFIB）间提供独立的数据。此外，MIB 也会可以充当客户端间的协调和通信点。
路由客户端会使用 MRIB 提供的服务，来生成路由条目，并且提取出其他客户端对路由条目所作的修改。除了路由客户端之外，MRIB 也有转发客户端（MFIB 实例）和特殊客户端（如 MLD）。MFIB 会从 MRIB 中提取出自己的转发条目，也会向 MRIB 通告与接收数据包有关的事件。这些通告既可以是由路由客户端显式请求的，也可以是由 MFIB 同步创建的。
MRIB 另一项重要的功能是协调多个路由客户端在同一个组播会话中建立组播联通性。MRIB 也可以实现 MLD 和路由协议之间的协调。

MFIB

MFIB 是一个独立于平台，也独立于路由协议的 IPv6 软件库。它的主要作用是是为 Inspur INOS 平台提供一个接口，以便系统在转发表变更时能够阅读 IPv6 组播转发表和组播通告消息。MFIB 提供的信息已经清晰地定义了转发的语义，让平台能够更轻松地将它们转换为自己的专用硬件或软件转发机制。
当网络中的路由或网络的拓扑发生变化时，设备就会更新 IPv6 路由表，这些更新也会反映在 MFIB 当中。MFIB 可以根据 IPv6 路由表中的信息来维护下一跳地址信息。由于 MFIB 条目和路由表条目之间存在一对一的对应关系，因此 MFIB 会包含所有已知的路由，同时 MFIB 也就不需要维护一个与交换路径相对应的路由缓存了（如快速交换或最优交换）。
MFIB
注释： 在堆栈环境中，主设备需要将 MFIB 信息分发给其他堆栈成员，此时分布式 MFIB 才体现出它的重要性。在下面一节中，“线卡”指的就是堆栈中的成员交换机。
MFIB 的作用是在分布式平台上交换机组播 IPv6 数据包。MFIB 也包含了特定平台的信息，这些信息会在线卡之间复制。所有转发环境中，都可以提供 MFIB 的基本操作，即实施核心转发功能。
MFIB 可以实施下列功能：

- 将线卡中因数据驱动而产生的协议事件转发给 PIM；
- 提供了一个 MFIB 应用程序接口（API），将 MFIB 的变更传送到编程硬件加速引擎的那个特定平台代码。这个 API 也提供了能够将数据包发送到软件中的接口（如果这个数据包触发了数据驱动事件的话），并且把有关流量的特定状态信息加载到硬件中。

将 MFIB 和 MRIB 这两个子系统结合起来使用，可以让交换机在每个线卡中都保存一个自定义的 MFIB 数据库副本，并且将 MFIB 相关的特定平台信息从 RP 传输给线卡。

IPv6 组播进程交换与快速交换

统一 MFIB 的作用是同时在 IPv6 组播环境中，为 PIM-SM 和 PIM-SSM 提供快速交换和进程交换。在进程交换中，INOS 守护进程必须对每个数据包进行检查、重写和转发。交换机会首先接收到数据包，然后将其复制到系统内存中。接下来，交互那几会查询路由表中的三层网络地址。接下来，它会用下一跳目的地址来重新封装二层数据帧，并且将其发送给出站接口。INOSd 也会计算循环冗余检验（CRC）。这种交换方式扩展性最差的 IPv6 数据包交换方式。
IPv6 组播快速交换机可以为交换机提供比进程交换更加理想的数据包转发性能。传统上那些存储在路由缓存中的信息会被存储到多个数据结构中，以备执行 IPv6 组播交换。这些数据接口可以优化查询过程，实现更加高效的数据包转发性能。

在 IPv6 组播转发环境中，如果 PIM 协议允许的话，那么交换机就会对第一个数据包执行快速转发。在 IPv6 组播快速转发环境中，设备会预先计算好 MAC 封装的头部。IPv6 组播快速交换会使用 MFIB 来判断如何基于 IPv6 目的前缀转发数据包。除了 MFIB 之外，IPv6 组播快速转发还会使用邻接表来预先计算二层的地址信息。而邻接表中会保存所有 MFIB 条目的二层下一跳地址。

在发现邻接设备时，设备就会创建出邻接表。每当设备创建一个邻接表条目（比如通过 ARP）时，它就会预先计算出这个邻接节点的链路层头部，并且将它一并存储在邻接表中。一旦设备判断出转发数据包的路由，它就会查看它的下一跳和对应的邻接条目。在后面交换数据包的过程中，这些信息会用来执行数据封装。

一条路由可能包含了很多通往目的前缀的路径，比如当用户给交换机配置了负载分担和冗余时就会出现这样的情况。对于每条解析出来的路径，设备都会给这条路径添加一个指针，指向下一跳接口对应的邻接设备。

将 IPv6 组播地址族用于多协议 BGP

将 IPv6 组播地址族用于多协议 BGP 这项特性可以让多协议 BGP 扩展到 IPv6 环境中，这项特性支持的功能与 IPv4 BGP 支持的功能类似。IPv6 增加了对组播 BGP 的支持，其中包括支持 IPv6 组播地址族，以及网路层可达性信息（NLRI）以及使用 IPv6 地址的下一跳（路径中去往目的的下一台交换机）属性。

组播 BGP 是一种增强型 BGP，它可以支持部署域间 IPv6 组播。多协议 BGP 会携带多个网络层协议地址族的路由信息，比如 IPv6 地址族和 IPv6 组播路由。IPv6 组播地址族包含了让 IPv6 PIM 协议执行 RPF 查找的路由，组播 BGP IPv6 也提供了域间传输。如果使用 BGP 传输组播流量的话，那么用户必须使用多协议 BGP 来实现 IPv6 组播传输，因为单播 BGP 学习到的路由不会用来转发 IPv6 组播流量。

组播 BGP 功能是通过一个独立的地址族提供的。随后的地址组标识符（SAFI）提供了关于属性中携带的，网络层可达性信息的类型信息。多协议 BGP 单播会使用 SAFI 1 消息，而多协议 BGP 组播则会使用 SAFI 2 消息。SAFI 1 消息会显示这条路由仅供 IP 单播使用，IP 组播无法使用。有鉴于此，在执行 IPv6 组播 RPF 查找时，IPv6 单播 RIB 中的那些 BGP 路由就一定会被忽略。

设备会维护一个独立的 BGP 路由器，这个 BGP 路由表会使用 IPv6 组播 RPF 查找来配置那些不一致策略和拓扑（例如 IPv6 单播和组播）。组播 RPF 查找与 IP 单播路由查找相当类似。

IPv6 组播 BGP 表不会关联 MRIB。但 IPv6 组播 BGP 会在必要时会依据单播 IPv6 RIB 进行操作。组播 BGP 不会向 IPv6 单播 RIB 中注入或者更新路由。

实施 IPv6 组播

启用 IPv6 组播路由

用户可以从特权 EXEC 模式中，执行下列步骤：

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal**	进入全局配置模式

	示例： `Device#` **configure terminal**	
步骤 2	**ipv6 multicast-routing** 示例： `Device (config)#` **ipv6 multicast-routing**	在所有启用了 IPv6 的接口上启用组播路由，同时在这台交换机所有开启的接口上给 PIM 和 MLD 启用组播转发。
步骤 3	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

自定义并验证 MLD 协议

在一个接口上自定义并验证 MLD
用户可以从特权 EXEC 模式中，执行下列步骤：
具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *type number* 示例： `(config)#` **interface GigabitEthernet 1/0/1**	指定接口类型和编号，并让交换机进入接口配置模式
步骤 3	**ipv6 mld join-group** [*group-address*] [**include** | **exclude**] {*source-address* | **source-list** [*acl*]} 示例： `(config-if) #` **ipv6 mld join-group FF04::10**	给指定的组和源配置 MLD 报告
步骤 4	**ipv6 mld access-group** *access-list-name* 示例： `(config-if) #` **ipv6 access-list**	允许所有用户执行 IPv6 组播接收方访问控制

	acc-grp-1	
步骤 5	**ipv6 mld static-group** [*group-address*] [**include** | **exclude**] {*source-address* | source-list [*acl*]} 示例： `(config-if) #` **ipv6 mld static-group ff04::10 include 100::1**	静态将这个组播组的流量转发给一个指定的接口，并且让该接口像连接了 MLD 加入设备那样工作
步骤 6	**ipv6 mld query-max-response-time** *seconds* 示例： `(config-if) #` **ipv6 mld query-max-response-time 20**	配置 MLD 查询消息中通告的最大响应时间
步骤 7	**ipv6 mld query-timeout** *seconds* 示例： `(config-if) #` **ipv6 mld query-timeout 130**	配置在交换机接替这个接口的查询方之前，等候的超时时间值
步骤 8	**exit** 示例： `(config-if) #` **exit**	连续两次输入这条命令，离开接口配置模式，并且进入特权 EXEC 模式
步骤 9	**show ipv6 mld groups [link-local]** [*group-name* | *group-address*] [*interface-type interface-number*] [**detail** | **explicit**] 示例： `#` **show ipv6 mld groups GigabitEthernet 1/0/1**	显示交换机直连的组播组，以及通过 MLD 学习到的组播组
步骤 10	**show ipv6 mld groups summary** 示例： `#` **show ipv6 mld groups summary**	显示 MLD 缓存中的(*, G)数量和(S, G)成员关系报告数量
步骤 11	**show ipv6 mld interface** [*type number*] 示例：	显示一个接口的组播相关信息

	# **show ipv6 mld interface GigabitEthernet 1/0/1**	
步骤 12	**debug ipv6 mld** [*group-name* | *group-address* | *interface-type*] 示例： # **show ipv6 mld interface GigabitEthernet 1/0/1**	启用 MLD 协议操作的相关调试信息
步骤 13	**debug ipv6 mld explicit** [*group-name* | *group-address* 示例： # **debug ipv6 mld explicit**	显示主机显式追踪的相关信息
步骤 14	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

实施 MLD 组限制

每个接口上的 MLD 限制与全局的 MLD 限制是相互独立操作的。同一台交换机上可以同时针对各个接口配置 MLD 限制，并且配置全局 MLD 限制。在默认情况下，设备上是没有配置 MLD 限制数量的（无论接口 MLD 限制还是全局 MLD 限制）。成员关系报告无论超出了接口限制，还是超出了全局状态限制都会被设备忽略。

总步骤

1. enable

2. configure terminal

3. ipv6 mld [*vrf vrf-name*] **state-limit** *number*

4. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： Device> **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： Device# **configure terminal**	进入全局配置模式
步骤 3	**ipv6 mld** [*vrf vrf-name*] **state-limit** *number* 示例： Device(config)# **ipv6 mld state-limit 300**	在全局限制 MLD 状态的数量
步骤 4	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

在接口实施 MLD 组限制

1. enable

2. configure terminal

3. interface type *number*

4. ipv6 mld limit *number* [**except**]*access-list*

5. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface type** *number* 示例： `Device(config)#` **interface GigabitEthernet 1/0/1**	指定接口类型和编号，并让交换机进入接口配置模式
步骤 4	**ipv6 mld limit** *number* [**except**]*access-list* 示例： `Device(config-if)#` **ipv6 mld limit 100**	在接口上限制 MLD 状态的数量
步骤 5	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置接收方显式追踪特性来追踪主机的行为

显示式追踪特性可以让交换机追踪 IPv6 网络中的主机行为，并且让快速离开机制能够与 MLDv2 主机报告一起使用。

用户可以从特权 EXEC 模式中，执行下列步骤：

具体步骤

	命令或操作	目的
步骤 1	**configure terminal**	进入全局配置模式
步骤 2	**interface type** *number* 示例： `Device(config)#` **interface GigabitEthernet 1/0/1**	指定接口类型和编号，并让交换机进入接口配置模式
步骤 3	**ipv6 mld explicit-tracking** *access-list-name* 示例：	启用主机显示追踪

	(config-if)# **ipv6 mld explicit-tracking list1**	
步骤 4	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

重置 MLD 流量计数器

用户可以从特权 EXEC 模式中，执行下列步骤：

具体步骤

	命令或操作	目的
步骤 1	**clear ipv6 mld traffic** 示例： # **clear ipv6 mld traffic**	重置所有 MLD 流量计数器
步骤 2	**show ipv6 mld traffic** 示例： # **show ipv6 mld traffic**	显示 MLD 流量计数器
步骤 3	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

清除 MLD 接口计数器

用户可以从特权 EXEC 模式中，执行下列步骤：

具体步骤

	命令或操作	目的
步骤 1	**clear ipv6 mld counters** *interface-type* 示例： # **clear ipv6 mld counters Ethernet1/0**	清除 MLD 接口计时器
步骤 2	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 PIM

在这一节中，我们会解释如何配置 PIM

给一个组范围配置 PIM-SM 并显示 PIM-SM 信息

用户可以从特权 EXEC 模式中，执行下列步骤：

具体步骤

	命令或操作	目的
步骤 1	**configure terminal**	进入全局配置模式
步骤 2	ipv6 pim **rp-address** *ipv6-address*[*group-access-list*] 示例：	给一个特定的组范围配置一个 PIM RP 的地址

	(config) # **ipv6 pim rp-address 2001:DB8::01:800:200E:8C6C acc-grp-1**	
步骤 3	**exit** 示例： (config-if) # **exit**	离开接口配置模式，并且进入特权 EXEC 模式
步骤 4	**show ipv6 pim interface** [state-on] [**state-off**] [*type-number*] 示例： # **show ipv6 pim interface**	显示配置了 PIM 的那些接口的信息
步骤 5	**show ipv6 pim group-map** [*group-name* | group-address] | [*group-range* | *group-mask*] [**info-source {bsr** | **default** | **embedded-rp** | **static**}] 示例： # **show ipv6 pim group-map**	显示一个 IPv6 组播组映射表
步骤 6	**show ipv6 pim neighbor** [**detail**] [*interface-type interface-number* | **count**] 示例： # **show ipv6 pim neighbor**	显示 Inspur INOS 软件发现的 PIM 邻居
步骤 7	**show ipv6 pim range-list** [**config**] [*rp-address* | *rp-name*] 示例： # **show ipv6 pim range-list**	显示关于 IPv6 组播范围列表的信息
步骤 8	**show ipv6 pim tunnel** [*interface-type interface-number*] 示例： # **debug ipv6 mld explicit**	显示关于一个接口上的 PIM 注册封装和解封装隧道信息
步骤 9	**debug ipv6 pim** [*group-name* | *group-address* | **interface** *interface-type* | **bsr** | **group** | **mvpn** | **neighbor**] 示例： # **debug ipv6 pim**	对 PIM 协议的操作启用调试

步骤 10	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 PIM 可选项

用户可以从特权 EXEC 模式中，执行下列步骤：

具体步骤

	命令或操作	目的
步骤 1	**configure terminal**	进入全局配置模式
步骤 2	**ipv6 pim spt-threshold infinity** [**group-list** *access-list-name*] 示例： `(config) #` **ipv6 pim spt-threshold infinity group-list acc-grp-1**	配置 PIM 叶交换机何时加入指定组 SPT
步骤 3	**ipv6 pim accept-register** {**list** *access-list* | **route-map** *map-name*} 示例： `(config) #` **ipv6 pim accept-register route-map reg-filter**	配置在 RP 上接受或拒绝注册
步骤 4	**interface type** *number* 示例： `Device(config)#` **interface GigabitEthernet 1/0/1**	指定接口类型和编号，并让交换机进入接口配置模式
步骤 5	**ipv6 pim dr-priority** *value* 示例： `(config-if) #` **ipv6 pim dr-priority 3**	在一台 PIM 交换机上配置 DR 优先级
步骤 6	**ipv6 pim hello-interval** *seconds* 示例： `(config-if) #` **ipv6 pim hello-interval 45**	在一个接口上配置 PIM hello 消息的频率
步骤 7	**ipv6 pim join-prune-interval** *seconds* 示例： `(config-if) #` **ipv6 pim join-prune-interval 75**	配置接口的周期性发送加入和修剪通告消息的时间间隔
步骤 8	**exit**	连续两次输入这条命令，离开接口配置模式，并且进入特权 EXEC 模式

	示例： (config-if) # **exit**	
步骤 9	**ipv6 pim join-prune statistic** [*interface-type*] 示例： (config-if) # **show ipv6 pim join-prune statistic**	显示每个接口最新汇聚数据包，平均是由多少加入-修剪数据包汇聚成一个数据包的
步骤 10	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

重置 PIM 流量计数器

如果 PIM 出现了故障，或者为了验证收发 PIM 数据包的预期数量，用户可以清空 PIM 流量计数器。在将流量计数器清空之后，用户可以输入命令 show ipv6 pim traffic 来查看 PIM 是否工作正常，以及 PIM 数据包的收发是否无误。

用户可以从特权 EXEC 模式中，执行下列步骤：

具体步骤

	命令或操作	**目的**
步骤 1	**clear ipv6 pim traffic** 示例： # **clear ipv6 pim traffic**	重置 PIM 流量计数器
步骤 2	**show ipv6 pim traffic** 示例： # **show ipv6 pim traffic**	显示 PIM 流量计数器
步骤 3	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

清空 PIM 拓扑表以重置 MRIB 连接

使用 MRIB 不需要进行任何配置。但在有些情况下，用户可能希望清空 PIM 拓扑表，以便重置 MRIB 连接并验证 MRIB 信息。

用户可以从特权 EXEC 模式中，执行下列步骤：

具体步骤

	命令或操作	**目的**
步骤 1	**clear ipv6 pim topology** [*group-name*	*group-address*] 示例： # **clear ipv6 pim topology FF04::10**	清除 PIM 拓扑表
步骤 2	**show ipv6 mrib client** [**filter**] [**name** {*client-name*	*client-name* **: client-id**}] 示例： # **show ipv6 mrib client**	显示关于一个接口的组播相关信息

步骤 3	**show ipv6 mrib route** {**link-local** | **summary** | [*sourceaddress-or-name* | *] [*groupname-or-address*[*prefix-length*]]] 示例： # **show ipv6 mrib route**	显示 MRIB 路由信息
步骤 4	**show ipv6 pimtopology** [*groupname-or-address* [*sourceaddress-or-name*] | **link-local** | **route-count** [**detail**]] 示例： # **show ipv6 pim topology**	显示某个指定组或所有组的 PIM 拓扑表信息
步骤 5	**debug ipv6 mrib client** 示例： # **debug ipv6 mrib client**	对 MRIB 客户端管理操作启用调试
步骤 6	**debug ipv6 mrib io** 示例： # **debug ipv6 mrib io**	对 MRIB I/O 时间启用调试
步骤 7	**debug ipv6 mrib proxy** 示例： # **debug ipv6 mrib proxy**	对交换机处理器和分布式交换机平台上的线卡之间的 MRIB 代理操作启用调试
步骤 8	**debug ipv6 mrib route** [*group-name* | *group-address*] 示例： # **debug ipv6 mrib route**	显示关于 MRIB 路由条目操作的信息
步骤 9	**debug ipv6 mrib table** 示例： # **debug ipv6 mrib table**	对 MRIB 表管理操作启用调试
步骤 10	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 PIM IPv6 末节路由

PIM 末节路由特性支持在分布层和接入层之间启用组播路由。该特性支持两类 PIM 接口：上行链路 PIM 接口和 PIM 被动接口。配置为 PIM 被动模式的路由接口不会传输和转发 PIM 控

制流量，它只会传输和转发 MLD 流量。

PIM IPv6 末节路由配置指导方针

- 在配置 PIM 末节路由之前，用户必须在末节路由器和中央路由器上都配置 IPv6 组播路由。此外，用户还必须在末节路由器的上行链路接口上配置（稀疏模式的）PIM 模式；
- PIM 末节路由器不会路由在分布层路由器之间传输的流量，但单播（EIGRP）末节路由会执行这项操作，而用户必须配置单播末节路由来帮助 PIM 末节路由器的转发行为。要想进一步了解相关信息，可以参阅“EIGRPv6 末节路由”；
- 二层接入域中只允许部署直连的组播（MLD）接收方和源。在接入域中不支持 PIM 协议；
- 不支持冗余 PIM 末节路由器拓扑。

默认的 IPv6 PIM 路由配置

这张表显示了设备上的默认 IPv6 PIM 路由配置。

表 22：默认的组播路由配置

特性	默认设置
组播路由	在所有接口上启用
PIM 版本	第 2 版
PIM 模式	没有定义任何模式
PIM 末节路由	无配置
PIM RP 地址	无配置
PIM 域边缘	禁用
PIM 组播边界	无
候选 BSR	禁用
候选 RP	禁用
最短路径树的门限值	0kb/s
PIM 路由器查询消息时间间隔	30 秒

启用 IPv6 PIM 末节路由

在开始前

在 IPv6 中，PIM 末节路由默认是禁用的。用户可以从特权 EXEC 模式中，按照下列步骤在一个接口上启用 PIM 末节路由。

总步骤

1. enable

2. configure terminal

3. ipv6 multicast pim-passive-enable

4. interface *interface-id*

5. ipv6 pim

6. ipv6 pim {**bsr**} | {**dr-priority** | *value*} | {**hello-interval** | *seconds*} | {**join-prune-interval** | *seconds*} | {**passive**}

7. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device#` **configure terminal**	
步骤 3	**ipv6multicast pim-passive-enable** 示例： `Device(config-if)#` **ipv6 multicast pim-passive-enable**	在交换机上启用 IPv6 组播 PIM 路由
步骤 4	**interface type** *number* 示例： `Device(config)#` **interface gigabitethernet 9/0/6**	指定要启用 PIM 末节路由的接口，并进入接口配置模式。 指定的接口必须为下列接口之一： • 路由端口：通过接口配置命令 **no switchport** 被配置为了三层端口的物理端口。用户还应该在这个接口上启用 IP PIM 稀疏模式，并让这个接口作为静态连接成员加入一个 MLD 静态组中； • SVI：通过全局配置命令 **interface vlan** *vlan-id* 创建的 VLAN 接口。用户还应该在这个 VLAN 上启用 IP PIM 稀疏模式，让这个接口作为静态连接成员加入一个 MLD 静态组中，并且在 VLAN、MLD 静态组和物理接口上启用 MLD snooping； 这些接口上必须配置 IPv6 地址。
步骤 5	**ipv6 pim** 示例： `Device(config-if)#` **ipv6 pim**	在接口上启用 PIM
步骤 6	**ipv6 pim** {**bsr**} | {**dr-priority** | *value*} | {**hello-interval** | *seconds*} | {**join-prune-interval** | *seconds*} | {**passive**} 示例： `Device(config-if)#` **ipv6 pim** `bsr|dr-priority|hello-interval|join-prune-interval|passive`	在接口上配置各类 PIM 末节特性。 输入 **bsr** 在 PIM 交换机上配置 BSR； 输入 **dr-priority** 在 PIM 交换机上配置 DR 优先级； 输入 **hello-interval** 在一个接口上配置 PIM hello 消息的频率； 输入 **join-prune-interval** 给一个接口配置周期性发送的加入与修改通告消息的时间间隔； 输入 **passive** 将 PIM 配置为被动模式
步骤 7	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式

监控 IPv6 PIM 末节路由

表 23：PIM 末节配置的 show 命令

命令	目的
show ipv6 pim interface `Device#` **show ipv6 pim interface**	显示在各个接口上启用的 PIM 末节
show ipv6 mld groups `Device#` **show ipv6 mld groups**	显示已加入某个组播源组的感兴趣客户端
show ipv6 mroute `Device#` **show ipv6 mroute**	验证从源发送给感兴趣客户端的组播流

配置一个 BSR

下面我们来介绍这一节的配置任务。

配置一个 BSR 并验证 BSR 的信息

用户可以从特权 EXEC 模式中，执行下列步骤：

具体步骤

	命令或操作	目的
步骤 1	**configure terminal**	进入全局配置模式
步骤 2	**ipv6 pim bsr candidate bsr** *ipv6-address*[*hash-mask-length*][**priority***priority-value*] 示例： `(config) #` **ipv6 pim bsr candidate bsr 2001:DB8:3000:3000::42 124 priority 10**	将一台交换机配置为候选 BSR
步骤 3	**interface type** *number* 示例： `Device(config)#` **interface GigabitEthernet 1/0/1**	指定接口类型和编号，并让交换机进入接口配置模式
步骤 4	**ipv6 pim bsr border** 示例： `(config-if) #` **ipv6 pim bsr border**	在指定接口上给任意范围内的全部 BSM 配置一个边界
步骤 5	**exit** 示例： `(config-if) #` **exit**	连续两次输入这条命令，离开接口配置模式，并且进入特权 EXEC 模式
步骤 6	**show ipv6 pim**bsr {**election** | **rp-cache** | **candidate-rp**}	显示与 PIM BSR 协议处理相关的信息

	示例： `(config-if) #` **show ipv6 pim bsr election**	
步骤 7	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

向 BSR 发送 PIM RP 通告

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal**	进入全局配置模式
步骤 2	**ipv6 pim bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] [**priority***priority-value*] [**interval** seconds] 示例： `(config) #` **ipv6 pim bsr candidate rp 2001:DB8:3000:3000::42 priority 0**	向 BSR 发送 PIM RP 通告
步骤 3	**interface type** *number* 示例： `Device(config)#` **interface GigabitEthernet 1/0/1**	指定接口类型和编号，并让交换机进入接口配置模式
步骤 4	**ipv6 pim bsr border** 示例： `(config-if) #` **ipv6 pim bsr border**	在指定接口上给任意范围内的全部 BSM 配置一个边界
步骤 5	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置在作用域内使用的 BSR

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal**	进入全局配置模式
步骤 2	**ipv6 pim bsr candidate bsr** *ipv6-address*[*hash-mask-length*][**priority***priority-value*] 示例： `(config) #` **ipv6 pim bsr candidate bsr 2001:DB8:1:1:4**	将一台交换机配置为候选 BSR
步骤 3	**ipv6 pim bsr candidate rp** *ipv6-address* [**group-list**	对候选 RP 进行配置，让其向 BSR 发送 PIM RP 通告

	access-list-name] [**priority** *priority-value*] [**interval** seconds] 示例： `(config) #` **ipv6 pim bsr candidate rp 2001:DB8:1:1:1 group-list list scope 6**	
步骤 4	**interface type** *number* 示例： `Device(config)#` **interface GigabitEthernet 1/0/1**	指定接口类型和编号，并让交换机进入接口配置模式
步骤 5	**ipv6 multicast boundary scope** *scope-value* 示例： `(config-if) #` **ipv6 multicast boundary scope 6**	在接口上给指定范围配置一个组播边界
步骤 6	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 BSR 交换机以通告范围与 RP 的映射关系（Scope-to-RP Mappings）

用户可以对 IPv6 BSR 交换机进行静态配置，让其直接通告范围与 RP 的映射关系，而不从候选 RP 消息学习这些映射关系。用户有可能希望对 BSR 交换机进行配置，让其通告范围与 RP 的映射关系，这样一来，不支持 BSR 的 RP 也就可以导入到 BSR 中了。启用这个特性也可以让本地候选 BSR 交换机上的已知远端 RP 学习到位于企业 BSR 域之外的 RP。

用户可以从特权 EXEC 模式中，执行下列步骤：

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal**	进入全局配置模式
步骤 2	**ipv6 pim bsr announced rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] 示例： `(config)#` **ipv6 pim bsr announced rp 2001:DB8:3000:3000::42 priority 0**	从 BSR 上将范围与 RP 的映射直接通告给指定的候选 RP
步骤 3	**interface type** *number* 示例： `Device(config)#` **interface GigabitEthernet 1/0/1**	指定接口类型和编号，并让交换机进入接口配置模式

步骤 4	**ipv6 pim bsr border** 示例： `(config-if) # ipv6 pim bsr border`	在指定接口上给任意范围内的全部 BSM 配置一个边界
步骤 5	**exit** 示例： `(config-if) # exit`	连续两次输入这条命令，离开接口配置模式，并且进入特权 EXEC 模式
步骤 6	**show ipv6 pim**bsr {**election** | **rp-cache** | **candidate-rp**} 示例： `(config-if) # show ipv6 pim bsr election`	显示与 PIM BSR 协议处理相关的信息
步骤 7	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 SSM 映射

在用户启用了 SSM 映射特性时，基于 DNS 的 SSM 映射也会自动启用，这表示交换机会向 DNS 服务器查询组播 MLDv1 报告的源。

用户也可以使用基于 DNS 的映射或者静态 SSM 映射，具体做法取决于交换机上的配置。如果用户选择使用静态 SSM 映射，那也可以配置多个静态 SSM 映射。如果用户配置了多个静态 SSM 映射，那么设备就会使用所有匹配的访问列表的源地址。

注释： 要使用基于 DNS 的 SSM 映射，那么交换机至少需要能够找到一台配置正确的 DNS 服务器，而这台 DNS 服务器有可能与交换机是直连的。

用户可以从特权 EXEC 模式中，执行下列步骤：

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal**	进入全局配置模式
步骤 2	**ipv6 mld ssm-map enable** 示例： `(config) # ipv6 mld ssm-map enable`	在配置的 SSM 范围内对组启用 SSM 映射特性
步骤 3	**no ipv6 mld ssm-map query dns** 示例： `(config) # no ipv6 mld ssm-map query dns`	禁用基于 DNS 的 SSM 映射
步骤 4	**ipv6 mld ssm-map static**	配置静态 SSM 映射

	access-list source-address 示例： `(config-if) #` **ipv6 mld ssm-map static SSM_MAP_ACL_2 2001:DB8:1::1**	
步骤 5	**exit** 示例： `(config-if) #` **exit**	离开全局配置模式，让交换机返回特权 EXEC 模式
步骤 6	**show ipv6 mld ssm-map** [*source-address*] 示例： `(config-if) #` **show ipv6 mld ssm-map**	显示 SSM 映射信息
步骤 7	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置静态组播路由（Mroute）

IPv6 中的静态组播路由（mroute）可以作为 IPv6 静态路由的一种扩展形式。用户可以在交换机上配置一条只能用来转发单播流量的静态路由，也可以配置一条只用于组播 RPF 选择的静态组播路由，或者配置一条既可以转发单播流量也可以执行组播 RPF 选择的路由。
用户可以从特权 EXEC 模式中，执行下列步骤：

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal**	进入全局配置模式
步骤 2	**ipv6 route** {*ipv6-prefix / prefix-length ipv6-address* | *interface-type interface-number ipv6-address*]} [*administrative-distance*][*administrative-multicast-distance*| *unicast* | *multicast*] [**tag** *tag*] 示例： `(config) #` **ipv6 mld ssm-map enable**	建立静态 IPv6 路由。示例中显示的是一条既可以用来转发单播路由，也可以用于组播 RPF 选择的静态路由
步骤 3	**exit** 示例：	离开全局配置模式，让交换机返回特权 EXEC 模式

	(config-if) # **exit**	
步骤 4	**show ipv6 mroute** [link-local | [*group-name* | *group-address* [*source-address* | *source-name*]] **[summary**] [**count**] 示例： # **show ipv6 mroute ff07::1**	显示 IPv6 组播路由表中的内容
步骤 5	**show ipv6 mroute** [**link-local** | *group-name* | *group-address*] **active** [*kbps*] 示例： (config-if) # **show ipv6 mroute active**	显示交换机上的活动组播流量
步骤 6	**show ipv6 rpf** [*ipv6-prefix*] 示例： (config-if) # **show ipv6 rpf 2001::1:1:2**	对一个给定的单播主机地址和前缀校验 RPF 信息
步骤 7	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

在 IPv6 组播中使用 MFIB

在启用 IPv6 组播路由时，组播转发也会自动启用。

查看 MFIB 在 IPv6 组播中的操作

用户可以从特权 EXEC 模式中，执行下列步骤：

具体步骤

	命令或操作	目的
步骤 1	**show ipv6 mfib** [**linkscope** | **verbose** | *group-address-name* | *ipv6-prefix / prefix-length* | *source-address-name* | **count** | **interface** | **status** | **summary**] 示例： # **show ipv6 mfib**	显示 IPv6 MFIB 中的转发条目和接口
步骤 2	**show ipv6 mfib** [all | linkscope | group-name | group-address [source-name | source-address]] **count**	显示 IPv6 组播路由表中的信息

	示例： # **show ipv6 mfib ff07::1**	
步骤 3	**show ipv6 mfib interface** 示例： # **show ipv6 mfib interface**	显示启用了 IPv6 组播的接口及这些接口的转发状态信息
步骤 4	**show ipv6 mfib status** 示例： # **show ipv6 mfib status**	显示总的 MFIB 配置及操作状态
步骤 5	**show ipv6 mfib summary** 示例： # **show ipv6 mfib summary**	显示关于 IPv6 MFIB 条目和接口数量的汇总信息
步骤 6	**debug ipv6 mfib** [*group-name* | *group-address*] [**adjacency** | **db** | **fs** | **init** | **interface** | 示例： # **debug ipv6 mfib FF04::10 pak**	对 IPv6 MFIB 启用调试

重置 MFIB 流量计数器

用户可以从特权 EXEC 模式中，执行下列步骤：

具体步骤

	命令或操作	**目的**
步骤 1	**clear ipv6 mfib counters** [*group-name* | **group-address** [*source-address* | *source-name*]] 示例： # **clear ipv6 mfib counters FF04::10**	重置所有活动 MFIB 流量计数器

配置 IPv6 ACL

IPv6 ACL 的前提条件

用户可以通过创建 IPv6 访问控制列表（ACL）并且将它们应用到接口的方式来过滤 IPv6（IP 第 6 版）流量，这种方法与创建和应用 IPv4（IP 第 4 版）命名 ACL 的方式相当类似。如果交换机运行的是 IP base 特性集，那么用户也可以创建和应用入站路由器 ACL 来过滤三层管理流量。

IPv6 ACL 的限制条件

在 IPv4 环境中，用户可以配置标准和扩展的编号 IP ACL、命名的 IP ACL 和 MAC ACL。而 IPv6 只支持命名的 ACL。

本设备支持大多数 Inspur INOS 支持的 IPv6 ACL，但下列 ACL 是例外：

- 本设备不支持匹配下列关键字：**flowlabel**、**routing header** 和 **undetermined-transport**；
- 本设备不支持自反 ACL（也就是不支持 **reflect** 这个关键字）；
- 本设备不支持对 IPv6 数据帧应用基于 MAC 的 ACL；
- 在配置 ACL 时，设备对于用户在 ACL 中输入的关键字没有限制，无论配置该命令的平台是否支持这些关键字。在将 ACL 应用到一个需要硬件转发的接口（物理端口或 SVI）时，设备会判断接口是否支持这个 ACL。如果不支持，那么设备会拒绝在接口上应用这个 ACL；
- 如果用户将 ACL 应用到了一个接口，接下来用户又准备向这个 ACL 中添加一条包含设备不支持的关键字的 ACE（访问控制条目），那么设备不会允许用户将这条 ACE 添加到这个接口上应用的 ACL。

关于 IPv6 ACL 的信息

访问控制列表是一系列用来限制访问某个接口的规则。用户需要在设备上配置 ACL，并且将 ACL 应用到管理接口、AP-管理员接口、任何动态接口、或者应用到控制器中央处理（CPU）以控制所有去往 CPU 的流量。

用户可以创建一个预认证 ACL 来执行 web 认证。这类 ACL 的作用是在认证完成之前，就放行某些类型的流量。

IPv6 ACL 支持的选项与 IPv4 ACL 相同，其中包括源、目的、源端口和目的端口。

注释： 用户可以在网络中通过阻塞 IPv6 流量方式来单独启用 IPv4 流量。也就是说，用户可以配置一个过滤所有 IPv6 流量的 IPv6 ACL，并且将它引用到某个 WLAN 或者所有 WAN 上。

理解 IPv6 ACL

交换机支持两种类型的 IPv6 ACL：

- 支持在三层接口上针对入站流量和出站流量配置 IPv6 路由器 ACL，这里所说的三层接口既可以是路由端口，也可以是交换机虚拟接口（SVI）或者三层 EtherChannel。IPv6 路由器 ACL 只能应用在那些路由 IPv6 数据包的接口上；
- 支持在二层接口上针对入站流量配置 IPv6 端口 ACL。IPv6 端口 ACL 会作用于所有进入这个接口的 IPv6 数据包；

运行 IP base 特性集的交换机只支持入站方向的路由器 IPv6 ACL。它不支持端口 ACL 或出站 IPv6 路由器 ACL。

注释： 如果用户配置了设备不支持的 IPv6 ACL，那么设备就会弹出错误消息，用户所作的配置也不会生效。

交换机不支持针对 IPv6 流量使用 VLAN ACL（即 VLAN map）。

用户可以同时在一个接口上应用 IPv4 ACL 和 IPv6 ACL。IPv6 端口 ACL 的优先级高于路由器 ACL，这一点 IPv4 ACL 和 IPv6 ACL 是一样的。

- 当一个 SVI 上同时应用了入站方向的路由器 ACL 和入站方向的端口 ACL 时，如果应用了 ACL 的那些端口接收到数据包，设备就会使用端口 ACL 来过滤数据包。如果其它一些端口接收到的被路由 IP 数据包，设备则会使用路由器 ACL 进行过滤。其余数据包则不会用 ACL 进行过滤；
- 当一个 SVI 上同时应用了出站方向的路由器 ACL 和入站方向的端口 ACL 时，如果应用了 ACL 的那些端口接收到数据包，设备就会使用端口 ACL 来过滤数据包。而出站的被路由 IPv6 数据包则会通过路由器 ACL 进行过滤。其余数据包则不会用 ACL 进行过滤。

注释： 只要一个接口上应用了端口 ACL（无论 IPv4 ACL、IPv6 ACL 还是 MAC ACL），那么设备就会用这个端口 ACL 来过滤数据包，而该端口所在的 SVI 上所应用的任何路由器 ACL 设备此时都会被设备忽略。

ACL 的类型

每用户（Per User）IPv6 ACL

对于每用户 ACL 来说，全部访问控制条目（ACE）都要以文本的形式配置在 ACS 上；

ACE 不是配置在控制器上的。ACE 会通过 ACCESS-Accept 这个属性发送给设备，设备会直接将它应用于客户端。设备不支持在出站方向部署每用户 ACL。

过滤器 ID IPv6 ACL

对于过滤器 ID ACL，全部 ACE 和 acl name(filter-id)都要配置在设备上，只有 filter-id 要配置在 ACS 上。ACS 会将 filter-id 放在 ACCESS-Accept 属性中发送给设备，而设备会使用 filter-id 来查找 ACE，然后将 ACE 应用于客户端。当客户端在二层漫游到另一台外来的设备上时，只有 filter-id 会通过 Handoff 消息发送给那台设备。设备不支持在出站方向针对不同用户配置 ACL 来执行过滤。用户还需要提前在那台外来设备上配置 filter-id 和 ACE。

可下载的 IPv6 ACL

对于可下载 ACL（dACL）来说，全部 ACE 和 dacl-名称都只能配置在 ACS 上。

注释： 控制器上并不能配置任何 ACL。

ACS 会将 dacl-名称通过 ACCESS-Accept 属性发送给设备，而设备则会提取出 dacl 名称，再把 dACL 名称通过 access-request 属性发回给 ACS，来获取 ACE。

ACS 会使用 access-accept 属性来响应设备请求的 ACE。而外来设备会通过 dacl 名称来联系 ACS 服务器获取 ACE。

IPv6 ACL 与交换机堆栈

堆栈主设备可以在硬件中支持 IPv6 ACL，也可以将 IPv6 ACL 分发给堆栈的成员设备。

注释： 要想在交换机堆栈中使用所有 IPv6 功能，那么堆栈的所有成员设备都必须运行 IP services 特性集。

如果一台新的交换机成为了主设备，它会将 ACL 的配置分发给所有堆栈成员设备。成员交换机会使用主设备分发的配置进行同步，并且将那些已经不再使用的条目清除掉。
在用户对 ACL 进行修改时、应用到接口上或者从接口上删除时，住设备都会将配置变更分发给堆栈中的所有成员设备。

配置 IPv6 ACL

要过滤 IPv6 流量，需要执行下面的步骤：

在开始前

在配置 IPv6 ACL 之前，用户必须从 IPv4 SDM 模版和 IPv6 SDM 模版中选择其一。

总步骤

1 创建一个 IPv6 ACL，并且进入 IPv6 访问列表配置模式；
2 配置 IPv6 ACL 来过滤（阻塞）或放行（允许）流量；
3 将 IPv6 ACL 应用到需要过滤流量的接口上；
4 将 IPv6 ACL 应用到一个接口上。对于路由器 ACL，用户还必须在应用 ACL 的接口上配置 IPv6 地址。

具体步骤

	命令与操作	**目的**
步骤 1	创建一个 IPv6 ACL，并且进入 IPv6 访问列表配置模式	
步骤 2	配置 IPv6 ACL 来过滤（阻塞）或放行（允许）流量	
步骤 3	将 IPv6 ACL 应用到需要过滤流量的接口上	
步骤 4	将 IPv6 ACL 应用到一个接口上。对于路由器 ACL，用户还必须在应用 ACL 的接口上配置 IPv6 地址	

默认的 IPv6 ACL 配置

默认状态设备上没有配置和应用 IPv6 ACL。

与其他特性与交换机的互动

- 如果用户配置了一条 IPv6 路由器 ACL 来拒绝数据包，那么路由器就不会路由这个数据包。这个数据包的副本会被发送到互联网控制消息协议（ICMP）队列中，给该数据帧生成一条 ICMP 不可达消息；
- 如果由于设备上配置了端口 ACL 去丢弃一个桥接数据帧，那么设备就不会去桥接这个数据帧；
- 用户可以同时在交换机或者交换机集群上配置 IPv4 和 IPv6 ACL，然后将 IPv4 和 IPv6 ACL 同时应用在一个接口上。每个 ACL 的命名必须是唯一的；如果用户要给一个 ACL 配置一个已经配置过的名称，那么系统就会显示错误消息。
- 用户要使用不同的命令来创建 IPv4 和 IPv6 ACL，并将它们关联到同一个二层或三层接口上。如果在关联 ACL 时用户输入的命令不正确（比如在将 IPv6 ACL 关联到接口上时输入了 IPv4 的命令），那么用户就会看到一条错误消息；
- 用户不能使用 MAC ACL 来过滤 IPv6 数据帧。MAC ACL 只能过滤非 IP 数据帧；

- 如果硬件的内存已满，那么对于用户继续配置的 ACL，相关的丢包操作会交由 CPU 来执行，ACL 也会应用到软件中。如果硬件已经满，console 就会显示一条消息，显示 ACL 已经被卸载，这个接口会开始丢弃数据包。

注释： 接口只会丢弃那些无法添加的 ACL（IPv4、IPv6 或 MAC）所对应类型的数据包。

如何配置 IPv6 ACL

创建 IPv6 ACL

用户可以从特权 EXEC 模式中，通过下列步骤来创建 IPv6 ACL：

总步骤

1. configure terminal

2. ipv6 access-list *acl_name*

3. {deny|permit} protocol

4. {deny|permit} tcp

5. {deny|permit} udp

6. {deny|permit} icmp

7. end

8. show ipv6 access-list

9. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 2	**ipv6 access-list** *acl_name* 示例： `ipv6 access-list access-list-name`	用名称来定义 IPv6 访问列表，并进入 IPv6 访问列表配置模式
步骤 3	**{deny|permit} protocol** 示例： `{deny | permit} protocol {source-ipv6-prefix/prefix-length | any | host source-ipv6-address} [operator [port number]]{destination-ipv6-prefix/prefix-length | any |host destination-ipv6-address} [operator`	输入 deny 或 permit 来设置是阻塞还是拒绝那些满足条件的数据包。这些条件包括： • 在 protocol 部分，可以输入网络协议的名称或协议号：ahp、esp、icmp、ipv6、pcp、stcp、tcp 或 udp、或者在 0 到 255 之内可以代表一个 IPv6 协议号的整数； • source-ipv6-prefix/prefix-length 或 destination-ipv6-prefix/prefix-length 是要设置匹配条件的那个源或目的 IPv6 网络，这个参数要用冒号分隔的 16 位值（即 RFC 2373 中定义的格式）来表示； • any 就是 IPv6 前缀::/0 的简称；

	[port-number]][dscp value] [fragments][log] [log-input] [routing][sequence value] [time-range name]	• 在 host source-ipv6-address 或者 host destination-ipv6-prefix 部分，要输入要设置匹配条件的那个源或目的 IPv6 主机地址，这个参数要用冒号分隔的 16 位值（即 RFC 2373 中定义的格式）来表示； • （可选）在 operator 部分，设置比较指定协议的源或目的端口时使用的操作符。操作符包括 lt（小于，less than）、gt（大于，greater than）、eq（等于，equal）、neq（不等于，not equal）和范围 如果操作符跟在参数 source-ipv6-prefix/prefix-length 后面，那么它匹配的就是源端口。如果操作符跟在参数 destination-ipv6-prefix/prefix-length 后，那匹配的就是目的端口。 • （可选）port-number 是一个从 0 到 65535 之间的十进制数，或者一个 TCP 或 UDP 端口的名称。用户只能在过滤 TCP 流量时使用 TCP 端口名，在过滤 UDP 流量时使用 UDP 端口名； • （可选）输入 dscp value 用差分服务代码点值匹配每个 IPv6 数据包头部中流量类型（Traffic Class）字段中的流量类型值。取值范围是从 0 到 63； • （可选）输入 fragments 来校验非初始数据帧。只有协议为 ipv6 时，才可以看到 fragments 这个关键字； • （可选）输入 log 让系统向 console 发送关于匹配条目的数据包的日志消息。输入 log-input 在日志条目中包含入站接口。只有路由器 ACL 支持日志记录功能； • （可选）输入 routing 指定要被路由的 IPv6 数据包； • （可选）输入 sequence value 指定访问列表语句的序号；取值范围是 1 到 4294967295； • （可选）输入 time-range name 指定这条 deny 或 permit 语句应用的时间。
步骤 4	**{deny|permit} tcp** 示例： {deny | permit} tcp {source-ipv6-prefix/prefix-length | any | hostsource-ipv6-address} [operator [port-number]]{destination-	（可选）定义 TCP 访问列表和访问条件。 对于传输控制协议（Transmission Control Protocol）应输入 tcp。这些参数与步骤 3 中介绍的参数相同，此外还多了下列可选参数： • ack：Acknowledgement 位置位； • established：指已建立的连接。如果 TCP 数据报中已经将 ACK 或 RST 位置位，就会出现匹配；

	ipv6-prefix/prefix-length | any |hostdestination-ipv6-address} [operator [port-number]][ack] [dscp value][established] [fin] [log][log-input] [neq {port |protocol}] [psh] [range{port | protocol}] [rst][routing] [sequence value] [syn] [time-range name][urg]	• fin：Finished 位置位，这表示发送方不会再发送更多数据； • neq {port |protocol}：只匹配那些不是指定端口号的流量； • psh：Push 功能位置位； • range {port | protocol}：只匹配端口号范围内的数据包； • rst：Reset 位置位； • syn：Synchronize 位置位； • urg：Urgent 指针位置位。
步骤 5	**{deny|permit} udp** 示例： {deny | permit} udp {source-ipv6-prefix/prefix-length | any | hostsource-ipv6-address} [operator [port-number]]{destination-ipv6-prefix/prefix-length | any | hostdestination-ipv6-address} [operator [port-number]][dscp value] [log][log-input] [neq {port |protocol}] [range {port |protocol}] [routing][sequence value][time-range name]	（可选）定义 UDP 访问列表和访问条件。 对于用户数据报协议（User Datagram Protocol）应输入 udp。这些参数与 TCP 中介绍的参数相同，但 operator [port-number]中设置的端口号或名称必须为 UDP 端口号或名称。此外，established 这项参数也不适用于 UDP。
步骤 6	**{deny|permit} icmp** 示例： {deny | permit} icmp {source-ipv6-prefix/prefix-length | any | hostsource-ipv6-address} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any | hostdestination-ipv6-address} [operator [port-number]][icmp-type [icmp-code] |icmp-message] [dscpvalue] [log] [log-	（可选）定义 ICMP 访问列表和访问条件。 对于互联网控制消息协议（Internet Control Message Protocol）应输入 icmp。这些参数与大部分前面步骤中的参数相同，但 ICMP 有一些特殊的消息类型和代码参数，这些可选关键字的表意如下： • icmp-type：输入要过滤的 ICMP 消息类型，这是一个取值范围在 0 到 255 之间的整数； • icmp-code：输入要过滤的 ICMP 消息所对应的 ICMP 消息代码类型，这是一个取值范围在 0 到 255 之间的整数； • icmp-message：输入要过滤的 ICMP 消息所对应的 ICMP 消息类型和代码名称。用户如果希望查看 ICMP 消息类型名称和代码名称的列表，可以输入?来查看这个版本系统提供的提示信息

	input] [routing] [sequence value][time-range name]	
步骤 7	**end** 示例： Device(config-if)# **end**	返回特权 EXEC 模式。此外，用户也可以按下 **Ctrl-Z** 返回全局配置模式
步骤 8	**show ipv6 access-list** 示例： show ipv6 access-list	查看访问列表的配置
步骤 9	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

将 IPv6 ACL 应用到一个接口

在这一节中，我们会介绍如何将 IPv6 ACL 应用到一个网络接口上。用户可以将一个 IPv6 ACL 应用到二层或三层接口的出站或入站方向。用户也可以将 IPv6 ACL 应用于三层接口的入站管理流量。

用户可以从特权 EXEC 模式中，通过下列步骤将访问控制列表应用到一个接口：

总步骤

1. configure terminal

2. interface interface_id

3. no switchport

4. ipv6 address *ipv6_address*

5. ipv6 traffic-filter *acl_name*

6. end

7. show running-config interface tenGigabitEthernet 1/0/3

8. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** 示例： Device# **configure terminal**	进入全局配置模式
步骤 2	**interface** interface_id 示例： Device# interface interface-id	指定要应用访问列表的二层接口（如 ACL 为端口 ACL）或三层交换机虚拟接口（如 ACL 为路由器 ACL），并进入接口的配置模式
步骤 3	**no switchport** 示例： Device# no switchport	将接口由（默认的）二层模式修改为三层模式（仅于要应用路由器 ACL 时）

步骤 4	**ipv6 address** *ipv6_address* 示例： `Device# ipv6 address ipv6-address`	在三层接口上配置一个 IPv6 地址（适用于路由器 ACL）。 **注释：**二层接口上不需要配置这条命令，如果用户已经在这个接口上配置了一个 IPv6 地址，也不需要配置这条命令
步骤 5	**ipv6 traffic-filter** *acl_name* 示例： `Device# ipv6 traffic-filter access-list-name {in | out}`	将访问列表应用于接口的入站或出站流量
步骤 6	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式。此外，用户也可以按下 **Ctrl-Z** 返回全局配置模式

步骤 7	**show running-config interface tenGigabitEthernet 1/0/3**
	
	示例：
	`Device# show running-config interface tenGigabitEthernet 1/0/3`
	`.....`
	`.....`
	`Building configuration .....`
	`.....`
	`Current configuration : 98 bytes`
	`!`
	`interface TenGigabitEthernet1/0/3`
	`switchport mode trunk`
	`ipv6 traffic-filter MyFilter out`
	`end`
		查看配置命令的汇总信息

步骤 8	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

查看 IPv6 ACL

显示 IPv6 ACL

用户可以通过一条或几条特权 EXEC 命令来查看设备上配置的所有访问列表、IPv6 访问列表，

或者某条指定的访问列表。

具体步骤

	命令或操作	目的
步骤 1	**show access-list** 示例： `Device# show access-lists`	显示设备上配置的所有访问列表
步骤 2	**show ipv6 access-list** *acl_name* 示例： `Device# show ipv6 access-list` *[access-list-name]*	显示所有 IPv6 访问列表或指定名称的访问列表

IPv6 ACL 的配置示例

示例：创建 IPv6 ACL

在这个示例中，用户配置了一个名为 CISCO 的 IPv6 访问列表。列表中的第 1 条 deny 条目会拒绝所有目的 TCP 端口号大于 5000 的数据包。第 2 条 deny 条目会拒绝源 UDP 端口号小于 5000 的数据包。此外，第 2 条 deny 语句也会将所有匹配的情形通过日志发送到 console 接口。列表中的第 1 条 permit 条目会放行所有 ICMP 数据包。列表中的第 2 条 permit 会放行所有其他的流量。第 2 条 permit 语句的存在十分必要，因为在每个 IPv6 访问列表的最后都有一个隐式的全部拒绝条目。

注释： 只有三层接口支持日志记录功能。

```
Device(config)# ipv6 access-list CISCO
Device(config-ipv6-acl)# deny tcp any any gt 5000
Device (config-ipv6-acl)# deny ::/0 lt 5000 ::/0 log
Device(config-ipv6-acl)# permit icmp any any
Device(config-ipv6-acl)# permit any any
```

示例：应用 IPv6 ACL

这个示例显示了如何将名为 Inspur 的访问列表应用到一个三层接口的出站方向上：

```
Device(config)# interface TenGigabitEthernet 1/0/3
Device(config-if)# no switchport
Device(config-if)# ipv6 address 2001::/64 eui-64
Device(config-if)# ipv6 traffic-filter CISCO out
```

示例：查看 IPv6 ACL

这个示例显示了特权 EXEC 命令 **show access-lists** 的输出信息。输出信息中会显示出所有配

置在交换机或交换机堆栈上的访问列表。

```
Device #show access-lists
Extended IP access list hello
10 permit ip any any
IPv6 access list ipv6
permit ipv6 any any sequence 10
```

这个示例显示了特权 EXEC 命令 show ipv6 access-list 的的输出信息。输出信息中只会显示交换机或交换机堆栈上配置的 IPv6 访问列表。

```
Device# show ipv6 access-list
IPv6 access list inbound
permit tcp any any eq bgp (8 matches) sequence 10
permit tcp any any eq telnet (15 matches) sequence 20
permit udp any any sequence 30
IPv6 access list outbound
deny udp any any sequence 10
deny tcp any any eq telnet sequence 20
```

（这里有 3 页空白）

示例：配置 IPv6 邻居绑定

总步骤

1. ipv6 neighbor binding [vlan]19 2001:db8::25:4 **interface tenGigabitEthernet** 1/0/3 aaa.bbb.ccc

具体步骤

	命令或操作	目的
步骤 1	**ipv6 neighbor binding** [vlan]19 2001:db8::25:4 **interface tenGigabitEthernet** 1/0/3 aaa.bbb.ccc 示例： `Device (config)# ipv6 neighbor binding vlan 19 2001:db8::25:4 interface tenGigabitEthernet 1/0/3 aaa.bbb.ccc`	通过设置，确保邻居 2001:db8::25: 4 只有在 VLAN 19 中通过接口 te1/0/3 传输，且源 MAC 地址为 aaa.bbb.ccc，才是有效的

其他参考资料

相关文档

相关主题	文档名
IPv6 命令参考	《IPv6 命令参考手册（Inspur 6650 交换机）》

ACK 配置	《安全配置指南（Inspur 6650 交换机）》

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

关于 IPv6 ACL 的特性信息

特性	版本	修改
IPv6 ACL 功能	Inspur INOS 11.3.1	引入该特性

第 5 部分 IP

配置 HSRP

配置 HSRP

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于配置 HSRP 的信息

HSRP 概述
HSRP 是一种 Inspur 标准的网络高可用性技术，它可以在一个配置了默认网关 IP 地址的 IEEE 802 局域网中，给 IP 主机提供第一跳冗余。HSRP 可以不依靠某一台路由器的可用性来路由 IP 流量。它可以让一系列路由器接口协作，在局域网中的主机看来，这些接口就像是一台虚拟路由器或默认网关一样。当用户在网络或网段中配置 HSRP 时，HSRP 会提供一个虚拟的

MAC（媒体访问控制）地址和一个虚拟 IP 地址，这些地址会在用户配置的一组路由器之间共享。HSRP 可以让两台或多台配置了 HSRP 的路由器使用这个虚拟路由器的 MAC 地址和 IP 网络地址。这台虚拟路由器并不真实存在，它代表的是一组相互提供备份的路由器。其中一台路由器会被选举为主用路由器，另一台路由器则充当备用路由器，当主用路由器出现故障时，备用路由器就会控制这个组 MAC 地址和 IP 地址。

注释： HSRP 组中的路由器可以是任何支持 HSRP 协议的接口，既可以是的路由端口，也可以是交换机虚拟接口（SVI）。

HSRP 可以为网络中的主机提供 IP 流量的冗余，因此增强了网络的可用性。在一组路由器接口中，主用路由器就是路由数据包的那台路由器，而备用路由器则是在主用路由器出现故障，或者满足某些条件时，接替主路由器的设备。

对于那些不支持路由器发现协议，因此无法在正常使用的路由器重启或掉电时转而使用其他路由器的这一类主机来说，HSRP 是十分重要的。当用户在一个网段中配置了 HSRP 时，HSRP 就会提供一个虚拟的 MAC 地址和一个虚拟 IP 地址，这些地址会在运行 HSRP 的路由器接口之间共享。协议选择为主用路由器的路由器接口会负责接收和转发去往组 MAC 地址的数据包。在有 N 台路由器运行 HSRP 时，协议就会分配 N+1 个 IP 地址和 MAC 地址。

HSRP 会检测指定主用路由器的故障，如果发生故障，那么协议选择的备用路由器就会接管热备份组的 MAC 地址和 IP 地址。此时，协议也会选择出新的备用路由器。运行 HSRP 的设备会收发基于 UDP 的组播 hello 数据包，以检测路由器是否发生了故障，并且由此指定哪台路由器充当主用路由器和备用路由器。当用户在一个接口上配置 HSRP 时，这个接口就会自动启用 ICMP（互联网控制消息协议）重定向消息。

用户可以给工作在三层的交换机和交换机堆栈之间配置多个热备份组，这样可以充份利用冗余的路由器。此时，用户需要给接口上配置的每个热备份组各自指定一个组编号。例如，用户可能会将交换机 1 上的一个接口配置为主用路由器，将交换机 2 上的一个接口配置为备用路由器，同时又将交换机 2 的一个接口配置为主用路由器，并将交换机 1 的一个接口配置为备用路由器。

下图显所示为一个配置了 HSRP 协议的网段。每台路由器上都配置了虚拟路由器的 MAC 地址和 IP 网络地址。网络上的主机此后不要再用路由器 A 的 IP 地址作为默认网关了，它们应该要将虚拟路由器的 IP 地址配置为默认网关。当主机 C 向主机 B 发送数据包时，它会把这些数据包发送给虚拟路由器的 MAC 地址。如果因某种原因，路由器 A 无法传输数据包了，那么路由器 B 就会开始响应虚拟 IP 地址和虚拟 MAC 地址，并接替主用路由器的职责成为新的主用路由器。主机 C 会继续使用虚拟路由器的 IP 地址来封装发送给主机 B 的数据包，而路由器 B 会接收到这个数据包并且将其转发给主机 B。当路由器 A 恢复工作时，HSRP 会让路由器 B 继续为主机 C 所在网络的用户提供所需的转发服务，使它们可以与主机 B 所在的网络通信，并让其继续在主机 A 的网段和主机 B 的网络之间执行正常的数据包处理工作。

图 15：典型的 HSRP 配置

Host B	主机 B
Active router	主用路由器
Virtual router	虚拟路由器
Standby router	备用路由器
Router A	路由器 A
Router B	路由器 B
Host C	主机 C
Host A	主机 A

用户可以给工作在三层的交换机和交换机堆栈之间配置多个热备份组，这样可以充份利用冗

余的路由器。此时，用户需要给接口上配置的每个热备份组各自指定一个组编号。例如，用户可能会将交换机 1 上的一个接口配置为主用路由器，将交换机 2 上的一个接口配置为备用路由器，同时又将交换机 2 的一个接口配置为主用路由器，并将交换机 1 的一个接口配置为备用路由器。

HSRP 的版本

Inspur INOS XE 3.3SE 及后续版本都支持下列热备份路由器协议（HSRP）的版本：

交换机支持下列版本的 HSRP：

- HSRPv1：即 HSRP 第 1 版，这是 HSRP 的默认版本。其特性包括：
  - HSRP 组数量的范围是 0 到 255 个；
  - HSRPv1 会使用组播地址 224.0.0.2 来发送 hello 数据包，这些数据包可能会与 Inspur 组管理协议（CGMP）的离开进程相冲突。用户不能同时启用 HSRPv1 和 CGMP；这两种协议是互斥的；
- HSRPv2——HSRP 第 2 版拥有下列特性：
  - HSRPv2 会使用组播地址 224.0.0.102 来发送 hello 数据包。HSRPv2 和 CGMP 离开消息不再是互斥的了，因此这两种协议可以同时启用；
  - HSRPv2 的数据包格式与 HSRPv1 的数据包格式不同。

运行 HSRPv1 的交换机无法识别发送 hello 数据包的物理路由器，因为路由器的源 MAC 地址是虚拟 MAC 地址。

HSRPv2 的数据包格式与 HSRPv1 的数据包格式不同。HSRPv2 数据包使用了 TLV（类型-长度-值）格式，还有一个 6 字节的标识符字段用来填充发送数据包的物理路由器 MAC 地址。

如果运行 HSRPv1 协议的接口接收到了一个 HSRPv2 数据包，那么接口就会忽略数据包的类型字段。

多 HSRP

交换机支持多 HSRP（MHSRP）协议，这是 HSRP 的一种扩展协议，它可以在两个或多个 HSRP 组之间实现负载分担。用户可以配置 MHSRP 实现负载分担，从源网络到服务器网络之间使用两台或多台备用组（和路径）。

在下图中，用户通过配置让一半客户端选择路由器 A，另一半客户端选择路由器 B。路由器 A 和路由器 B 共同组成了两个 HSRP 组。在组 1 中，路由器 A 是默认的主用路由器，因为用户给它分配了最高优先级，而路由器 B 则是备用路由器；在组 2 中，路由器 B 是默认的主用路由器，因为用户给它分配了最高优先级，而路由器 A 则是备用路由器。在正常工作状态下，两台路由器会分担 IP 流量负载。当其中一台路由器变得不可用时，另一台路由器就会成为主用路由器，并承担那台不可用路由器的数据包转发工作。

注释： 对于 MHSRP，用户需要输入 HSRP 接口命令 **standby preempt**。这样，当一台路由器出现故障而又恢复之后，这台恢复的路由器才会抢占回主用身份，并分担负载。

图 16：MHSRP 负载分担

Active router for group 1 Standby router for group 2	在组 1 中充当主用路由器 在组 2 中充当备用路由器
Active router for group 2 Standby router for group 1	在组 2 中充当主用路由器 在组 1 中充当备用路由器
Client 1	客户端 1
Client 2	客户端 2
Client 3	客户端 3
Client 4	客户端 4

SSO HSRP

当一台带有冗余路由处理器（RP）的设备上配置了状态化故障切换（SSO）冗余模式，SSO HSRP 就会改变 HSRP 的操作方式。在这种模式下，RP 是主用设备，另一台 RP 是备用设备，当主用 RP 出现故障时，SSO 就会让备用 RP 接管主用 RP 的身份。

通过这种功能，HSRP SSO 信息会与备用 RP 同步，这可以让故障切换期间，那些使用 HSRP 虚拟 IP 地址发送的流量能够连续不断地进行发送，而不会因为路径出现了变化而导致丢包。此外，如果主用 HSRP 设备上的 RP 出现了故障，那么备用 HSRP 设备就会接管主用 HSRP 设备的身份。

当冗余操作模式被设置为 SSO 时，这种特性在默认状态下是启用的。

HSRP 与交换机堆栈

堆栈主设备会生成 HSRP hello 消息。如果主用 HSRP 堆栈的主设备发生了故障，那么 HSRP 主用状态可能会出现变化。这是因为在新的堆栈主用设备选举出来并启用之前，没有设备会生成 HSRP hello 消息，而在堆栈主设备诶故障时，备用路由器可能就会成为主用设备。

配置 IPv6 HSRP

运行 IP Services 和 IP Base 特性集的交换机支持为 IPv6 流量运行热备份路由器协议（HSRP）。HSRP 可以让路由 IPv6 流量不依赖于某一台路由器是否可用，因此可以实现路由冗余。IPv6 主机会通过 IPv6 邻居发现路由器通告消息了解到路由器是否可用。主机会周期性地通过组播或者进行发送，或者在收到请求时发送。

HSRP IPv6 组有一个虚拟 MAC 地址（这个虚拟 MAC 地址取自于 HSRP 组号）和一个虚拟 IPv6 链路本地地址，后者默认取自于从 HSRP 虚拟 MAC 地址。

当 HSRP 组工作正常时，周期性消息会发送到 HSRP 虚拟 IPv6 链路本地地址。当组不再处于 active 状态时，设备在发送最后一条消息之后停止继续发送消息。

注释： 在配置 IPv6 HSRP 时，用户必须在接口上启用 HSRP 第 2 版（HSRPv2）。

如何配置 HSRP

默认的 HSRP 配置

表 26：默认的 HSRP 配置

特性	默认设置
HSRP 版本	第 1 版
HSRP 组	未配置
备用组号	0
备用 MAC 地址	系统会分配：0000.0c07.acXX 其中 XX 是 HSRP 的组号
备用优先级	100
备用延迟	0（无延迟）
备用追踪接口优先级	10
备用 hello 时间	3 秒
备用抑制时间	10 秒

HSRP 配置指导方针

- HSRPv2 和 HSRPv1 是互斥的。HSRPv2 无法在一个接口上与 HSRPv1 实现互操作，反之亦然。
- 在配置过程中，指定接口必须是下列几种三层接口之一：
  - 路由模式端口：在接口配置模式下使用命令 **no switchport** 配置的物理端口；

- SVI：在全局配置模式下使用命令 **interface vlan** *vlan_id* 创建出来的 VLAN 接口，默认是三层接口；
- 三层模式的 EtherChannel 端口：在全局配置模式下使用命令 **interface port-channel** *port-channel-number* 创建出来，并且在组中绑定了以太网接口的一种 port-channel 逻辑接口。

- 用户必须给所有三层接口都配置 IP 地址；
- 如果用户在一个接口上修改了 HSRP 的版本，每个 HSRP 组都会重置，因为 HSRP 组现在有一个新的虚拟 MAC 地址；
- 有效的组号与无效的组号包括：
- 假设用户配置的组号是十进制数值 2、150 和 225，那么就不能再给另一个组配置 3850 这个组号。因为 3850 不在 0 至 255 这个范围之内；
- 假设用户配置的组号是十进制数值 520、600 和 700，那么就不能再给另一个组配置 900 这个组号。因为 900 不在 512 至 767 这个范围之内；♀

① 通过十六进制判断合理的取值范围：十六进制数值 00 至 FF 对应十进制数值 0 至 255；十六进制数值 200 至 2FF 对应十进制数值 512 至 767。——译者注

启用 HSRP

接口配置命令 **standby ip** 会在所配置的接口上激活 HSRP 协议。如果用户设置了 IP 地址，那么设备就会将这个地址作为热备份组的指定地址。如果没有设置 IP 地址，那么设备就会通过备份功能来学习这个地址。用户必须至少在局域网中的一个三层接口上上配置了这个指定地址。如果 IP 地址一定会覆盖当前使用的指定地址。

如果用户在一个接口上配置了 **standby ip** 这条命令，同时启用了代理 ARP，那么如果接口的热备份状态为主用，那么设备就会用这个热备份组 MAC 地址来响应代理 ARP 请求。如果接口处于另一种状态下，那么设备就会抑制对代理 ARP 的响应。

总步骤

1. configure terminal

2. interface *interface-id*

3. standby version { **1** | **2** }

4. standby [*group-number*] **ip** [*ip-address* [**secondary**]]

5. end

6. show standby [*interface-id* [*group*]]

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Switch#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *interface-id* 示例： `Switch(config)#` **interface gigabitethernet1/0/1**	进入接口配置模式，并且进入要启用 HSRP 的那个接口
步骤 3	**standby version** { **1**	**2** }	（可选）在接口上配置 HSRP 的版本 • 1：选择 HSRPv1

	示例： `Switch(config-if)#` **standby version 1**	• 2：选择 HSRPv2 如果没有输入这条命令，或者没有设置关键字，那么接口默认运行的 HSRP 版本是 HSRPv1
步骤 4	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] 示例： `Switch(config-if)#` **standby 1 ip**	使用组号和虚拟 IP 地址来创建（或启用）HSRP 组。 • （可选）*group-number*：在接口上给启用的 HSRP 设置组号。取值范围是从 0 到 255；默认值为 0。如果只有一个 HSRP 组，不需要设置组号； • （除任一接口外其余接口可选）*ip-address*：热备份路由器接口的虚拟 IP 地址。用户必须在至少一个接口上输入虚拟 IP 地址；其他接口可以学习到这个地址； • （可选）**secondary**：该 IP 地址为热备份路由器接口的辅助地址。如果没有路由器被设置为辅助路由器或备份路由器，同时也没有设置优先级，那么设备就会比较主用 IP 地址，较高的 IP 地址就会成为主用路由器，次高为备用路由器
步骤 5	**end** 示例： `Switch(config-if)#` **end**	返回特权 EXEC 模式
步骤 6	**6 show standby** [*interface-id* [*group*]] 示例： `Switch #` **show standby**	验证备用组的配置
步骤 7	**copy running-config startup-config** 示例： `Switch#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 HSRP 优先级

接口配置命令 **standby priority**、**standby preempt** 和 **standby track** 的作用都是给寻找主用路由器和备用路由器设置特征，以及规定新的主用路由器在何种情况下会接管数据的转发功能。

在配置 HSRP 优先级时，应该按照下列指导方针：

- 分配优先级可以让用户选择主用路由器和备用路由器。如果用户启用了抢占功能，那么拥有较高优先级的路由器就会成为主用路由器。当优先级相等时，当前的主用路由器身份不会变化；
- 数值（1 到 255）最高即代表优先级最高（也即最有可能成为主用路由器）；
- 在设置优先级时、设置抢占、或者设置这两者时，用户必须至少设置一个关键字（**priority**、**preemtp** 或两者皆用）；

- 如果用户在一个接口上配置了命令 **standby track**，那么当路由器的另一个接口出现问题时，设备的优先级可能出现显著变化；
- 接口配置命令 standby track 会将路由器热备份优先级与接口的可用性进行绑定，这条命令在追踪那些没有配置 HSRP 的接口时十分有用。当一个被追踪接口出现故障时，配置了追踪的设备，其热备份优先级就会减 10。如果没有追踪接口，那么该接口的状态就不会影响设备的热备份优先级。对于每个配置了热备份的接口，用户都可以配置一个独立的追踪接口列表；
- 接口配置命令 **standby track** *interface-priority* 会指定当被追踪接口出现故障时，热备份优先级减少多少。当接口恢复时，优先级也会提升相同的数值；
- 当多个配置了 *interface-priority* 值的接口出现故障时，设备减少的优先级值也会累加。如果没有配置优先级值的被追踪接口出现故障，那么默认减少的数值就是 10，也不会累加；
- 当一个接口首次启用路由功能时，这个接口并没有完整的路由表。如果该接口配置了抢占功能，那么这个接口就会成为主用路由器，哪怕该接口无法提供充份的路由服务。要想避免出现这种问题，用户可以配置一个延迟时间，让路由器有时间更新自己的路由表。

用户可以从特权 EXEC 模式中，执行下列步骤在接口上配置 HSRP 优先级特征：

总步骤

1. configure terminal

2. interface *interface-id*

3. standby [*group-number*] **priority***priority*

4. standby [*group-number*] **preempt** [**delay** [**minimum***seconds*] [**reload***seconds*] [**sync***seconds*]]

5. standby [*group-number*] **track** *type number* [*interface-priority*]

6. end

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Switch#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *interface-id* 示例： `Switch(config)#` **interface gigabitethernet1/0/1**	进入接口配置模式，并且进入要设置优先级的那个接口
步骤 3	**standby** [*group-number*] **priority***priority* 示例： `Switch(config-if)#` **standby 120 priority 50**	设置选择主用路由器时使用的 **priority** 值。取值范围是从 1 到 255，默认值为 100。数值越高表示优先级越高： • （可选）*group-number*：这条命令应用的组号 在命令前面加上 no 这个关键字，可以让优先级恢复默认值
步骤 4	**standby** [*group-number*] **preempt** [**delay**	配置路由器执行 **preempt**（抢占），这表示当这台路由器比主用路由器的优先级更高时，它就会成为

	[**minimum***seconds*] [**reload***seconds*] [**sync***seconds*]] 示例： `Switch(config-if)#` **standby 1 preempt delay 300**	主用路由器： • （可选）*group-number*：这条命令应用的组号 • （可选）**delay minimum**：让本地路由器延迟指定的秒数再接管主用路由器的角色。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）； • （可选）**delay reload**：让本地路由器在重启后经历一段时间的延迟再接管主用路由器的角色。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）； • （可选）**delay sync**：让本地路由器延迟指定的秒数再接管主路由器的角色，让 IP 冗余客户端能够（使用 ok 或 wait）作出响应。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）。 在命令前面加上 no 这个关键字，可以让优先级恢复默认值
步骤 5	**standby** [*group-number*] **track** *type number* [*interface-priority*] 示例： `Switch(config-if)#` **standby track interface gigabitethernet1/1/1**	配置一个接口来对其他接口进行追踪，当其他接口出现故障时，主机的热备份优先级就会降低： • （可选）*group-number*：这条命令应用的组号； • *type*：输入要追踪的接口类型（与接口编号）； • *number*：输入要追踪的接口号（与接口类型）； • （可选）*interface-priority*：输入当接口发生故障或者恢复时，热备份优先级增减的数值。默认值为 10
步骤 6	**end** 示例： `Switch(config-if)#` **end**	返回特权 EXEC 模式
步骤 7	**show running-config**	验证备用组的配置
步骤 8	**copy running-config startup-config** 示例： `Switch#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 MHSRP

要启用 MHSRP 和负载分担，用户需要将两台路由器配置为它们所在组的主用路由器，而将虚拟路由器作为备用路由器，如多 HSRP 一节中的 MHSRP 负载分担一图所示。用户需要在每个 HSRP 接口的配置模式下输入命令 **standby preempt**，这样当一台路由器经历了故障并恢复之后，路由器就会执行抢占并且重新执行负载分担。

用户可以将路由器 A 配置为组 1 的主用路由器，将路由器 B 配置为组 2 的主用路由器。路由器 A 的 HSRP 接口 IP 地址为 10.0.0.1，其在组 1 的备用优先级为 110（默认为 100）。路由

器 B 的 HSRP 接口 IP 地址为 10.0.0.2，其在组 2 的备份优先级为 110。
组 1 使用的虚拟 IP 地址为 10.0.0.3，而组 2 的虚拟 IP 地址则为 10.0.0.4。
配置路由器 A
总步骤
1. configure terminal
2. interface *type number*
3. no switchport
4. ip address *ip-address mask*
5. standby [*group-number*] **ip** [*ip-address* [**secondary**]]
6. standby [*group-number*] **priority** *priority*
7. standby [*group-number*] **preempt** [**delay** [**minimum** *seconds*] [**reload** *seconds*] [**sync** *seconds*]]
8. standby [*group-number*] **ip** [*ip-address* [**secondary**]]
9. standby [*group-number*] **preempt** [**delay** [**minimum** *seconds*] [**reload** *seconds*] [**sync** *seconds*]]
10. end
11. show running-config
12. copy running-config startup-config
具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Switch#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *type number* 示例： `Switch(config)#` **interface gigabitethernet1/0/1**	配置接口类型并进入接口配置模式
步骤 3	**no switchport** 示例： `Switch (config)#` **no switchport**	将二层模式下的接口切换为三层模式，以执行三层的配置
步骤 4	**ip address** *ip-address mask* 示例： `Switch (config-if)#` **10.0.0.1 255.255.255.0**	给一个接口设置 IP 地址
步骤 5	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] 示例： `Switch (config-if)#` **standby 1 ip 10.0.0.3**	使用组号和虚拟 IP 地址来创建（或启用）HSRP 组。 • （可选）*group-number*：在接口上给启用的 HSRP 设置组号。取值范围是从 0 到 255；默认值为 0。如果只有一个 HSRP 组，不需要设置组号； • （除任一接口外其余接口可选）*ip-address*：热备份路由器接口的虚拟 IP 地址。用户必须在

		至少一个接口上输入虚拟 IP 地址；其他接口可以学习到这个地址； • （可选）**secondary**：该 IP 地址为热备份路由器接口的辅助地址。如果没有路由器被设置为辅助路由器或备份路由器，同时也没有设置优先级，那么设备就会比较主用 IP 地址，较高的 IP 地址就会成为主用路由器，次高为备用路由器
步骤 6	**standby** [*group-number*] **priority***priority* 示例： `Switch(config-if)#` **standby 1 priority 110**	设置选择主用路由器时使用的 **priority** 值。取值范围是从 1 到 255，默认值为 100。数值越高表示优先级越高： • （可选）*group-number*：这条命令应用的组号 在命令前面加上 no 这个关键字，可以让优先级恢复默认值
步骤 7	**standby** [*group-number*] **preempt** [**delay** [**minimum***seconds*] [**reload***seconds*] [**sync***seconds*]] 示例： `Switch(config-if)#` **standby 1 preempt delay 300**	配置路由器执行 **preempt**（抢占），这表示当这台路由器比主用路由器的优先级更高时，它就会成为主用路由器： • （可选）*group-number*：这条命令应用的组号 • （可选）**delay minimum**：让本地路由器延迟指定的秒数再接管主用路由器的角色。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）； • （可选）**delay reload**：让本地路由器在重启后经历一段时间的延迟再接管主用路由器的角色。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）； • （可选）**delay sync**：让本地路由器延迟指定的秒数再接管主路由器的角色，让 IP 冗余客户端能够（使用 ok 或 wait）作出响应。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）。 在命令前面加上 no 这个关键字，可以让优先级恢复默认值
步骤 8	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] 示例： `Switch (config-if)#` **standby 2 ip 10.0.0.4**	使用组号和虚拟 IP 地址来创建（或启用）HSRP 组。 • （可选）*group-number*：在接口上给启用的 HSRP 设置组号。取值范围是从 0 到 255；默认值为 0。如果只有一个 HSRP 组，不需要设置组号； • （除任一接口外其余接口可选）*ip-address*：热备份路由器接口的虚拟 IP 地址。用户必须在至少一个接口上输入虚拟 IP 地址；其他接口可以学习到这个地址； • （可选）**secondary**：该 IP 地址为热备份路由

		器接口的辅助地址。如果没有路由器被设置为辅助路由器或备份路由器，同时也没有设置优先级，那么设备就会比较主用 IP 地址，较高的 IP 地址就会成为主用路由器，次高为备用路由器
步骤 9	**standby** [*group-number*] **preempt** [**delay** [**minimum***seconds*] [**reload***seconds*] [**sync***seconds*]] 示例： `Switch(config-if)#` **standby 2 preempt delay 300**	配置路由器执行 **preempt**（抢占），这表示当这台路由器比主用路由器的优先级更高时，它就会成为主用路由器： • （可选）*group-number*：这条命令应用的组号 • （可选）**delay minimum**：让本地路由器延迟指定的秒数再接管主用路由器的角色。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）； • （可选）**delay reload**：让本地路由器在重启后经历一段时间的延迟再接管主用路由器的角色。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）； • （可选）**delay sync**：让本地路由器延迟指定的秒数再接管主路由器的角色，让 IP 冗余客户端能够（使用 ok 或 wait）作出响应。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）。 在命令前面加上 no 这个关键字，可以让优先级恢复默认值
步骤 10	**end** 示例： `Switch(config-if)#` **end**	返回特权 EXEC 模式
步骤 11	**show running-config**	验证备用组的配置
步骤 12	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置路由器 B

总步骤

1. configure terminal

2. interface *type number*

3. no switchport

4. ip address *ip-address mask*

5. standby [*group-number*] **ip** [*ip-address* [**secondary**]]

6. standby [*group-number*] **priority** *priority*

7. standby [*group-number*] **preempt** [**delay** [**minimum** *seconds*] [**reload** *seconds*] [**sync** *seconds*]]

8. standby [*group-number*] **ip** [*ip-address* [**secondary**]]

9. standby [*group-number*] **preempt** [**delay** [**minimum** *seconds*] [**reload** *seconds*] [**sync** *seconds*]]

10. end

11. show running-config

12. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *type number* 示例： `Switch(config)#` **interface gigabitethernet1/0/1**	配置接口类型并进入接口配置模式
步骤 3	**no switchport** 示例： `Switch (config)#` **no switchport**	将二层模式下的接口切换为三层模式，以执行三层的配置
步骤 4	**ip address** *ip-address mask* 示例： `Switch (config-if)#` **10.0.0.2 255.255.255.0**	给一个接口设置 IP 地址
步骤 5	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] 示例： `Switch (config-if)#` **standby 1 ip 10.0.0.3**	使用组号和虚拟 IP 地址来创建（或启用）HSRP 组。 • （可选）*group-number*：在接口上给启用的 HSRP 设置组号。取值范围是从 0 到 255；默认值为 0。如果只有一个 HSRP 组，不需要设置组号； • （除任一接口外其余接口可选）*ip-address*：热备份路由器接口的虚拟 IP 地址。用户必须在至少一个接口上输入虚拟 IP 地址；其他接口可以学习到这个地址； • （可选）**secondary**：该 IP 地址为热备份路由器接口的辅助地址。如果没有路由器被设置为辅助路由器或备份路由器，同时也没有设置优先级，那么设备就会比较主用 IP 地址，较高的 IP 地址就会成为主用路由器，次高为备用路由器
步骤 6	**standby** [*group-number*] **priority***priority* 示例： `Switch(config-if)#`	设置选择主用路由器时使用的 **priority** 值。取值范围是从 1 到 255，默认值为 100。数值越高表示优先级越高： • （可选）*group-number*：这条命令应用的组号在命令前面加上 no 这个关键字，可以让优先级恢

	standby 1 priority 110	复默认值
步骤 7	**standby** [*group-number*] **preempt** [**delay** [**minimum***seconds*] [**reload***seconds*] [**sync***seconds*]] 示例： `Switch(config-if)#` **standby 1 preempt delay 300**	配置路由器执行 **preempt**（抢占），这表示当这台路由器比主用路由器的优先级更高时，它就会成为主用路由器： • （可选）*group-number*：这条命令应用的组号 • （可选）**delay minimum**：让本地路由器延迟指定的秒数再接管主用路由器的角色。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）； • （可选）**delay reload**：让本地路由器在重启后经历一段时间的延迟再接管主用路由器的角色。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）； • （可选）**delay sync**：让本地路由器延迟指定的秒数再接管主路由器的角色，让 IP 冗余客户端能够（使用 ok 或 wait）作出响应。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）。 在命令前面加上 no 这个关键字，可以让优先级恢复默认值
步骤 8	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] 示例： **standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Switch (config-if)#` **standby 2 ip 10.0.0.4**	使用组号和虚拟 IP 地址来创建（或启用）HSRP 组。 • （可选）*group-number*：在接口上给启用的 HSRP 设置组号。取值范围是从 0 到 255；默认值为 0。如果只有一个 HSRP 组，不需要设置组号； • （除任一接口外其余接口可选）*ip-address*：热备份路由器接口的虚拟 IP 地址。用户必须在至少一个接口上输入虚拟 IP 地址；其他接口可以学习到这个地址； • （可选）**secondary**：该 IP 地址为热备份路由器接口的辅助地址。如果没有路由器被设置为辅助路由器或备份路由器，同时也没有设置优先级，那么设备就会比较主用 IP 地址，较高的 IP 地址就会成为主用路由器，次高为备用路由器
步骤 9	**standby** [*group-number*] **preempt** [**delay** [**minimum***seconds*] [**reload***seconds*] [**sync***seconds*]] 示例： `Switch(config-if)#`	配置路由器执行 **preempt**（抢占），这表示当这台路由器比主用路由器的优先级更高时，它就会成为主用路由器： • （可选）*group-number*：这条命令应用的组号 • （可选）**delay minimum**：让本地路由器延迟指定的秒数再接管主用路由器的角色。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）；

	standby 2 preempt delay 300	• （可选）**delay reload**：让本地路由器在重启后经历一段时间的延迟再接管主用路由器的角色。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）； • （可选）**delay sync**：让本地路由器延迟指定的秒数再接管主路由器的角色，让 IP 冗余客户端能够（使用 ok 或 wait）作出响应。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）。 在命令前面加上 no 这个关键字，可以让优先级恢复默认值
步骤 10	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 11	**show running-config**	验证备用组的配置
步骤 12	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 HSRP 认证与计时器

用户可以根据自己的需求来配置 HSRP 认证或者修改 Hello 时间间隔和抑制时间。

在配置这些属性，可以参照下面的指导方针：

- 所有 HSRP 消息中都会带有未加密的认证字符串。用户必须在一条线缆连接的每一台路由器和接入服务器上都配置相同的认证字符串，这样才能确保设备之间可以实现互操作。认证不匹配会让设备无法从其它 HSRP 设备那里学习到指定的热备份 IP 地址和计时器值。
- 那些没有配置备用计时器值的路由器或接入服务器可以从主用路由器或者备用路由器那里学习到计时器值。主用路由器上配置的计时器值永远会覆盖其他的计时器值设置。
- 热备份组中的所有路由器应该使用相同的计时器值。一般来说，*抑制时间*要大于等于 3 倍的 *hello 时间*。

用户可以从特权 EXEC 模式中，通过下列步骤在一个接口上配置 HSRP 认证和计时器：

总步骤

1. configure terminal

2. interface *interface-id*

3. standby [*group-number*] **authentication** *string*

4. standby [*group-number*] **timers** *hellotime holdtime*

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal**	进入全局配置模式

	示例： `Device#` **configure terminal**	
步骤 2	**interface** *type number* 示例： `Switch(config)#` **interface gigabitethernet1/0/1**	进入要设置优先级的那个接口的配置模式
步骤 3	**standby** [*group-number*] **authentication** *string* 示例： `Switch(config-if) #` **standby 1 authentication word**	• （可选）**authentication** *string*：输入所有 HSRP 消息中都要携带的字符串。认证字符串的长度最多为 8 字节，默认的字符串为 **inspur**； • （可选）*group-number*：这条命令应用的组号
步骤 4	**standby** [*group-number*] **timers** *hellotime holdtime* 示例： `Switch(config-if) #` **standby 1 timers 5 15**	（可选）设置发送 hello 数据包的时间间隔，以及其他路由器宣告主用路由器失效前经历的时间。 • *group-number*：这条命令应用的组号； • *hellotime*：让本地路由器延迟指定的秒数再接管主用路由器的角色。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）； • *holdtime*：让本地路由器在重启后经历一段时间的延迟再接管主用路由器的角色。取值范围是 0 到 3600 秒（即 1 小时），默认值为 0（即不加延迟直接接替主用路由器的角色）。
步骤 5	**end** 示例： `Device(config-if)#` **end**	返回特权 EXEC 模式
步骤 6	**show running-config**	验证备用组的配置
步骤 7	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

对 ICMP 重定向消息启用 HSRP 支持

配置了 HSRP 的接口会自动启用 ICMP 重定向消息。ICMP 是一种网络层协议，可以通过消息数据包来报告错误和其他与 IP 处理有关的信息。ICMP 可以提供诊断功能，包括将错误数据包发送给主机。这种特性可以通过 HSRP 过滤出站的 ICMP 重定向消息，因为下一跳 IP 地址可能会被修改为 HSRP 虚拟 IP 地址。要想了解详细信息，可以参阅 Inspur INOS IP 配置指南 12.4 版。

配置 HSRP 组和集群

当一台设备参与到了 HSRP 备用路由，同时这台设备启用了集群时，用户就可以使用相同的备用组来实现命令交换机（command switch）冗余和 HSRP 冗余。用户可以使用全局配置命令 **cluster standby-group** *HSRP-group-name* [**routing-redundancy**]让同一个 HSRP 组来提供命令交换机冗余和路由冗余。如果用户用同一个 HSRP 备份组的名称创建集群，但又没有输入

routing-redundancy 这个关键字，那么这个组就会禁用 HSRP 备份路由。

验证 HSRP

验证 HSRP 的配置

用户可以从特权 EXEC 模式中，通过这条命令来查看 HSRP 的设置：

show standby [*interface-id* [*group*]] [**brief**] [**detail**]

用户可以查看整台交换机上的 HSRP 信息，也可以查看某个接口、某个 HSRP 组，或者某个接口上的 HSRP 组的信息。用户可以指定是显示简洁的 HSRP 信息，还是显示详细的 HSRP 信息。系统默认显示的是详细信息（**detail**）。如果有大量的 HSRP 组，而用户在使用命令 show standby 时有没有指定限定符，那么系统就会显示大量的信息。

```
Switch #show standby
VLAN1 - Group 1
Local state is Standby, priority 105, may preempt
Hellotime 3 holdtime 10
Next hello sent in 00:00:02.182
Hot standby IP address is 172.20.128.3 configured
Active router is 172.20.128.1 expires in 00:00:09
Standby router is local
Standby virtual mac address is 0000.0c07.ac01
Name is bbb
VLAN1 - Group 100
Local state is Standby, priority 105, may preempt
Hellotime 3 holdtime 10
Next hello sent in 00:00:02.262
Hot standby IP address is 172.20.138.51 configured
Active router is 172.20.128.1 expires in 00:00:09
Active router is local
Standby router is unknown expired
Standby virtual mac address is 0000.0c07.ac64
Name is test
```

配置 HSRP 的配置案例

启用 HSRP：示例

这个示例显示了如何在接口上对组 1 激活 HSRP。热备份组启用的 IP 地址会使用 HSRP 学习到。

注释： 这个流程是启用 HSRP 的最基本配置。其他的都是可选配置。

```
Switch # configure terminal
Switch(config) # interface gigabitethernet1/0/1
Switch(config-if)# no switchport
Switch(config-if)# standby 1 ip
Switch(config-if)# end
```

```
Switch # show standby
```

配置 HSRP 优先级：示例

这个示例激活了一个端口，给端口设置了一个 IP 地址，并且将优先级设置为了 120（高于默认值），同时让它在成为主用路由器之前等待 300 秒（5 秒）：

```
Switch # configure terminal
Switch(config) # interface gigabitethernet1/0/1
Switch(config-if)# no switchport
Switch(config-if)# standby ip 172.20.128.3
Switch(config-if)# standby priority 120 preempt delay 300
Switch(config-if)# end
Switch # show standby
```

配置 MHSRP：示例

这个示例显示了如何在图 MHSRP 负载分担中完成 MHSRP 的配置：

路由器 A 的配置

```
Switch # configure terminal
Switch(config) # interface gigabitethernet1/0/1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.0.0.1 255.255.255.0
Switch(config-if)# standby ip 10.0.0.3
Switch(config-if)# standby 1 priority 110
Switch(config-if)# standby 1 preempt
Switch(config-if)# standby 2 ip 10.0.0.4
Switch(config-if)# standby 2 preempt
Switch(config-if)# end
```

路由器 B 的配置

```
Switch # configure terminal
Switch(config) # interface gigabitethernet1/0/1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.0.0.2 255.255.255.0
Switch(config-if)# standby ip 10.0.0.3
Switch(config-if)# standby 1 preempt
Switch(config-if)# standby 2 ip 10.0.0.4
Switch(config-if)# standby 1 priority 110
Switch(config-if)# standby 2 preempt
Switch(config-if)# end
```

配置 HSRP 认证与计时器：示例

这个示例显示了如何将 word 配置为组 1 中热备份路由器之间相互认证的字符串：

```
Switch # configure terminal
Switch(config) # interface gigabitethernet1/0/1
Switch(config-if)# no switchport
Switch(config-if)# standby 1 authentication word
Switch(config-if)# end
```

这个示例显示了如何在备用组 1 中，将 hello 数据包的时间间隔计时器设置为 5 秒，同时将认为一台路由器故障的时间设置为 15 秒：

```
Switch # configure terminal
Switch(config) # interface gigabitethernet1/0/1
Switch(config-if)# no switchport
Switch(config-if)# standby 1 ip
Switch(config-if)# standby 1 timers 5 15
Switch(config-if)# end
```

配置 HSRP 组与集群：示例

这个示例显示了如何将备用组 my_hsrp 与集群绑定，用同一个 HSRP 组提供命令交换机冗余和路由器冗余。这条命令只能在集群的命令交换机上配置。如果备用组名或备用组编号不存在，或者这台交换机知识集群的成员交换机，那么系统就会显示一条错误消息：

```
Switch # configure terminal
Switch(config) # cluster
```

配置 HSRP 的其他参考资料

相关文档

相关主题	文档名
Inspur INOS 命令	《Inspur INMOS 管理命令集，适用所有版本》

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com

标准与 RFC

标准/RFC	标题
RFC 2281	Inspur 热备份路由器协议

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

关于配置 HSRP 的特性信息

表 27：关于配置 HSRP 的特性信息

版本	修改
11.3.1	引入该特性

下一跳解析协议（NHRP）是一种类似于地址解析协议（ARP）的协议，NHRP 可以动态映射一个非组播多路访问网络，而无需再由用户手动配置所有的隧道端点。通过 NHRP，连接在 NBMA 网络中的系统可以动态学习网络中其他系统的 NBMA（物理）地址，让这些系统可以直接相互通信。这个协议提供了一种 ARP 式的解决方案，让工作站的数据链路地址可以动态解析出来。

NHRP 是一种客户端服务器协议，其中心节点是下一跳服务器（NHS，Next Hop Server），而分支节点则是下一跳客户端（NHC，Next Hop Client）。中心节点会维护一个包含每个分支节点公共接口地址的 NHRP 数据库。各个分支节点会在启动时将自己的非 NBMA（真实）地址注册到这个数据库中，并且会向 NHRP 数据库查询目的分支节点的地址，以建立直连的隧道。在文档的这一部分中，我们会解释如何通过 GRE（通用路由封装）配置 NHRP。在 Inspur INOS XE Denali 16.3.1 中，NHRP 只支持分支站点配置。

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于配置 NHRP 的信息

NHRP 与 NBMA 网络的相互关系

大多数 WAN 网络都是一系列的点到点链路。虚拟隧道网络（例如通用路由封装[GRE]隧道）也是一系列点到点链路。为了高效地扩展这些点到点链路的连通性，这些链路往往会被划分为一个网络，或者一个多层星型网络。多点接口（如 GRE 隧道接口）可以减少这类网络中枢心路由器上的配置。这样的网络就是一个 NBMA 网络。

由于通过一个多点接口可以访问多条隧道的端点，因此必须把逻辑隧道端点 IP 地址映射为物理隧道端点 IP 地址才能将数据包从这个连接 NBMA 网络的接口转发出去。这个映射关系固然可以是管理员静态配置的，但最好能够动态发现或者学习。

NHRP 是一种 ARP 式的协议，它可以解决 NBMA 网络中的这类问题。通过 NHRP，连接到

NBMA 网络的系统可以动态学习到网络中其他系统的 NBMA 地址，让这些系统可以直接相互通信，而不需要让流量穿过中间跳设备。

路由器、访问服务器和主机都可以使用 NHRP 来发现其他连接在 NBMA 网络中的路由器和主机的地址。部分互联的 NBMA 网络往往在这个 NBMA 网络背后有多个逻辑网络。在这种配置环境中，穿越 NBMA 网络的数据包可能会在 NBMA 网络中经历多跳，才能到达出站路由器（也就是最接近目的网络的路由器）。

NHRP 注册功能可以支持下列 NBMA 网络：

- NHRP 注册：NHRP 可以让下一跳客户端（NHC）动态注册到下一跳服务器（NHS）上。这种注册功能可以让 NHC 加入到 NBMA 网络中，而不需要在 NHS 上对配置进行任何修改，如果 NHC 有一个动态物理 IP 地址，或者 NHC 在一台动态修改物理 IP 地址的 NAT（网络地址转换）路由器身后，那么这种注册功能就格外好用。因为在上述情况下，人们几乎无法在 NHS 上提前修改这台 NHC 的逻辑物理地址间的映射关系。

动态建立的星型网络

NBMA 网络最初是一个星型网络，有一层又一层的分支节点 NHC 与中心节点的 NHS 相连。NHC 上要配置去往 NHS 的静态映射信息，会连接到其 NHS 并且向 NHS 发送一条 NHRP 注册消息。这种配置可以让 NHS 动态学习分支站点的映射信息，减少中心站点上所需的配置，让分支站点能够获取动态的 NBMA（物理）IP 地址。

如何配置 NHRP

在接口上启用 **NHRP**

下面的工作是在一台交换机上给一个接口启用 NHRP。总地来说，所有在一个逻辑 NBMA 网络中的 NHRP 工作站上都应该配置相同的网络标识符。

NHRP 网络 ID 的作用是给 NHRP 接口定义 NHRP 域，并且当同一个 NHRP 节点（交换机）上有两个或多个 NHRP 域（GRE 隧道接口）时，NHRP 网络 ID 可以用来区分不同的 NHRP 域或网络。当同一台交换机上配置了两个 NHRP 网络（云）时，NHRP 网络 ID 可以让这两个 NHRP 网络相互隔离。

NHRP 网络 ID 是一个纯本地参数。它仅具有交换机本地意义，并不会通过 NHRP 数据包传输给其他的 NHRP 节点。有鉴于此，即使两台交换机处于同一个 NHRP 域中，一台交换机上配置 NHRP 网络 ID 值也不必与另一台交换机的 NHRP 网络 ID 相匹配。当 NHRP 数据包到达一个 GRE 接口时，它们就会分配到本地 NHRP 域中给这个接口配置的 NHRP 网络 ID。

我们推荐用户给同一个 NHRP 网络中，所有交换机的 GRE 接口都配置相同的 NHRP 网络 ID。这样有助于用户判断哪些 GRE 接口是哪些 NHRP 网络中的成员。

在一台交换机上，每个 GRE 隧道接口都可以有一个唯一的 NHRP 域（网络 ID）。NHRP 域可以沿着一条路由，跨越 GRE 隧道接口进行扩展。此时，在 GRE 隧道接口上使用相同的 NHRP 网络 ID 可以让两个 GRE 接口融合为一个 NHRP 网络。

总步骤

1. enable

2. configure terminal

3. interface *type number*

4. ip address *ip-address network-mask*

5. ip nhrp network-id *number*

6. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Switch> enable`	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** 示例： `Switch# configure terminal`	进入全局配置模式

步骤 3	**interface** *type number*
	
	示例：
	`Switch(config)# interface tunnel 100`
	`Device(config-if)#`
		配置一个接口，并且进入接口的配置模式

步骤 4	**ip address** *ip-address network-mask* 示例： `Switch(config-if)# ip address 10.0.0.1 255.255.255.0`	启用 IP 并且给接口配置 IP 地址
步骤 5	**ip nhrp network-id** *number* 示例： `Switch(config-if)# ip nhrp network-id 1`	在接口上启用 NHRP
步骤 6	**end** 示例： `Switch(config)# end`	离开接口配置模式并返回特权 EXEC 模式

给多点操作配置 GRE 隧道

下面的工作是给多点操作环境（NBMA）配置一条 GRE 隧道。

多点隧道接口的隧道网络可以理解为是一个 NBMA 网络。当一台交换机上配置了多条 GRE 隧道时，这些隧道就必须有唯一的隧道 ID 值或者唯一的隧道源地址。

总步骤

1. enable
2. configure terminal
3. interface *type number*
4. ip address *ip-address*
5. ip mtu *bytes*
6. ip pim sparse-dense-mode
7. ip nhrp map *ip-address nbma-address*
8. ip nhrp map multicast *nbma-address*
9. ip nhrp network-id *number*
10. ip nhrp nhs *nhs-address*
11. tunnel source vlan *interface-number*
12. tunnel destination *ip-address*
13. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Switch> enable`	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** 示例： `Switch# configure terminal`	进入全局配置模式
步骤 3	**interface** *type number* 示例： `Switch(config)# interface tunnel 100`	配置一个接口，并且进入接口的配置模式
步骤 4	**ip address** *ip-address network-mask* 示例： `Switch(config-if)# ip address 172.16.1.1 255.255.255.0`	启用 IP 并且给接口配置 IP 地址
步骤 5	**ip mtu** *bytes* 示例： `Switch(config-if)# ip mtu 1400`	设置这个接口发送数据包的最大传输单元（MTU）
步骤 6	**ip pim sparse-dense-mode** 示例：	在接口上启用协议独立组播（PIM）并将接口执行稀疏模式或密集模式的操作，具体选择取决于这个组播组的工作模式

	Switch(config-if)# ip pim sparse-dense-mode	
步骤 7	**ip nhrp map** *ip-address nbma-address* 示例： Switch(config-if)# ip nhrp map 172.16.1.2 10.10.10.2	静态配置 IP 与 NBMA（非广播多路访问）地址的映射关系。 • *ip-address*：通过 NBMA 网络可达的目的 IP 地址。这个地址会被映射到 NBMA 地址； • *nbma-address*：通过 NBMA 网络达的 NBMA 地址。这个地址的格式会根据使用的媒介而定。比如，ATM 为网络服务接入点（NSAP）地址。以太网是 MAC 地址，而 SMDS（交换式多兆位数据服务）为 E.164 地址。这个地址会被映射到 IP 地址；
步骤 8	**ip nhrp map multicast** *nbma-address* 示例： Switch(config-if)# ip nhrp map multicast 10.10.10.2	配置用来充当那些要通过隧道网络进行发送的广播或组播数据包的目的地址的非广播多路访问（NBMA）地址
步骤 9	**ip nhrp network-id** *number* 示例： Switch(config-if)# ip nhrp network-id 1	在接口上启用下一跳解析协议（NHRP） • *number*：来自非广播多路访问网络（NBMA）的全局唯一 32 位网络 ID。取值范围是从 1 到 4294967295
步骤 10	**ip nhrp nhs** *nhs-address* 示例： Switch(config-if)# ip nhrp nhs 172.16.1.2	设置一个或多个 NHRP 服务器的地址。 • *nhs-address*：设置下一跳服务器的地址
步骤 11	**tunnel source vlan** *interface-number* 示例： Switch(config-if)# tunnel source vlan 1	设置隧道接口的源地址
步骤 12	**tunnel destination** *ip-address* 示例： Switch(config-if)# tunnel destination 10.10.10.2	设置隧道接口的目的地址
步骤 13	**end**	离开接口配置模式并返回特权 EXEC 模式

	示例： `Switch(config-if)# end`	

NHRP 的配置示例

逻辑 NBMA 的物理网络设计示例

逻辑 NBMA 网络可以理解为是参与 NHRP 并且拥有相同网络标识符的一组接口与主机。下图显示了一个物理 NBMA 网络之上的两个 NBMA 网络（每个圆圈为一个 NBMA 网络）。路由器 A 可以与路由器 B 和路由器 C 进行通信，因为它们共享相同的网络标识符（2）。路由器 C 也可以和路由器 D 与路由器 E 通信，因为它们也共享相同的网络标识符（7）。在完成地址解析之后，路由器 A 可以用 1 跳向路由器 C 发送 IP 数据包，而路由器 C 则可以再用 1 跳将这些数据包发送给路由器 E，如下图虚线部分所示。

图 17：一个物理 NBMA 网络之上建立的两个逻辑 NBMA 网络

Destination host	目的主机
Router E	路由器 E
Router D	路由器 D
Router C	路由器 C
Router B	路由器 B
Router A	路由器 A
Source host	源主机
Statically configured tunnel endpoints or permanent virtual circuits	静态配置的隧道端点或永久虚电路
Dynamically created virtual circuits	动态创建的虚电路

上图中 5 台路由器的物理拓扑有可能如下图所示，其中源主机与路由器 A 直连，而目的主机与路由器 E 直连。同一台交换机连接着所有这 5 台路由器，这样建立起一个物理的 NBMA 网络。

图 18：示例 NBMA 网络的物理拓扑

Destination host	目的主机
Router E	路由器 E
Router D	路由器 D
Router C	路由器 C
Router B	路由器 B
Router A	路由器 A
Source host	源主机

请再次观察上图。首先，在 NHRP 解析出 NBMA 地址之前，从源主机发往目的主机的 IP 数据包会穿越 5 台与交换机相连的路由器，才最终到达目的设备。当路由器 A 第一次向目的主机发送 IP 数据包时，路由器 A 也会向该目的主机的 IP 地址生成一个 NHRP 请求消息。这个请求消息会被转发给路由器 C，路由器 C 因此会生成一个响应消息。之所以路由器 C 会作出响应，是因为它是两个逻辑 NBMA 网络的出站路由器。

同样，路由器 C 也会生成一个自己的 NHRP 请求消息，而路由器 E 则会对这个消息作出响

应。在这个示例中，源和目的之间后续的 IP 流量会需要通过 2 跳穿越 NBMA 网络，因为 IP 流量必须在两个逻辑 NBMA 网络之间进行转发。如果没有把这个 NBMA 网络分为两部分，那么数据包 1 跳就可以到达目的地。

多点操作 GRE 隧道的示例

通过多点隧道，一个隧道接口可以与多台邻居交换机相连。与点到点隧道不同之处在于，用户需要对隧道目的进行配置。实际上，在配置时，隧道目的必须对应一个 IP 组播地址。
在下面的示例中，交换机 A 和交换机 B 共享了同一个以太网段。我们通过多点隧道网络配置了最基本的连通性,通过这种方式创建了一个部分互联网的 NBMA 网络。通过静态的 NHRP 映射条目，交换机 A 知道如何访问交换机 B，反之亦然。
下面的示例显示了如何配置 GRE 多点隧道：

交换机 A 的配置：

```
Switch(config)# interface tunnel 100 !为PIM流量配置隧道接口
Switch(config-if)# no ip redirects
Switch(config-if)# ip address 192.168.24.1 255.255.255.252
Switch(config-if)# ip mtu 1400
Switch(config-if)# ip pim sparse-dense-mode
Switch(config-if)# ip nhrp map 192.168.24.3 172.16.0.1 !用户可以根据需要配置NHRP，让它动态发现隧道端点
Switch(config-if)# ip nhrp map multicast 172.16.0.1
Switch(config-if)# ip nhrp network-id 1
Switch(config-if)# ip nhrp nhs 192.168.24.3
Switch(config-if)# tunnel source vlan 1
Switch(config-if)# tunnel destination 172.16.0.1
Switch(config-if)# end
```

交换机 B 的配置：

```
Switch(config)# interface tunnel 100
Switch(config-if)# no ip redirects
Switch(config-if)# ip address 192.168.24.2 255.255.255.252
Switch(config-if)# ip mtu 1400
Switch(config-if)# ip pim sparse-dense-mode
Switch(config-if)# ip nhrp map 192.168.24.4 10.10.0.3
Switch(config-if)# ip nhrp map multicast 10.10.10.3
Switch(config-if)# ip nhrp network-id 1
Switch(config-if)# ip nhrp nhs 192.168.24.4
Switch(config-if)# tunnel source vlan 1
Switch(config-if)# tunnel destination 10.10.10.3
Switch(config-if)# end
```

配置 NHRP 的其他参考资料

相关文档

相关主题	文档名
Inspur 6650 配置	
Inspur 3850 配置	

标准与 RFC

标准/RFC	标题
RFC 2332	NBMA 下一跳解析协议（NHRP）

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

关于配置 NHRP 的特性信息

下表提供了关于这部分文档所描述的版本信息。这个表仅罗列了在一个版本系列中引入这个特性的具体软件版本。如无特别说明，这个软件版本系列的后续版本同样支持这个特性。用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

表 28：关于配置 NHRP 的特性信息

特性名	版本	特性信息
下一跳解析协议	Inspur INOS XE Polaris 16.3.1	下一跳解析协议（NHRP）是一种地址解析协议（ARP）类的协议，它可以动态映射非广播多路访问网络（NBMA），而不需要用户手动配置所有隧道端点。通过 NHRP，与 NBMA 网络相连的系统可以动态学习网络中其他系统的 NBMA（物理）地址，这些系统由此可以实现直接通信

VRRPv3 协议支持

VRRPv3 协议支持

虚拟路由器冗余协议（VRRP）可以让一组设备称为一个虚拟设备，以此来提供冗余。接下来，LAN 客户端可以将这台虚拟设备配置为它们的网关。虚拟设备代表的是一组设备，这组设备也称为 VRRP 组。VRRP 第 3 版协议支持特性可以支持 IPv4 地址和 IPv6 地址，而 VRRP 第 2 版只支持 IPv4 地址。这部分文档会解释与 VRRPv3 相关的概念，并且介绍如何在网络中创建和自定义一个 VRRP 组。使用 VRRPv3 协议支持的好处包括：

- 在多厂商环境中实现互操作；
- VRRPv3 支持使用 IPv4 地址和 IPv6 地址，而 VRRPv2 只支持 IPv4 地址；
- 通过 VRRS 路径（Pathway）提升了协议的扩展性。

注释： 在这部分文档中，我们会混用 VRRP 和 VRRPv3 这两个词。

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

VRRPv3 协议支持的限制条件

- VRRPv3 的目的不是为了替代任何现有的动态协议。VRRPv3 旨在用于那些拥有多路访问、广播或组播功能的以太网环境中；
- 以太网、快速以太网、BVI（桥组虚拟接口）和吉比特以太网接口都支持 VRRPv3；多协议标签交换（MPLS）虚拟专用网络（VPN）、可感知 VRF 的 MPLS VPN 和 VLAN 也都支持 VRRPv3；
- 由于 BVI 接口启动时相关的转发延迟，用户一定不能将 VRRPv3 的通告计时器配置得大于等于 BVI 接口的转发延迟，这样设置会让刚刚启动的 BVI 接口所连接的 VRRP 设备无法接替主用设备的角色。用户可以使用命令 **bridge forward-time** 来设置 BVI 接口的转发延迟。可以使用命令 **vrrp timers advertise** 来设置 VRRP 通告计时器；
- VRRPv3 不支持状态换自动切换（SSO，Stateful Switchover）；
- 只有当 VRRP 通过 VRRS 路径冗余接口相同的网络进行工作时，才有可能实现全网络冗余。要想实现全网络冗余，需要注意下面的限制条件：

- VRRS 路径不应该与父 VRRP 组共享不同的物理接口，也不应该在与父 VRRP 组不同的物理接口的子接口上进行配置；
- 只要 VLAN 不与配置父 VRRP 组的 VLAN 共享相同 trunk 链路，VRRP 路径不应该配置在这个 VLAN 的交换虚拟接口（SVI）上；

关于 VRRPv3 协议支持的信息

VRRPv3 的优点

支持 IPv4 和 IPv6

VRRPv3 支持使用 IPv4 地址和 IPv6 地址，而 VRRPv2 只支持 IPv4 地址。

注释： 在使用 VRRPv3 时，就不能使用 VRRPv2。要想配置 VRRPv3，需要在全局配置模式下输入命令 **fhrp version vrrp**。

冗余

VRRP 可以让用户将多台设备配置为默认网关设备，这可以降低网络因单点故障而无法通信的几率。

负载分担

用户可以用这样一种方式配置 VRRP，即往返于 LAN 客户端的流量可以由多台设备共享，这样设备间就可以更加公平地共享流量负载了。

多虚拟设备

VRRP 支持最多在一台设备物理接口上配置 255 台虚拟设备，这限制了 VRRP 的扩展。多虚拟设备支持可以让用户在自己的 LAN 拓扑中部署冗余和负载分担。在扩展的环境中，VRRS 路径应该与 VRRP 控制组结合起来使用。

多 IP 地址

虚拟设备可以管理多个 IP 地址，包括辅助 IP 地址。因此，如果一个以太网接口上配置了多个子网，那么用户可以给每个子网配置 VRRP。

注释： 要在 VRRP 组中使用辅助 IP 地址，同一个组中配置配置主用地址。

抢占

VRRP 的冗余功能让可以让一台拥有高优先级、刚刚成为可用设备的备份虚拟设备，从一台之前接替了主用虚拟设备的备份虚拟设备那里抢占主用设备的角色。

注释： 在低优先级主用设备上启用抢占功能时，可以设置一个延迟值。

通告协议

VRRP 使用专门的 IANA（互联网数字分配机构）标准组播地址来执行 VRRP 通告。对于 IPv4，这个组播地址是 224.0.0.18。对于 IPv6，这个组播地址是 FF02:0:0:0:0:0:0:12。这种编址方式减少了必须提供组播服务的设备数量，而且可以在一个网段上测试设备，以准确地识别 VRRP 数据包。IANA 分配给 VRRP 的 IP 协议号是 112。

VRRP 设备优先级与抢占

VRRP 冗余方案中的一大重要组成部分是 VRRP 设备优先级。优先级决定了每个 VRRP 设备的角色，以及当虚拟主设备出现故障时的情形。

如果一台 VRRP 设备拥有虚拟设备的 IP 地址和物理接口的 IP 地址，那么这台设备就会充当虚拟主设备。

优先级也决定了一台 VRRP 设备是虚拟备份设备还是虚拟主设备，以及当虚拟主设备出现故障时，其他设备集成虚拟主设备的次序。用户可以对每台虚拟备份设备进行配置，使用命令 **priority** 赋予它们一个从 1 到 254 之间数值（使用命令 **vrrp address-family** 进入 VRRP 配置模式，然后输入 **priority** 可选项）。

例如，如果一台设备 A（LAN 拓扑中的虚拟主设备）发生了故障，其他设备通过选举判断虚拟备份设备 B 还是虚拟备份设备 C 来接替设备 A 成为虚拟主设备。如果用户分别给设备 B 和设备 C 配置的优先级是 101 和 100，那么设备 B 就会被选举为虚拟主设备，因为设备 B 的优先级更高。如果设备 B 和设备 C 上配置的优先级都是 100，那么拥有更高 IP 地址的那台虚拟备份设备就会被选举为虚拟主设备。

在默认情况下，如果启用了抢占功能，那么当一台高优先级设备接入环境中，就会接替刚刚被选举为虚拟主设备的虚拟备份设备。用户可以使用命令 **no preempt** 来禁用抢占功能（使用命令 vrrp address-family 进入 VRRP 配置模式，然后再输入 **no preempt** 命令）。如果禁用了抢占功能，那么被选举为虚拟主设备的虚拟备份设备，即使在原本的虚拟主设备恢复之后，还是会一直保留主设备的身份。

注释： 在低优先级主用设备上启用抢占功能时，可以设置一个延迟值。

VRRP 通告

虚拟主设备会向同一个组中的其他 VRRP 设备发送 VRRP 通告。通告中会设计优先级与虚拟主设备的状态。VRRP 通告可以（根据 VRRP 组的配置）被封装到 IPv4 数据包或 IPv6 数据包当中，并且发送给分配给对应 VRRP 组的组播地址。对于 IPv4，这个组播地址是 224.0.0.18。对于 IPv6，这个组播地址是 FF02:0:0:0:0:0:0:12。在默认情况下，通告会每秒发送一次，这个时间间隔是可以配置的。

Inspur 设备可以以毫秒为单位配置计时器，这是与 VRRPv2 的区别。用户需要手动在主用设备和备份设备上配置以毫秒为单位的计时器值。但在备用设备上，命令 **show vrrp** 显示的主用通告值永远是 1，因为备用设备上的数据包不接受毫秒值。

用户一定要在绝对必要的情况再使用毫秒计时器，使用时务虑考虑仔细且经过周密的测试。毫秒值只有在有利的环境下才会生效，使用毫秒计时器值可以兼容第三方厂商，只要这个第三方厂商支持 VRRPv3。用户可以将计时器值设置为 100 毫秒到 40000 毫秒之间的某个数值。

关于 VRRPv3 协议支持的信息

VRRPv3 的优点

支持 IPv4 和 IPv6

VRRPv3 支持使用 IPv4 地址和 IPv6 地址，而 VRRPv2 只支持 IPv4 地址。

注释： 在使用 VRRPv3 时，就不能使用 VRRPv2。要想配置 VRRPv3，需要在全局配置模式下输入命令 **fhrp version vrrp**。

冗余

VRRP 可以让用户将多台设备配置为默认网关设备，这可以降低网络因单点故障而无法通信的几率。

负载分担

用户可以用这样一种方式配置 VRRP，即往返于 LAN 客户端的流量可以由多台设备共享，这样设备间就可以更加公平地共享流量负载了。

多虚拟设备

VRRP 支持最多在一台设备物理接口上配置 255 台虚拟设备，这限制了 VRRP 的扩展。多虚拟设备支持可以让用户在自己的 LAN 拓扑中部署冗余和负载分担。在扩展的环境中，VRRS 路径应该与 VRRP 控制组结合起来使用。

多 IP 地址

虚拟设备可以管理多个 IP 地址，包括辅助 IP 地址。因此，如果一个以太网接口上配置了多个子网，那么用户可以给每个子网配置 VRRP。

注释： 要在 VRRP 组中使用辅助 IP 地址，同一个组中配置配置主用地址。

抢占

VRRP 的冗余功能让可以让一台拥有高优先级、刚刚成为可用设备的备份虚拟设备，从一台之前接替了主用虚拟设备的备份虚拟设备那里抢占主用设备的角色。

注释： 在低优先级主用设备上启用抢占功能时，可以设置一个延迟值。

通告协议

VRRP 使用专门的 IANA（互联网数字分配机构）标准组播地址来执行 VRRP 通告。对于 IPv4，这个组播地址是 224.0.0.18。对于 IPv6，这个组播地址是 FF02:0:0:0:0:0:0:12。这种编址方式减少了必须提供组播服务的设备数量，而且可以在一个网段上测试设备，以准确地识别 VRRP 数据包。IANA 分配给 VRRP 的 IP 协议号是 112。

VRRP 设备优先级与抢占

VRRP 冗余方案中的一大重要组成部分是 VRRP 设备优先级。优先级决定了每个 VRRP 设备的角色，以及当虚拟主设备出现故障时的情形。

如果一台 VRRP 设备拥有虚拟设备的 IP 地址和物理接口的 IP 地址，那么这台设备就会充当虚拟主设备。

优先级也决定了一台 VRRP 设备是虚拟备份设备还是虚拟主设备，以及当虚拟主设备出现故障时，其他设备集成虚拟主设备的次序。用户可以对每台虚拟备份设备进行配置，使用命令 **priority** 赋予它们一个从 1 到 254 之间数值（使用命令 **vrrp address-family** 进入 VRRP 配置模式，然后输入 **priority** 可选项）。

例如，如果一台设备 A（LAN 拓扑中的虚拟主设备）发生了故障，其他设备通过选举判断虚拟备份设备 B 还是虚拟备份设备 C 来接替设备 A 成为虚拟主设备。如果用户分别给设备 B 和设备 C 配置的优先级是 101 和 100，那么设备 B 就会被选举为虚拟主设备，因为设备 B 的优先级更高。如果设备 B 和设备 C 上配置的优先级都是 100，那么拥有更高 IP 地址的那台虚拟备份设备就会被选举为虚拟主设备。

在默认情况下，如果启用了抢占功能，那么当一台高优先级设备接入环境中，就会接替刚刚被选举为虚拟主设备的虚拟备份设备。用户可以使用命令 **no preempt** 来禁用抢占功能（使用命令 vrrp address-family 进入 VRRP 配置模式，然后再输入 **no preempt** 命令）。如果禁用了抢占功能，那么被选举为虚拟主设备的虚拟备份设备，即使在原本的虚拟主设备恢复之后，还是会一直保留主设备的身份。

注释： 在低优先级主用设备上启用抢占功能时，可以设置一个延迟值。

VRRP 通告

虚拟主设备会向同一个组中的其他 VRRP 设备发送 VRRP 通告。通告中会设计优先级与虚拟主设备的状态。VRRP 通告可以（根据 VRRP 组的配置）被封装到 IPv4 数据包或 IPv6 数据包当中，并且发送给分配给对应 VRRP 组的组播地址。对于 IPv4，这个组播地址是 224.0.0.18。对于 IPv6，这个组播地址是 FF02:0:0:0:0:0:0:12。在默认情况下，通告会每秒发送一次，这个时间间隔是可以配置的。

Inspur 设备可以以毫秒为单位配置计时器，这是与 VRRPv2 的区别。用户需要手动在主用设备和备份设备上配置以毫秒为单位的计时器值。但在备用设备上，命令 **show vrrp** 显示的主用通告值永远是 1，因为备用设备上的数据包不接受毫秒值。

用户一定要在绝对必要的情况再使用毫秒计时器，使用时务虑考虑仔细且经过周密的测试。毫秒值只有在有利的环境下才会生效，使用毫秒计时器值可以兼容第三方厂商，只要这个第三方厂商支持 VRRPv3。用户可以将计时器值设置为 100 毫秒到 40000 毫秒之间的某个数值。

如何配置 VRRPv3 协议支持

GLBP 的启用与验证

下面的工作是给在一个接口上启用 GLBP，并且对其配置和操作进行验证。GLBP 的配置比较简单。GLBP 组中的每台网关上都必须配置相同的组号，在 GLBP 组中至少有一台网关上必须配置上这个组使用的虚拟 IP 地址。其他参数都可以通过学习来获得。

在开始前

如果一个接口上使用了 VLAN，那么每个 VLAN 的 GLBP 组号必须不同。

总步骤

1. enable

2. configure terminal

3. interface *type number*

4. ip address *ip-address mask* [**secondary**]

5. glbp *group* **ip** [*ip-address* [**secondary**]]

6. end

7. show glbp [*interface-type interface-number*] [*group*] [*state*] [**brief**]

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *type number* 示例： `Device(config)# interface GigabitEthernet 1/0/1`	配置一个接口，并且进入接口的配置模式
步骤 4	**ip address** *ip-address mask* [**secondary**] 示例： `Device(config-if)# ip address 10.21.8.32 255.255.255.0`	设置接口的主用 IP 地址或备用 IP 地址
步骤 5	**glbp** *group* **ip** [*ip-address* [**secondary**]]	在接口上启用 GLBP，并设置虚拟网关的主用 IP 地址。

	示例： `Device(config-if)# glbp 10 ip 10.21.8.10`	• 在设置了主用 IP 地址之后，用户可以使用再次使用命令 **glbp** *group* **ip**，同时加上 **secondary** 关键字来给这个组设置另一个 IP 地址；
步骤 6	**end** 示例： `Device(config-if)# end`	离开接口配置模式并返回特权 EXEC 模式
步骤 7	**show glbp** [*interface-type interface-number*] [*group*] [*state*] [**brief**] 示例： `Device(config)# show glbp GigabitEthernet 1/0/1 10`	（可选）显示一台设备上关于 GLBP 组的信息。 • 通过可选关键字 **brief** 可以用一行信息显示关于每个虚拟网关或虚拟转发设备的信息；

在下面的示例中，示例的输出信息显示了本地设备上关于这个编号为 10 的 GLBP 组的状态：

```
Device# show glbp GigabitEthernet 1/0/1 10
GigabitEthernet1/0/1 - Group 10
State is Active
1 state change, last state change 00:04:52
Virtual IP address is 10.21.8.10
Hello time 3 sec, hold time 10 sec
Next hello sent in 0.608 secs
Redirect time 600 sec, forwarder time-out 14400 sec
Preemption disabled
Active is local
Standby is unknown
Priority 100 (default)
Weighting 100 (default 100), thresholds: lower 1, upper 100
Load balancing: round-robin
Group members:
ac7e.8a35.6364 (10.21.8.32) local
There is 1 forwarder (1 active)
Forwarder 1
State is Active
1 state change, last state change 00:04:41
MAC address is 0007.b400.0a01 (default)
Owner ID is ac7e.8a35.6364
Redirection enabled
Preemption enabled, min delay 30 sec
Active is local, weighting 100
```

创建并自定义一个 VRRP 组

要创建一个 VRRP 组，需要执行下面的配置步骤。步骤 6 到步骤 14 所示为针对这个组所作

的自定义设备，这些步骤的配置都是可选的：

总步骤

1. enable

2. configure terminal

3. fhrp version vrrp v3

4. interface *type number*

5. vrrp *group-id* **address-family** {**ipv4** | **ipv6**}

6. address *ip-address* [**primary** | **secondary**]

7. description *group-description*

8. match-address

9. preempt delay minimum *seconds*

10. priority *priority-level*

11. timers advertise *interval*

12. vrrpv2

13. vrrs leader *vrrs-leader-name*

14. shutdown

15. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**fhrp version vrrp v3** 示例： `Device(config)# fhrp version vrrp v3`	启用配置 VRRPv3 和 VRRS 的功能。 **注释**：在使用 VRRPv3 时，VRRPv2 就是不可用的
步骤 4	**interface** *type number* 示例： `Device(config)# interface GigabitEthernet 0/0/0`	配置一个接口，并且进入接口的配置模式
步骤 5	**vrrp** *group-id* **address-family** {**ipv4**	**ipv6**} 示例： `Device(config-if)# vrrp 3 address-family`	创建 VRRP 组，并进入 VRRP 配置模式

	ipv4	
步骤 6	**address** *ip-address* [**primary** | **secondary**] 示例： Device(config-if-vrrp)# address 100.0.1.10 primary	给 VRRP 组设置一个主用地址或辅助地址。 **注释：**IPv6 VRRP 需要配置一个主用虚拟链路本地 IPv6 地址这个组才能正常工作。当用户给组配置了主用链路本地 IPv6 地址之后，还可以添加辅助的全局地址
步骤 7	**escription** *group-description* 示例： Device(config-if-vrrp)# description group 3	（可选）给 VRRP 组设置描述信息
步骤 8	**match-address** 示例： Device(config-if-vrrp)# match-address	（可选）将通告数据包中的辅助地址与配置的地址进行配置。 • 匹配辅助地址的操作是默认启用的
步骤 9	**preempt delay minimum** *seconds* 示例： Device(config-if-vrrp)# preempt delay minimum 30	（可选）启用低优先级主用设备抢占功能，并根据需要配置一个延迟时间。 • 抢占操作是默认启用的
步骤 10	**priority** *priority-level* 示例： Device(config-if-vrrp)# priority 3	（可选）给 VRRP 组设置优先级值。 • VRRP 组的优先级值默认为 100
步骤 11	**timers advertise** *interval* 示例： Device(config-if-vrrp)# timers advertise 1000	（可选）设置通告计时器，单位为毫秒。 • 通告时间间隔默认会设置为 1000 毫秒
步骤 12	**vrrpv2** 示例： Device(config-if-vrrp)# vrrpv2	（可选）同时启用对 VRRPv2 的支持，这项操作的目的是为了与只支持 VRRPv2 的设备进行互操作。 • VRRPv2 默认是禁用的
步骤 13	**vrrs leader** *vrrs-leader-name* 示例： Device(config-if-vrrp)#	（可选）设置在 VRRS 上注册的领导者（leader）名称，以供追随者（follower）使用。 • 注册的 VRRS 名默认是不可用的

	vrrs leader leader-1	
步骤 14	**shutdown** 示例： Device(config-if-vrrp)# shutdown	（可选）对 VRRP 组禁用 VRRP 配置。 • VRRP 组默认启用 VRRP 配置
步骤 15	**end** 示例： Device(config-if)# end	返回特权 EXEC 模式

配置 FHRP 客户端启动之前的延迟周期

要配置一个接口上的所有 FHRP 客户端启动前的延迟周期，需要执行下面的配置步骤：

总步骤

1. enable

2. configure terminal

3. fhrp version vrrp v3

4. interface *type number*

5. fhrp delay {[**minimum**] [**reload**] *seconds*}

6. end

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： Device> **enable**	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** 示例： Device# **configure terminal**	进入全局配置模式
步骤 3	**fhrp version vrrp v3** 示例： Device(config)# fhrp version vrrp v3	启用配置 VRRPv3 和 VRRS 的功能。 **注释**：在使用 VRRPv3 时，VRRPv2 就是不可用的
步骤 4	**interface** *type number* 示例： Device(config)# interface GigabitEthernet 0/0/0	进入接口的配置模式
步骤 5	**fhrp delay** {[**minimum**] [**reload**] *seconds*} 示例：	设置在一个接口启动之后，FHRP 客户端初始化的延迟周期。 • 取值范围是 0-3600 秒

	Device(config-if)# fhrp delay minimum 5	
步骤 6	**end** 示例： Device(config-if)# end	离开接口配置模式并返回特权 EXEC 模式

VRRPv3 协议支持的配置示例

示例：在一台设备上启用 VRRPv3

下面的示例显示了如何在一台设备上启用 VRRPv3：

```
Device> enable
Device# configure terminal
Device(config)# fhrp version vrrp v3
Device(config-if-vrrp)# end
```

示例：创建和自定义一个 VRRP 组

下面的示例显示了如何创建和自定义一个 VRRP 组：

```
Device> enable
Device# configure terminal
Device(config)# fhrp version vrrp v3
Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# vrrp 3 address-family ipv4
Device(config-if-vrrp)# address 100.0.1.10 primary
Device(config-if-vrrp)# description group 3
Device(config-if-vrrp)# match-address
Device(config-if-vrrp)# preempt delay minimum 30
Device(config-if-vrrp)# end
```

注释： 在上例中，我们是在全局配置模式下配置了命令 **fhrp version vrrp v3**。

示例：配置 FHRP 客户端启动之前的延迟周期

这个示例显示了如何配置 FHRP 客户端启动之前的延迟周期：

```
Device> enable
Device# configure terminal
Device(config)# fhrp version vrrp v3
Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# fhrp delay minimum 5
Device(config-if-vrrp)# end
```

注释： 在上例中，我们给接口开启之后，FHRP 客户端启动的延迟周期设置为了 5 秒。用户可以在这里设置的时间周期是从 0 到 3600 秒。

示例：VRRP 状态、配置与具体的统计数据

下面是关于一个 VRRP 组状态、配置和具体统计数据的输出信息：

```
Device> enable
Device# show vrrp detail
GigabitEthernet1/0/1 - Group 3 - Address-Family IPv4
Description is "group 3"
```

```
State is MASTER
State duration 53.901 secs
Virtual IP address is 100.0.1.10
Virtual MAC address is 0000.5E00.0103
Advertisement interval is 1000 msec
Preemption enabled, delay min 30 secs (0 msec remaining)
Priority is 100
Master Router is 10.21.0.1 (local), priority is 100
Master Advertisement interval is 1000 msec (expires in 832 msec)
Master Down interval is unknown
VRRPv3 Advertisements: sent 61 (errors 0) - rcvd 0
VRRPv2 Advertisements: sent 0 (errors 0) - rcvd 0
Group Discarded Packets: 0
VRRPv2 incompatibility: 0
IP Address Owner conflicts: 0
Invalid address count: 0
IP address configuration mismatch : 0
Invalid Advert Interval: 0
Adverts received in Init state: 0
Invalid group other reason: 0
Group State transition:
Init to master: 0
Init to backup: 1 (Last change Sun Mar 13 19:52:56.874)
Backup to master: 1 (Last change Sun Mar 13 19:53:00.484)
Master to backup: 0
Master to init: 0
Backup to init: 0
Device# exit
```

其他参考资料

相关文档

相关主题	**文档名**
Inspur INOS 命令	管理命令列表，适用于所有系统
FHRP 命令	第一跳冗余协议命令参考手册
配置 VRRPv2	*配置 VRRP*

标准与 RFC

标准/RFC	**标题**
RFC 5798	*虚拟路由器冗余协议*

技术助手

描述	**链接**
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档	http://www.icntnetworks.com

和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	

关于 VRRPv3 协议支持的特性信息

下表提供了关于这部分文档所描述的版本信息。这个表仅罗列了在一个版本系列中引入这个特性的具体软件版本。如无特别说明，这个软件版本系列的后续版本同样支持这个特性。用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

表 29：关于 VRRPv3 协议支持的特性信息

特性名	版本	特性信息
VRRPv3 协议支持	Inspur INOS XE 3.6E	虚拟路由器冗余协议（VRRP）可以让一组设备称为一个虚拟设备，以此来提供冗余。接下来，LAN 客户端可以将这台虚拟设备配置为它们的网关。虚拟设备代表的是一组设备，这组设备也称为 VRRP 组。 VRRP 第 3 版协议支持特性可以支持 IPv4 地址和 IPv6 地址。 在 Inspur INOS XE 3.6E 版中，下列平台可以支持这一特性： 这一版系统引入或修改了下列命令：**fhrp delay**、**show vrrp**、**vrrp address-family**

词汇表

- **虚拟 IP 地址所有者**：即拥有虚拟设备 IP 地址的那台 VRRP 设备。地址拥有者是指用虚拟设备地址作为其物理接口地址的设备。
- **虚拟设备**：组成组的一台或几台 VRRP 设备。虚拟设备会充当 LAN 客户端的默认网关设备。虚拟设备也称为 VRRP 组。
- **虚拟备份设备**：当虚拟主设备出现故障时，一台或几台可以接替数据包转发工作的 VRRP 设备。
- **虚拟主设备**：当前负责向虚拟设备 IP 地址转发数据包的 VRRP 设备。一般来说，虚拟主设备也会充当 IP 地址所有者。

- **VRRP 设备：**即运行 VRRP 的设备。

配置 GLBP

配置 GLBP

网关负载分担协议（GLBP）可以像热备份路由器协议（HSRP）和虚拟路由器冗余协议（VRRP）那样对数据包提供保护，让它们不受故障设备的影响，同时可以在一组冗余设备之间实现数据包的负载分担。

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

GLBP 的限制条件

EOT（Enhanced Object Tracking）不能感知状态化故障切换（SSO），也不能在 SSO 模式下与 GLBP 一起使用。

GLBP 的前提条件

在配置 GLBP 之前，要保证设备的物理接口上可以支持多 MAC 地址。用户每多配置一个 GLBP 转发设备，就要在原来的基础上多添加一个 MAC 地址。

关于 GLBP 的信息

GLBP 概述

GLBP 可以在一个 IEEE 802.3 局域网中，给只配置了一台默认网关设备的 IP 主机提供自动设备备份。局域网中的多个首跳设备会共同提供一个虚拟第一跳 IP 设备，这些设备会分担 IP 数据包的转发功能。局域网中的其他设备会则会充当冗余 GLBP 设备，在当前转发设备发生故障时成为主用设备。

GLBP 会给用户提供与 HSRP 和 VRRP 比较类似的功能。HSRP 和 VRRP 可以让多台设备参与到一个用户配置了虚拟 IP 地址的虚拟设备组中。其中一台成员设备会被选举为主用设备，负责转发那些发送给组虚拟 IP 地址的数据包。组中的其他设备为冗余设备，直至主用设备发生故障为止。这些备份设备会拥有一些协议没有使用的带宽。虽然用户可以将同一组设备配置为多个虚拟设备组，但主机上必须配置不同的默认网关，所以这就会增加管理负担。GLBP 的优点在于，它还可以使用一个虚拟 IP 地址和多个虚拟 MAC 地址，来通过多台设备（网关）实现负载分担。GLBP 组中的所有设备会分担转发的负担，而不是由其中的一台设备来处理转发流量，其他设备都处于空闲状态。每台主机上都要配置相同的虚拟 IP 地址，而虚拟设备组中的所有设备都要参与数据包的转发。GLBP 成员设备之间会通过每 3 秒发送一次的 hello 消息相互进行通信，这个消息的目的是组播地址 224.0.0.102，目的端口是 3222（源端口和目的端口）。

GLBP 数据包的类型

GLBP 会使用 3 种不同的数据包来完成操作。这些数据包类型是 Hello 消息，请求消息和响应消息。Hello 数据包的目的是通告协议信息。Hello 数据包是组播数据包，所有处于 Speak、Standby 和 Active 状态下的虚拟网关或虚拟转发设备都会发送 Hello 数据包。请求消息和响应消息的作用是分配虚拟 MAC 地址。这两类消息都是单播数据包，以主用虚拟网关（AVG）作为消息的源和目的。

GLBP 主用虚拟网关

GLBP 组的成员会将一台网关选为组的主用虚拟网关（AVG），其他组成员则会在 AVG 不可用时给 AVG 提供备份。AVG 会给 GLBP 组中的每个成员分配一个虚拟的 MAC 地址。每台网关都会认为自己应该负责给那些发往（AVG 分配给自己的）虚拟 MAC 地址的数据包提供转发。这些网关称为其虚拟 MAC 地址的主用虚拟转发设备（AVF）。

AVG 也负责响应地址解析协议发送给虚拟 IP 地址的请求。负载分担是通过 AVG 用不同虚拟 MAC 地址响应 ARP 请求的方式实现的。

当用户配置了命令 **no glbp load-balancing** 时，如果 AVG 没有 AVF，它会优先用第一个侦听虚拟转发设备（VF）的 MAC 地址来响应 ARP 请求，这会让流量通过另一个网关进行路由，直到 VF 迁移回当前的 AVG。

在下图中，路由器 A（或设备 A）是 GLBP 组的 AVG，它负责响应去往虚拟 IP 地址 10.21.8.10 的流量。路由器 A 也是虚拟 MAC 地址 0007.b400.0101 的 AVF。路由器 B（或设备 B）是同一个 GLBP 组的成员，它是虚拟 MAC 地址 0007.b400.0102 的 AVF。客户端 1 的默认网关 IP 地址为 10.21.8.10，且其网关 MAC 地址为 0007.b400.0101。客户端 2 共享了同一个默认网关 IP 地址，但会接收发送给网关 MAC 地址 0007.b400.0102 的流量，因为路由器 B 会为路由器 A 分担负载。

图 19：GLBP 拓扑

WAN Link1	WAN 链路 1
WAN Link2	WAN 链路 2

Router A	路由器 A
Router B	路由器 B
Virtual IP address 10.21.8.10	虚拟 IP 地址 10.21.8.10
Virtual MAC address 0007.b400.0101	虚拟 MAC 地址 0007.b400.0101
Virtual MAC address 0007.b400.0102	虚拟 MAC 地址 0007.b400.0102
active virtual gateway	主用虚拟网关
active virtual forwarder	主用虚拟转发设备
Client 1	客户端 1
Client 2	客户端 2
Default gateway: Virtual IP address 10.21.8.10	默认网关：虚拟 IP 地址 10.21.8.10
Gateway MAC: Virtual MAC address 0007.b400.0101	网关 MAC：虚拟 MAC 地址 0007.b400.0101
Virtual IP address 10.21.8.10	虚拟 IP 地址 10.21.8.10
Virtual MAC address 0007.b400.0102	虚拟 MAC 地址 0007.b400.0102

如果路由器 A 不可用，客户端 1 并不会失去与 WAN 的连接，因为路由器 B 会继续负责转发那些发送给路由器 A 虚拟 MAC 地址的数据包，并响应那些发送给路由器 B 自己虚拟 MAC 的数据包。路由器 B 也会在整个 GLBP 组中承担 AVG 的角色。当 GLBP 组中一台设备出现故障时，去往 GLBP 成员的通信并不会受到影响。

GLBP 虚拟 MAC 地址分配

GLBP 组中最多支持 4 个虚拟 MAC 地址。AVG 负责向组中的每个成员分配虚拟 MAC 地址。其他的组成员会在通过 hello 消息发现了 AVG 之后请求虚拟 MAC 地址。网关会按顺序分配到下一个 MAC 地址。AVG 分配了虚拟 MAC 地址的虚拟转发设备称为主虚拟转发设备（primary virtual forwarder）。GLBP 组的其他成员会通过 hello 消息学习到这个虚拟 MAC 地址。学习到这个虚拟 MAC 地址的虚拟转发设备称为次虚拟转发设备（secondary virtual forwarder）。

GLBP 虚拟网关冗余

GLBP 操作虚拟网关冗余的方式与 HSRP 相同。其中一台网关会被选举为 AVG，另一台网关则会被选举为备用虚拟网关，剩下的网关则会进入一种侦听状态。

如果 AVG 出现了故障，备用的虚拟网关就会开始处理与虚拟 IP 地址有关的流量。接下来，GLBP 会从侦听状态中选举出一台新的备用虚拟网关。

GLBP 虚拟转发设备冗余

虚拟转发设备冗余与虚拟网关冗余类似。如果 AVF 出现故障，一台处于侦听状态的备用虚拟转发设备就会开始处理与虚拟 IP 地址有关的流量。

新的 AVF 是一个不同转发设备编号的主用虚拟转发设备。GLBP 会将主机迁移到一个与老转发设备不同的编号，此时 GLBP 会使用两个计时器，一旦网关变更为主用虚拟转发设备的状态，这两个计时器就会启动。GLBP 会使用 hello 消息来交互当前计时器状态的信息。

重定向时间是指 AVG 继续将主机重定向到老的虚拟转发设备 MAC 地址的间隔时间。当重定向时间过期时，AVG 就不会继续在 ARP 响应消息中使用老的虚拟转发设备 MAC 地址了，不过虚拟转发设备还是会继续转发那些发送给老虚拟转发设备 MAC 地址的数据包。

备用抑制时间（secondary holdtime）是指虚拟转发设备继续保持有效的间隔时间。当备用抑制时间过期时，这台虚拟转发设备就会就会从 GLBP 中的所有网关上被移除。超时的虚拟转发设备编号也可以由 AVG 重新进行分配。

GLBP 网关优先级

GLBP 网关优先级的作用是用来判断各个 GLBP 扮演的角色，以及当 AVG 失效时应该执行的

操作。
优先级也会用来决定一台 GLBP 设备是否应该充当备用虚拟网关，以及当前 AVG 失效时，其他设备继任 AVG 的顺序。用户可以使用命令 **glbp priority** 来给每个备用虚拟网关配置一个从 1 到 255 之间的优先级值。
在“GLBP 拓扑”一图中，如果路由器 A（或设备 A），也就是拓扑中的 AVG 发生故障时，GLBP 就会通过选举来判断哪台备用虚拟网关设备应该接替它的角色。在这个示例中，路由器 B（或设备 B）是组中唯一其他的成员，所以它会自动成为信的 AVG。如果同一个 GLBP 组中还有另一个优先级更高的设备，那么这台优先级更高的设备就会被选中。如果两台设备的优先级相同，那么 IP 地址更高的备用虚拟网关就会被选中，称为主用虚拟网关。
在默认情况下，GLBP 虚拟网关的抢占功能是关闭的。只有在当前的 AVG 出现故障时，备用虚拟网关才会成为 AVG，无论这些虚拟网关的优先级有多高。用户可以使用命令 **glbp preempt** 来启用 GLBP 虚拟网关抢占功能。如果启用抢占功能，那么如果备用虚拟网关的优先级高于当前的 AVG，备用虚拟网关就可以成为 AVG。

GLBP 网关权重与追踪

GLBP 使用了权重机制，来定义 GLBP 组中每台设备的转发能力。分配给 GLBP 组中每台设备的权重，可以用来判断它是否会参与数据包转发，如果参与转发，那么它会为局域网中多少比例的主机转发数据包。用户可以通过设置门限值，让一个 GLBP 组的权重值低于某个参数时，转发数据包的操作就会被禁用，而当权重值高于门限值时，转发自动重新启用。
GLBP 组的权重值可以通过追踪设备中的接口状态来自动调节。如果被追踪的接口状态宕机，GLBP 组的权重值就会降低用户所设置的数值。用户可以对不同的接口设置追踪，让它们在出现故障时，GLBP 权重值降低不同的数值。
在默认情况下，GLBP 虚拟转发设备抢占特性是启用的，延迟时间为 30 秒。如果当前 AVF 权值低于低权值门限值的时间超过 30 秒，备用虚拟转发设备就会成为 AVF。用户可以使用命令 **no glbp forwarder preempt** 来禁用 GLBP 转发设备的抢占功能，也可以使用命令 **glbp forwarder preempt delay minimum** 来修改延迟时间。

GLBP MD5 认证

GLBP MD5 认证会使用行业标准的 MD5 算法来提供增强的可靠性和安全性。MD5 认证可以比明文认证这种方式提供更加强大的安全性，防止欺骗软件带来的风险。
MD5 认证可以让每个 GLBP 组成员使用密钥来创建一个加密的 MD5 散列值，并且将这个散列值插入到出站数据包当中如果入站数据包中的散列值与设备生成的散列值不同，那么这个数据包就会被忽略。
MD5 散列值的参数既可以直接以密钥串的方式添加在配置当中，或者可以通过密钥链来间接提供。密钥串的长度不能超过 100 个字符。
在一个 GLBP 组中，如果两台设备的认证配置不同，那么接收方就会忽略自己接收到的 GLBP 数据包。GLBP 的认证方式包括：

- 不认证
- 明文认证
- MD5 认证

在下列情况下，GLBP 数据会被拒绝：

- 设备采用的认证方式与入站数据包的认证方式不同；
- 设备的 MD5 摘要值与入站数据包的摘要值不同；
- 设备上配置的文字认证字符串与入站数据包的文字认证字符串不同。

ISSU-GLBP

GLBP 支持 ISSU（服务期间软件升级）。ISSU 可以让高可用性（HA）系统运行在状态化故障切

换（SSO）模式下，即使主用和备用路由器处理器（RP）或线卡上运行的 Inspur INOS 软件版本不同。

ISSU 可以让设备一边继续转发数据包并维护当前的会话，一边从一个设备支持的 Inspur INOS 版本升级或降级到另一个版本，这样可以减少设备中断的时间。ISSU 之所以可以实现这种升降级功能，是因为 ISSU 可以让主用 RP 和备用 RP 在短时间运行不同的软件版本来维护 RP 间的状态化信息。这个特性可以让系统切换到运行升级后（或降级后）软件版本的备用 RP 上并继续转发数据包，这个过程中会话不会丢弃，丢包的数量可以降至最低。这项特性默认是启用的。

GLBP SSO

通过引入 GLBP SSO 功能，GLBP 可以感知到状态化故障切换（SSO）。GLBP 可以检测到一台设备何时因故障切换到了备用路由处理器（RP），并保存在其当前的组状态中。

SSO 通过支持双 RP 的网络设备（一般是边缘设备）发挥其功能，它可以通过让一台 RP 充当主用处理器，另一台 RP 充当备用处理器的方式，提供 RP 冗余。SSO 也可以在 RP 之间同步重要的状态信息，让网络设备可以动态在 RP 之间维护。

如果无法感知 SSO，那么如果在一台包含冗余 RP 的设备上部署了 GLBP，那么主用 RP 与备用 RP 的角色切换会导致设备放弃自己的 GLBP 组成员身份，然后重新再加入到组中，就像这台设备重新启动了一样。GLBP SSO 特性可以让 GLBP 在进行故障切换的过程中，已然维持自己的组成员身份。由于设备有能力维护冗余 RP 之间的 GLBP 状态信息，因此备用 RP 在故障切换和故障切换之后，可以继续 GLBP 中的设备活动。

这项特性默认是启用的。要想禁用这项特性，需要在全局配置模式中输入命令 **no glbp**。

GLBP 的优势

负载分担

用户可以通过配置 GLBP，让来自局域网客户端的流量由多台设备分担，通过这种方式在可用之间更加平均地分担流量负载。

多虚拟设备

GLBP 支持最多在一台设备物理接口上配置 1024 个虚拟设备（GLBP 组），每个组可以添加最多 4 个虚拟转发设备。

抢占

GLBP 的冗余功能让可以让一台拥有高优先级、刚刚成为可用设备的备份虚拟设备，抢占主用虚拟网关（AVG）的角色。转发设备抢占功能的工作方式也是类似的，但转发设备抢占功能会使用权值而不是优先级，而且转发设备抢占功能默认是启用的。

认证

GLBP 支持行业标准的消息摘要 5（MD5）算法，可以提供更加强大的可靠性与安全性，也可以保护网络免受 GLBP 欺骗软件的攻击。GLBP 组中的设备如果配置了与其他成员不同的认证字符串，那么其他成员就会忽略它发送的消息。用户可以选择在 GLBP 组成员之间选择明文密钥认证，以检测配置错误。

如何配置 GLBP

自定义 GLBP

自定义 GLBP 的操作是可选的。用户应该明白，一旦启用 GLBP 组，这个组就会开始工作。如果用户先启用了一个 GLBP 组然后再自定义 GLBP，设备有可能会在用户完成特性自定义之前就成为 AVG，并接开始对这个组进行控制。因此，如果用户打算自定义 GLBP，最好先自定义，再启用 GLBP。

总步骤

1. enable

2. configure terminal

3. interface *type number*

4. ip address *ip-address mask* [**secondary**]

5. glbp *group* **timers** [**msec**] *hellotime* [**msec**] *holdtime*

6. glbp *group* **timers redirect** *redirect timeout*

7. glbp *group* **load-balancing** [**host-dependent** | **round-robin** | **weighted**]

8. glbp *group* **priority** *level*

9. glbp *group* **preempt** [**delay minimum** *seconds*]

10. glbp *group* **client-cache maximum** *number* [**timeout** *minutes*]

11. glbp *group* **name** *redundancy-name*

12. exit

13. no glbp sso

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *type number* 示例： `Device(config)# interface GigabitEthernet 1/0/1`	设置接口的类型和编号，并且进入接口的配置模式
步骤 4	**ip address** *ip-address mask* [**secondary**] 示例： `Device(config-if)# ip address 10.21.8.32 255.255.255.0`	给接口设置主用 IP 地址或辅助 IP 地址
步骤 5	**glbp** *group* **timers** [**msec**] *hellotime* [**msec**] *holdtime* 示例： `Device(config-if)# glbp 10 timers 5 18`	配置一个 GLBP 组中，AVG 发送前后两个 hello 消息之间的时间间隔： • *holdtime* 参数的用途是以秒为单位设置时间间隔，在这个时间超时时间，AVG 就会认为 hello 消息中的虚拟网关和虚拟转发设备是有效的； • 可选关键字 **mesc** 的作用是设置下列参数的设

		置为毫秒，而不是默认的单位秒
步骤 6	**glbp** *group* **timers redirect** *redirect timeout* 示例： `Device(config-if)# glbp 10 timers redirect 1800 28800`	配置 AVG 继续将客户端重定向给 AVF 的时间间隔。默认时间为 600 秒（即 10 分钟）， • *timeout* 参数的用途是以秒为单位，设置备用虚拟转发设备变为不可用之前需要经历的时间间隔。默认为 14400 秒（即 4 小时）。 **注释**：*redirect* 值是可以取 0 的，因为 Inspur INOS 软件的预配置已经使用了 0 这个值，而且也给升级阶段带来了负面影响。不过，我们不推荐将这个参数设置为 0，如果设置为 0，重定向计时器就永远不会超时。如果重定向计时器永不超时，而设备又发生了故障，那么新主机所分配到的，仍然会是那台已经发生了故障的设备，而不会被重定向给备用设备
步骤 7	**glbp** *group* **load-balancing** [**host-dependent** | **round-robin** | **weighted**] 示例： `Device(config-if)# glbp 10 load-balancing host-dependent`	设置 GLBP AVG 采用的负载分担方式
步骤 8	**glbp** *group* **priority** *level* 示例： `Device(config-if)# glbp 10 priority 254`	设置 GLBP 组中网关的优先级别. • 默认值为 100
步骤 9	**glbp** *group* **preempt** [**delay minimum** *seconds*] 示例： `Device(config-if)# glbp 10 preempt delay minimum 60`	让设备在比当前 AVG 拥有更高优先级时，即接替它成为 GLBP 组中的 AVG • 这条命令默认是禁用的； • 用户可以使用可选的关键字 **delay** 和 **minimum** 和 *seconds* 参数来设置抢占 AVG 之前的最小延迟时间间隔，单位为秒
步骤 10	**glbp** *group* **client-cache maximum** *number* [**timeout** *minutes*] 示例： `Device(config-if)# glbp 10 client-cache maximum 1200 timeout 245`	（可选）启用 GLBP 客户端缓存 • 这条命令默认是禁用的； • *number* 参数的用途是设置缓存会给这个 GLBP 组保存的客户端最大数量。取值范围是从 8 到 2000； • 用户可以使用可选的关键字与参数组合 **timeout** *minutes* 来配置自客户端信息最后一次更新之后，这个客户端条目可以停留在 GLBP 客户端缓存中的最大时长。取值范围是从 1 到 1440 分钟（即 1 天）。

		注释：对于 IPv4 网络来说，Inspur 推荐将 GLBP 客户端缓存超时值设置得比最大期待的终端主机 ARP（地址解析协议）缓存超时时间略长。
步骤 11	**glbp** *group* **name** *redundancy-name* 示例： `Device(config-if)# glbp 10 name abc123`	给 GLBP 组分配一个名称，以启用 IP 冗余。 • GLBP 冗余客户端必须配置与 GLBP 组相同的名称，这样冗余客户端和 GLBP 组才能够建立关联
步骤 12	**exit** 示例： `Device(config-if)# exit`	离开接口配置模式，让设备返回全局配置模式
步骤 13	**no glbp sso** 示例： `Device(config)# no glbp sso`	（可选）禁用 GLBP 对 SSO 的支持

使用密钥串来配置 GLBP MD5 认证

总步骤

1. enable

2. configure terminal

3. interface *type number*

4. ip address *ip-address mask* [**secondary**]

5. glbp *group-number* **authentication md5 key-string** [**0** | **7**] *key*

6. glbp *group-number* **ip** [*ip-address* [**secondary**]]

7. Repeat Steps 1 through 6 on each device that will communicate.

8. end

9. show glbp

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *type number* 示例： `Device(config)# interface`	设置接口的类型和编号，并且进入接口的配置模式

	GigabitEthernet 1/0/1	
步骤 4	**ip address** *ip-address mask* [**secondary**] 示例： Device(config-if)# ip address 10.21.0.1 255.255.255.0	给接口设置主用 IP 地址或辅助 IP 地址
步骤 5	**glbp** *group-number* **authentication md5 key-string** [**0** | **7**] *key* 示例： Device(config-if)# glbp 1 authentication md5 key-string d00b4r987654321a	配置 GLBP MD5 认证的认证密钥。 • 密钥串的长度不能超过 100 个字符； • *key* 这个参数之前没有前缀，但如果设置 0 表示这个密钥不进行加密； • 设置 7 表示要对密钥进行加密。如果用户配置了全局配置命令 **service password-encryption**，密钥串认证密钥就会自动加密。
步骤 6	**glbp** *group-number* **ip** [*ip-address* [**secondary**]] 示例： Device(config-if)# glbp 1 ip 10.0.0.10	在接口上启用 GLBP，并且设置虚拟网关的主用 IP 地址
步骤 7	在每台要相互通信的设备上重复第 1 步到第 6 步的配置	——
步骤 8	**end** 示例： Device(config-if)# end	返回特权 EXEC 模式
步骤 9	**show glbp** 示例： Device# show glbp	（可选）显示 GLBP 信息。 用户可以使用这条命令来验证自己所作的配置。如果配置了密钥字符串和认证类型的话，这些信息也会通过这条命令显示出来

使用密钥链配置 GLBP MD5 认证

要使用密钥链配置 GLBP MD5 认证，需要执行下面的配置步骤。密钥链可以根据用户的配置，让不同的密钥使用不同的次数。GLBP 会查询相应的密钥链来从指定密钥链中获取当前正在使用的密钥和密钥 ID。

总步骤

1. enable

2. configure terminal

3. key chain *name-of-chain*

4. key *key-id*

5. key-string *string*

6. exit

7. exit

8. interface *type number*

9. ip address *ip-address mask* [**secondary**]

10. glbp *group-number* **authentication md5 key-chain** *name-of-chain*

11. glbp *group-number* **ip** [*ip-address* [**secondary**]]

12. Repeat Steps 1 through 10 on each device that will communicate.

13. end

14. show glbp

15. show key chain

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**key chain** *name-of-chain* 示例： `Device(config)# key chain glbp2`	针对路由协议启用认证，标识一组认证密钥，并进入密钥链配置模式
步骤 4	**key** *key-id* 示例： `Device(config-keychain)# key 100`	在一个密钥链中用数字标识一个密钥串。 • *key-id* 值必须设置为一个数字
步骤 5	**key-string** *string* 示例： `Device(config-keychain-key)# key-string abc123`	给密钥设置认证字符串，并进入密钥链密钥认证模式。 • *string* 这个值的参数可以是 1 到 80 之间的参数，也可以是大小写字母；但第 1 个字符不能是数字
步骤 6	**exit** 示例： `Device(config-keychain-key)# exit`	返回密钥链配置模式
步骤 7	**exit** 示例： `Device(config-keychain)#`	返回全局配置模式

	exit	
步骤 8	**interface** *type number* 示例： Device(config)# interface GigabitEthernet 1/0/1	设置接口的类型和编号，并且进入接口的配置模式
步骤 9	**ip address** *ip-address mask* [**secondary**] 示例： Device(config-if)# ip address 10.21.0.1 255.255.255.0	给接口设置主用 IP 地址或辅助 IP 地址
步骤 10	**glbp** *group-number* **authentication md5 key-chain** *name-of-chain* 示例： Device(config-if)# glbp 1 authentication md5 key-chain glbp2	配置 GLBP MD5 认证的认证 MD5 密钥链。 • 密钥链的名称必须与步骤 3 中设置的名称相同
步骤 11	**glbp** *group-number* **ip** [*ip-address* [**secondary**]] 示例： Device(config-if)# glbp 1 ip 10.21.0.12	在接口上启用 GLBP，并且设置虚拟网关的主用 IP 地址
步骤 12	在每台要相互通信的设备上重复第 1 步到第 10 步的配置	——
步骤 13	**end** 示例： Device(config-if)# end	返回特权 EXEC 模式
步骤 14	**show glbp** 示例： Device# show glbp	（可选）显示 GLBP 信息。 用户可以使用这条命令来验证自己所作的配置。如果配置了密钥链和认证类型的话，这些信息也会通过这条命令显示出来
步骤 15	**show key chain** 示例： Device# show key chain	（可选）显示认证密钥的信息

配置 GLBP 文本认证

文本认证提供的安全性不高。如果对安全性有要求，应使用 MD5 认证。

总步骤

1. enable

2. configure terminal

3. interface *type number*

4. ip address *ip-address mask* [**secondary**]

5. glbp *group-number* **authentication text** *string*

6. glbp *group-number* **ip** [*ip-address* [**secondary**]]

7. Repeat Steps 1 through 6 on each device that will communicate.

8. end

9. show glbp

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *type number* 示例： `Device(config)# interface GigabitEthernet 1/0/1`	设置接口的类型和编号，并且进入接口的配置模式
步骤 4	**ip address** *ip-address mask* [**secondary**] 示例： `Device(config-if)# ip address 10.21.0.1 255.255.255.0`	给接口设置主用 IP 地址或辅助 IP 地址
步骤 5	**glbp** *group-number* **authentication text** *string* 示例： `Device(config-if)# glbp 10 authentication text stringxyz`	对从组中其他设备那里接收到的 GLBP 数据包进行认证。 • 如果配置了认证，那么 GLBP 组中的所有设备都必须使用相同的认证串
步骤 6	**glbp** *group-number* **ip** [*ip-address* [**secondary**]]	在接口上启用 GLBP，并且设置虚拟网关的主用 IP 地址

	示例： `Device(config-if)# glbp 1 ip 10.0.0.10`	
步骤 7	在每台要相互通信的设备上重复第 1 步到第 6 步的配置	——
步骤 8	**end** 示例： `Device(config-if)# end`	返回特权 EXEC 模式
步骤 9	**show glbp** 示例： `Device# show glbp`	（可选）显示 GLBP 信息。 用户可以使用这条命令来验证自己所作的配置。如果配置了密钥字符串和认证类型的话，这些信息也会通过这条命令显示出来

配置 GLBP 权重值与对象追踪

GLBP 权重的作用是判断一个 GLBP 组是否可以充当虚拟转发设备。用户可以对初始的权重值进行设置，同时根据需要指定门限值。用户可以追踪接口状态，并且设置当该接口宕机时权重值的减少量。当 GLBP 组权重低于某个值时，组就不再充当主用的虚拟转发设备。当权重高于指定值时，组可以恢复主用虚拟转发设备的角色。

总步骤

1. enable

2. configure terminal

3. track *object-number* **interface** *type number* {**line-protocol** |{**ip** | **ipv6**} **routing**}

4. exit

5. interface *type number*

6. glbp *group* **weighting** *maximum* [**lower** *lower*] [**upper** *upper*]

7. glbp *group* **weighting track** *object-number* [**decrement** *value*]

8. glbp *group* **forwarder preempt** [**delay minimum** *seconds*]

9. exit

10. show track [*object-number* | **brief**] [**interface** [**brief**] | **ip route** [**brief**] | **resolution** | **timers**]

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**track** *object-number* **interface** *type number* {**line-protocol** |{**ip**	**ipv6**}	配置要追踪的接口（这个接口的状态变化会影响 GLBP 网关的权重），并进入追踪配置模式。 • 这条命令可以对接口进行配置，并配置命令

	routing} 示例： `Device(config)# track 2 interface GigabitEthernet 1/0/1 ip routing`	**glbp weighting track** 中使用的对象值； • 关键字 **line-protocol** 的作用是追踪该接口的状态是否为 up。关键字 ip routing 也会校验该接口上是否启用了 IP 路由，以及是否配置了 IP 地址
步骤 4	**exit** 示例： `Device(config-track)# exit`	返回全局配置模式
步骤 5	**interface** *type number* 示例： `Device(config)# interface GigabitEthernet 1/0/1`	进入接口的配置模式
步骤 6	**glbp** *group* **weighting** *maximum* [**lower** *lower*] [**upper** *upper*] 示例： `Device(config-if)# glbp 10 weighting 110 lower 95 upper 105`	给 GLBP 网关设置初始权重值及上下门限值。
步骤 7	**glbp** *group* **weighting track** *object-number* [**decrement** *value*] 示例： `Device(config-if)# glbp 10 weighting track 2 decrement 5`	设置要追踪的对象，该对象会影响 GLBP 网关的权重值。 • *value* 这个参数的作用是指定当被追踪对象出现故障时，GLBP 网关权重值减少的数量
步骤 8	**glbp** *group* **forwarder preempt** [**delay minimum** *seconds*] 示例： `Device(config-if)# glbp 10 forwarder preempt delay minimum 60`	让 GLBP 组当前的 AVF 权重值降低到权重值门限之下时，本地设备开始在 GLBP 组中承担 AVF 的角色。 • 这条命令默认是启用的，延迟时间为 30 秒； • 用户可以使用可选的关键字 **delay** 和 **minimum** 和 *seconds* 参数来设置抢占 AVG 之前的最小延迟时间间隔，单位为秒
步骤 9	**exit**	返回特权 EXEC 模式

	示例： `Device(config-if)# exit`	
步骤 10	**show track** [*object-number* | **brief**] [**interface** [**brief**] | **ip route** [**brief**] | **resolution** | **timers**] 示例： `Device# show track 2`	显示追踪信息

GLBP 的排错

GLBP 引入了 5 条特权 EXEC 模式的命令，这些命令的作用是显示针对各类与 GLBP 操作相关事件的分析信息。命令 **debug condition glbp**、**debug glbp errors**、**debug glbp events**、**debug glbp packets** 和 **debug glbp terse** 都只能用来进行排错的，因为这些命令会让系统生成大量的信息，造成设备性能严重下降。要想减小命令 **debug glbp** 对设备性能造成的影响，用户可以执行下面介绍的操作。

下面的流程可以减小设备因使用命令 **debug condition glbp**or **debug glbp** 而给设备性能造成的影响，因为 console 端口不会再逐字符地造成处理器中断（processor interrupts）。如果用户不能直接连接到 console 端口，那也可以通过一台终端服务器来运行这个流程。但如果用户必须中断 Telnet 连接，那么用户可能会无法再次建立连接，这是因为设备有可能会因为处理器创建了大量调试输出信息，而无法对用户连接作出响应。

在开始前

要执行下面的步骤，要求用户能够直接通过 console 端口来连接运行 GLBP 的设备。

总步骤

1. enable

2. configure terminal

3. no logging console

4. Use Telnet to access a device port and repeat Steps 1 and 2.

5. end

6. terminal monitor

7. debug condition glbp *interface-type interface-number group* [*forwarder*]

8. terminal no monitor

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**no logging console** 示例：	禁用所有发送给 console 终端的日志消息。 • 要想重新启用发送给 console 的日志消息，需要在全局配置模式下输入命令 **logging console**

	Device(config)# no logging console	
步骤 4	用Telnet访问设备端口，并重复步骤1和步骤2	在一条递归 Telnet 会话中进入全局配置模式，将输出信息重定向到 console 端口之外
步骤 5	**end** 示例： Device(config)# end	回到特权 EXEC 模式
步骤 6	**terminal monitor** 示例： Device# terminal monitor	在虚拟终端上启用日志消息输出
步骤 7	**debug condition glbp** *interface-type interface-number group* [*forwarder*] 示例： Device# debug condition glbp GigabitEthernet 0/0/0 1	禁用关于 GLBP 条件的调试消息。 • 尝试只输入 **debug condition glbp** 或 **debug glbp** 命令，让输出信息仅限于某一项信息，并尽可能减小给处理器造成的负载。用户可以通过使用合理的参数和关键字来让设备显示关于某项信息的具体的调试信息； • 完成后输入 **no debug condition** 或 **no debug glbp**
步骤 8	**terminal no monitor** 示例： Device# terminal no monitor	在虚拟终端上禁用日志记录

GLBP 的配置示例

示例：自定义 GLBP 的配置

```
Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# ip address 10.21.8.32 255.255.255.0
Device(config-if)# glbp 10 timers 5 18
Device(config-if)# glbp 10 timers redirect 1800 28800
Device(config-if)# glbp 10 load-balancing host-dependent
Device(config-if)# glbp 10 priority 254
Device(config-if)# glbp 10 preempt delay minimum 60
Device(config-if)# glbp 10 client-cache maximum 1200 timeout 245
```

示例：使用密钥串来配置 GLBP MD5 认证

下面的示例显示了如何使用密钥串来配置 GLBP MD5 认证：

```
Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# ip address 10.0.0.1 255.255.255.0
Device(config-if)# glbp 2 authentication md5 key-string ThisStringIsTheSecretKey
Device(config-if)# glbp 2 ip 10.0.0.10
```

示例：使用密钥链来配置 GLBP MD5 认证

在下面的示例中，GLBP 查询了名为 AuthenticateGLBP 的密钥链，以获取这个密钥链当前正在使用的密钥和密钥 ID。

```
Device(config)# key chain AuthenticateGLBP
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string ThisIsASecretKey
Device(config-keychain-key)# exit
Device(config-keychain)# exit
Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# ip address 10.0.0.1 255.255.255.0
Device(config-if)# glbp 2 authentication md5 key-chain AuthenticateGLBP
Device(config-if)# glbp 2 ip 10.0.0.10
```

示例：配置 GLBP 文本认证

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.21.8.32 255.255.255.0
Device(config-if)# glbp 10 authentication text stringxyz
Device(config-if)# glbp 10 ip 10.21.8.10
```

示例：配置 GLBP 权重值

在下面的示例中，用户对设备进行了配置，让它追踪 POS 接口 5/0/0 和 6/0/0 的 IP 路由状态，同时用户也设置了初始 GLBP 权重值的高低门限值，同时将权重值的减量设置为了 10。如果 POS 接口 5/0/0 和 6/0/0 宕机，设备的权重值就会降低。

```
Device(config)# track 1 interface GigabitEthernet 1/0/1 line-protocol
Device(config)# track 2 interface GigabitEthernet 1/0/3 line-protocol
Device(config)# interface TenGigabitEthernet 0/0/1
Device(config-if)# ip address 10.21.8.32 255.255.255.0
Device(config-if)# glbp 10 weighting 110 lower 95 upper 105
Device(config-if)# glbp 10 weighting track 1 decrement 10
Device(config-if)# glbp 10 weighting track 2 decrement 10
```

示例：启用 GLBP 配置

在下面的示例中，用户在设备上启用了 GLBP，并且给 GLBP 组 10 设置了虚拟 IP 地址 10.21.8.10：

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.21.8.32 255.255.255.0
Device(config-if)# glbp 10 ip 10.21.8.10
```

GLBP 的其他参考资料

相关文档

相关主题	文档名
GLBP 命令：完整的命令语法、命令模式、命令历史、默认值、使用知道方针与示例	Inspur INOS IP 应用服务命令参考手册
服务期间软件升级（ISSU）配置	《Inspur INOS 高可用性配置指南》中的服务期间软件升级部分
密钥链与密钥管理命令：完整的命令语法、	《Inspur INOS 独立于 IP 路由的命令参考手

命令模式、命令历史、默认值、使用知道方针与示例	册》
对象追踪	“配置增强型对象追踪”部分
状态化故障切换	《Inspur INOS 高可用性配置指南》中的状态化故障切换部分
VRRP	“配置 VRRP”部分
HSRP	“配置 HSRP”部分
GLBP 对 IPv6 的支持	“FHRP-GLBP 对 IPv6 的支持”部分

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

关于 GLBP 的特性信息

下表提供了关于这部分文档所描述的版本信息。这个表仅罗列了在一个版本系列中引入这个特性的具体软件版本。如无特别说明，这个软件版本系列的后续版本同样支持这个特性。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

表 30：GLBP 的特性信息

特性名	版本	特性信息
网关负载分担协议		GLBP 可以保护数据流量，使其免受路由器或链路故障的影响。它的工作方式与 HSRP 和 VRRP 类似，而且可以让数据包在一组冗余路由器之间实现负载分担。 这一版系统引入或修改了下列命令：**glbp forwarder preempt**、**glbp ip**、**glbp load-balancing**、**glbp name**、**glbp preempt**、**glbp priority**、**glbp sso**、**glbp timers**、**glbp timers redirect**、**glbp weighting**、**glbp weighting track**、**show glbp**

GLBP MD5 认证	Inspur INOS XE 3.6E	MD5 认证可以提供比明文认证更加强大的安全性。MD5 认证可以让每个 GLBP 组成员用密钥来创建一个加密的 MD5 散列值，将这个散列值添加到出站数据包当中。设备会比较入站数据包的加密散列值，如果与设备生成的散列值不匹配，这个数据包就会被设备忽略。 这一版系统引入或修改了下列命令：**glbp authentication、 show glbp**
ISSU-GLBP		GLBP 支持服务期间软件升级（ISSU）特性。ISSU 可以让一个高可用性（HA）系统运行在状态化故障切换（SSO）模式下，即使主用、备用路由处理器（RP）或线卡运行的 Inspur INOS 软件版本不同。 这个特性可以让客户在系统升级期间，仍然享有系统在正常工作时的高可用性水准。也就是说，系统可以切换到备用 RP 并且继续转发数据包，这个过程中会话不会断开。丢包的数量也可以因此降至最低，或者根本不出现丢包的情况。 这个特性默认就是启用的。 这一版系统没有新增或修改命令
SSO-GLBP		GLBP 目前可以感知 SSO。GLBP 可以检测出路由器切换到了备用 RP，同时继续保存其在 GLBP 组中的状态。 在能够感知 SSO 之前，GLBP 无法检测到网络中还有第二个 RP，可以在主用 RP 出现故障时接替它的角色。当主用 RP 出现故障时，GLBP 设备就会停止参与 GLBP 组，同时根据其角色，主用 RP 故障可能会导致组中的另一台路由器接替主用路由器。通过这种增强特性，GLBP 可以检测到 RP 故障切换到了另一台 RP，此时 GLBP 组不会有任何变化。如果备用 RP 也发生了故障，同时主用 RP 仍然不可用，那么 GLBP 组也会检测到这个事件，并且重新选举一个新的主用 GLBP 路由器。 这个特性默认就是启用的。 这一版系统引入或修改了下列命令：**debug glbp events、glbp sso、show glbp**

词汇表

- **主用 RP**：控制系统、提供网络服务、运行路由协议，并提供系统管理接口口的路由处

理器（RP）。

- **AVF**：主用虚拟转发设备。GLBP 组中一台被选举为一个指定 MAC 地址的主用虚拟转发设备，这台设备会负责转发那些发送到这个 MAC 地址的数据包。每个 GLBP 组中可以有多个主用虚拟转发设备。
- **AVG**：主用虚拟网关。GLBP 组中一台被选举为主用虚拟网关的虚拟网关，这台设备会复杂执行协议操作。
- **GLBP 网关**：网关负载分担协议（GLBP）网关。运行 GLBP 的路由器或网关。每个 GLBP 网关可能会参与到一个或多个 GLBP 组中。
- **GLBP 组**：网关负载分担协议（GLBP）组。在相连的以太网接口上配置了相同 GLBP 组号的一台或多台 GLBP 网关。
- **ISSU**：服务期间软件升级（ISSU）。这个进程可以让设备在继续转发数据包的同时，对 Inspur INOS XE 软件进行升级或其他的修改。在大多数网络中，根据规划对软件进行升级是网络中断的主因。ISSU 可以让设备在对软件进行修改的同时继续转发数据包，这样可以增加网络的可用性，减少因为要按计划对软件进行升级而造成的网络中断。
- **NSF**：不停止转发（nonstop forwarding）。路由器继续向另一台有可能刚刚完成故障恢复的路由器转发流量的功能。这同样也是指那些刚刚从故障中恢复过来的路由器，能够继续准确地将其他设备发送过来的流量转发出去的能力。
- **RP**：路由处理器。是机框中中央控制单元的一种通用称谓。每个平台往往会有这个平台的专用术语，比如 Inspur 7500 叫作 RSP，Inspur 10000 叫作 PRE，Inspur 7600 则叫作 SUP+MSFC。
- **RPR**：路由处理器冗余。RPR 可以提供另一种高系统可用性（HAS）特性。HAS 可以在主用 RP 出现故障时重置系统并且使用备用路由处理器（RP）。通过 RPR，用户可以减少意外的宕机时间，因为 RPR 可以在主用 RP 遭遇致命错误时，实现从主用 RP 到备用 RP 的快速切换。
- **RPR+**：RPR 的增强版，在 RPR+中，备用 RP 是完全初始化的状态。
- **SSO**：状态化故障切换。可以让应用和特性在主用和备用单元之间维护状态化信息。
- **备用 RP**：即已经完全初始化，并且准备好在主用 RP 被关闭或者遭遇严重故障时接替主用 RP 的 RP。
- **故障切换**：即系统控制与路由协议执行由主用 RP 迁移到备用 RP 的过程。故障切换有可能是手动操作的结果，也有可能是因为硬件或软件故障造成的。故障切换有可能也包括迁移系统的数据包转发功能，因为有些系统就是由系统控制和数据包转发两部分有机组成的。
- **vIP**：虚拟 IP 地址。一类 IPv4 地址。每个用户配置的 GLBP 组必须有且只有 1 个虚拟 IP 地址。虚拟 IP 地址必须至少配置在一个 GLBP 组成员上。其他 GLBP 组成员可以通过 hello 消息学习这个虚拟 IP 地址。

第 6 部分 IP 组播路由

IP 组播路由技术概述

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于 IP 组播技术的信息

IP 组播在信息传输中的角色

IP 组播是一种节省带宽的技术，它可以将一组信息流同时发送给千家万户，因此可以起到减少流量的作用。使用组播技术的应用包括视频会议、企业通信、远程学习和软件分发、股票报价和新闻。

IP 组播路由可以让一台主机（即源）将数据包发送给处于 IP 网络中各个地方的一组主机（即接收方），这种技术需要使用一种特殊形式的 IP 地址来实现，这类地址称为 IP 组播组地址。发送方主机会将组播组地址诸如数据包的 IP 目的地址字段中，而 IP 组播路由器和多层交换机会将入站的 IP 组播数据包从所有连接了组播组成员的接口发送出去。任何主机都可以向组播组发送数据，无论它是不是这个组播组的成员。但只有组播组的成员可以接收到这些消息。

IP 组播路由协议

软件支持下列协议实施 IP 组播路由：

- IGMP 用于同一个局域网的主机和路由器（及多层交换机）之间，它可以追踪局域网主机所加入的组播组。要想参与 IP 组播转发，组播主机、路由器和多层设备必须运行互联网组管理协议（IGMP）；
- 协议独立组播（PIM）用于路由器之间，让路由器可以追踪应该将哪些组播数据包转发给其他路由器，应该将哪些组播数据包转发给它们直连的局域网；
- IGMP Snooping 用于在二层交换环境中发送组播流量。它可以动态配置二层接口来介绍组播流量的泛洪，让组播流量只会转发给连接了 IP 组播设备的接口。

下图显示了在 IP 组播环境中，这些协议的运行范围。

图 20：IP 组播路由协议

Host	主机
Host	主机
Access Layer	接入层
Router	路由器

根据 IPv4 组播标准，MAC 组播地址要以 0100:5e 开头，而且 MAC 组播地址的要包含 IP 地址的后 32 位。例如，如果 IP 目的地址为 239.1.1.39，那么 MAC 目的地址就是 0100:5e01:0127。当目的 IPv4 地址与目的 MAC 不匹配时，组播数据包就不会匹配。设备会根据 MAC 地址表，用硬件转发那些不匹配的组播数据包。如果目的 MAC 地址不在 MAC 地址表中，那么设备就会以同一个 VLAN 中的所有端口都作为接收方端口，并将数据包转发出去。

组播组传输体系

IP 通信由是由充当流量发送方和接收方的主机所组成的，如第 1 张图所示。其中发送方称为源。传统 IP 通信是由 1 台主机源向其他主机（单播传输）或所有主机（广播传输）发送数据包的方式实现的。IP 组播提供了第三种选择，让一台主机可以将数据包发送给所有主机的一个子集（即组播传输）。这种接收方主机的子集称为一个组播组。属于一个组播组中的主机称为组成员。

组播是以组的概念作为基础的。一个组播组是由任意多台为了接收某种数据流而加入到一个组中的接收方所组成的。这种组播组没有物理或地理层面的边界，主机可以隔离在互联网中的任何位置或者位于任何私有网络当中。主机如何对接收从一个源发送给某个特定组的流量感兴趣，它们就必须加入这个组当中。加入组的工作要由主机接收方通过互联网组管理协议（IGMP）来完成。

在一个组播环境中，任意主机都可以向组发送数据包，无论它是不是组的成员。但只有组的成员可以接收到发送给这个组的数据包。组播数据包在向组传输时，采取的是尽力而为的方式，这一点与 IP 单播数据包相同。

Unicast transmission-One host sends and other receives.	单播传输——一台主机发送，另一台主机接收
Source	源
IP network	IP 网络
Receiver	接收方

Broadcast transmission-One sender to all receivers.	广播传输——一台发送方向所有接收方发送消息
Source	源
IP network	IP 网络
Receivers	接收方
Multicast transmission-One sender to a group of receivers.	组播传输——一台发送方向一组接收方发送消息
Source	源
IP network	IP 网络
Receivers	接收方
Multicast Group	组播组

在下一张图中，接收方（指定组播组）对接收来自于源的视频数据流感兴趣。接收方通过向网络中路由器发送 IGMP 主机报告的方式表达了它们的兴趣。于是，路由器就会开始转发从源去往接收方的数据。在这里，路由器使用的是协议独立组播（PIM）来动态创建组播分发树的。接下来，视频数据流就只会被转发给那些位于源和接收方路径中的网段了。

Source	源
Multicast Group	组播组
Receiver A	接收方 A
Receiver B	接收方 B
Receiver C	接收方 C
Receiver D	接收方 D

IP 组播边界

如图所示，地址的作用范围定义了域的边界，让拥有相同 IP 地址的（包含 RP 的）域不会相互泄漏信息。作用范围需要定义在大域的子网边界上，以及域与 Internet 的边界上。

图 21：边界的地址作用范围

用户可以在一个接口上使用命令 **ip multicast boundary** 来给组播组地址设置管理范围边界，配置命令时可以包含 *access-list* 这个参数。标准访问列表可以定义受影响的地址范围。在设置好边界之后，组播数据包在双向都不允许穿越这个边界。边界可以让同一个组地址在不同的管理域中进行复用。

互联网数字分配机构（IANA）将组播地址范围 239.0.0.0 到 239.255.255.255 指定为了管理范围地址。这个范围内的地址可以在不同机构管理的域内进行复用。这些地址可以认为是本地地址，而不是全局唯一的地址。

用户可以配置关键字 **filter-autorp** 在管理范围边界查看和过滤 Auto-RP 发现和宣告消息。所有被边界访问控制列表（ACL）拒绝的 Auto-RP 数据包，其 Auto-RP 组范围通告也会被删除。只有当 Auto-RP 组范围中的所有地址边界 ACL 都可以放行时，Auto-RP 组范围通告才会被放行，并且穿越边界。如果有任何一个地址没有被允许，那么整个组范围都会被过滤，并且会在设备转发 Auto-RP 消息之前从 Auto-RP 消息中删除。

IP 组播组的编址

一个组播组是由组播组地址标识的。组播数据包会被发送给这个组播组地址。与单播地址可以唯一标识一台主机不同，组播 IP 地址并不会标识一台主机。要接收到发送给一个组播地址的数据，主机必须加入这个地址所标识的组播组当中。组播数据会被发送给组播地址，所有加入了这个组的主机也都是希望接收到发送给这个组流量的主机，因此它们都会接收到这些数据。组播地址会在源被分配给一个组。网络管理员在分配组播地址族时，一定要确保地址符合互联网数字分配机构（IANA）所保留的组播地址范围。

IP D 类地址

IANA 将 IP 组播地址分配给了 IPv4 D 类地址空间。D 类地址的最高 4 位为 1110。因此，主机组地址的范围是从 224.0.0.0 到 239.255.255.255。组播地址是在源为组播组中的接收方选择的。

注释： D 类地址范围仅用于 IP 组播流量的组地址或目的地址。组播数据报文的源地址永远都是单播源地址。

IP 组播地址作用范围

组播地址范围可以进一步细分，这样可以让组播地址在各类地址范围和较小域中实现复用。下表总结了组播地址的范围。在表格之后，有多各个范围的简单描述。

表 31：组播地址范围的划分

名称	范围	描述
保留的链路本地地址	224.0.0.0-224.0.0.255	保留给本地网段中的网络协议使用
全局范围地址	224.0.1.0-238.255.255.255	保留给机构间或跨域互联网发送组播数据使用
特定源组播	232.0.0.0-232.255.255.255	保留给 SSM 数据报传输模型使用，即只将数据发送给显式加入了组的那些接收方
GLOP 地址	233.0.0.0-233.255.255.255	保留给那些已经获得了自治系统域编号（AS）的机构静态定义的地址
限制范围地址	239.0.0。0-239.255.255.255	保留给管理范围或限制范围地址，用于私有组播域中

保留的链路本地地址

IANA 为本地网络中的网络协议保留了 224.0.0.0 到 224.0.0.255 这个范围的地址。地址在这个范围内的数据包都是本地范围的数据包，这些数据包不会发送给 IP 路由器。以链路本地地址作为目的地址的数据包往往生成时间值（TTL）会被设置为 1，因此路由器不会转发这些数据包。

在这个范围中，保留的链路本地地址可以给网络协议提供为它们保留的功能。网络协议可以使用这些地址来实现自动路由器发现，并且相互通信重要的路由信息。比如，开放式最短路径优先（OSPF）就使用了 224.0.0.5 和 224.0.0.6 来交换链路状态信息。

IANA 从 224.0.1.xxx 地址范围中为网络协议或网络应用分配了某个组播地址。组播路由器会转发这些组播地址的流量。

全局范围地址

范围在 224.0.1.0 到 238.255.255.255 之间的地址称为全局范围地址。这些地址的作用是在机

构间或跨域互联网发送组播数据。其中有些地址已经被 IANA 保留了下来，供组播应用使用。例如，IP 地址 224.0.1.1 就保留给了网络时间协议（NTP）使用。

特定源组播

范围在 232.0.0.0/8 的地址被 IANA 保留给了特定源组播（SSM）。在 Inspur INOS 软件中，用户可以使用命令 **ip pim ssm** 来给任意 IP 组播地址配置 SSM。SSM 是协议独立组播（PIM）的一种扩展，可以实现更加高效的一对多通信的数据转发机制。本文档在 IP 组播传输模式中对 SSM 进行了描述。

GLOP 地址

GLOP 编址计划（由 RFC 2770 提出，GLOP 编址范围为 233.0.0.0/8）提出，233.0.0.0/8 这个范围要保留给那些已经获得了自治系统域编号（AS）的机构静态定义的地址。这种做法称为 GLOP 编址计划。域的 AS 号会插入到 233.0.0.0/8 这个地址范围的第 2 和第 3 个十进制数中。例如，AS 620101 以十六进制的格式就是 F23A。分为两个十进制数，那么 F2 和 3A 就就等于 242 和 58。因此这些值最后得到的子网就是 233.242.58.0/24，这就是保留给 AS 62010 全局使用的地址。

限制范围地址

范围在 239.0.0.0 到 239.255.255.255 之间的地址被保留给管理范围或限制范围地址，用于私有组播域中。这些地址只能用于本地组或本地机构。企业、高校和其他机构可以使用限制范围地址来部署那些不会向其域外转发数据的组播应用。路由器在部署时往往会通过配置过滤器的方法，防止以这个地址范围内的地址作为目的组播地址的流量，流向自治系统（AS）或任何用户定义的域之外。在 AS 或域之内，限制范围地址可以进一步进行划分，以此来定义本地组播的边界。

注释： 网络管理员可以在域内使用这个范围内的组播地址，以避免和其他互联网中的流量出现冲突。

二层组播地址

过去，局域网段中的网络接口卡（NIC）只能接收到那些以它们烧录的 MAC 地址或广播 MAC 地址作为目的的数据包。但在 IP 组播中，很多主机都需要能够接收到包含一个共同目的 MAC 地址的数据。因此，必须对这种机制进行完善，才能让多台主机都能接收到同一个数据包，同时又能够区分不同的组播组。一种方法是将 IP 组播 D 类地址直接映射到 MAC 地址当中。通过这种方式，NIC 就可以接收到去往许多不同 MAC 地址的数据包了。

在与 Inspur 交换机相连的路由器上部署 Inspur 组管理协议（CGMP），就可以让路由器执行与部署 IGMP 的路由器类似的任务。CGMP 对于无法区分 IP 组播数据包和 IGMP 报告消息的 Inspur 交换机来说必不可少，因为这两类消息在 MAC 层面都会编址为同一个组播地址。

IP 组播传输模式

IP 组播传输模式只会因接收方主机而异，与源主机无关。源主机会以自己的 IP 地址作为数据包的源地址，以组地址作为数据包的 IP 目的地址来发送 IP 组播数据包。

特定源组播

特定源组播是一种数据报文传输模型，这种模型特别适合用于那些执行一对多传输的应用，也就是通称的广播应用。SSM 是 Inspur 在音频和视频广播应用环境中实施 IP 组播的核心网络技术。

对于 SSM 传输模式来说，IP 组播接收方主机必须使用 IGMP 第 3 版协议（IGMPv3）来订阅(S,G)信道。通过订阅这条信道，接收方主机也就表达了自己希望接收由源主机 S 发送给组 G 的流量。网络会将从源主机 S 发往组 G 的 IP 组播数据包，发送给网络中所有订阅了(S,G)信道的主机。
SSM 不需要在网络中分配组地址，只需要给每个源地址分配组地址即可。同一台源主机使用的不同应用必须使用不同的 SSM 组。不用源主机上运行的不同应用可以任意复用 SSM 组地址，而不用担心造成网络中产生过多的流量。

配置 IGMP

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

IGMP 和 IGMP snooping 的前提条件

IGMP 的前提条件

- 在阅读这部分文档之前，用户应该熟悉文档“IP 组播路由技术概述”部分中介绍的内容；
- 在这部分文档中，我们会假设 IP 组播已经启用，而且用户已经按照“配置基本 IP 组播路由”一部分文档中所介绍的方法配置了协议独立组播（PIM）接口。

IGMP Snooping 的前提条件

在配置 IGMP Snooping 查询器时，请留意下列知道方针：

- 在全局配置模式下配置 VLAN；

- 在 VLAN 接口下配置 IP 地址。在启用之后，IGMP Snooping 查询器会使用 IP 地址作为查询的源地址；
- 如果 VLAN 接口上配置 IP 地址，那么 IGMP Snooping 查询器就会尝试给 IGMP 查询器使用配置的全局 IP 地址。如果用户没有配置全局 IP 地址，那么 IGMP 查询器就会尝试使用 VLAN 设备虚拟接口（SVI）IP 地址（如有）。如果没有 SVI IP 地址，那么设备就会使用设备上配置的第一个可用的 IP 地址。第一个可用的 IP 地址会出现在特权 EXEC 命令 **show ip interface** 的输出信息当中。如果 IGMP snooping 查询器无法在设备上找到可用的 IP 地址，那么它就不会创建查询消息；
- IGMP snooping 查询器支持 IGMP 第 1 版和第 2 版；
- 在管理启用 IGMP snooping 查询器时，如果它检测到网络中存在组播路由器，那么它就会进入到非查询器（nonquerier）状态；
- 在管理启用 IGMP snooping 查询器时，它在如下情况下进入到操作禁用（operationally disabled）状态：
  - 该 VLAN 中禁用了 IGMP snooping；
  - 对应 VLAN 的 SVI 上启用了 PIM。

IGMP 和 IGMP snooping 的限制条件

配置 IGMP 的限制条件

下面是配置 IGMP 的限制条件：

- 设备支持 IGMP 第 1 版、第 2 版和第 3 版；

注释： IGMP 第 3 版只支持 IGMP 第 3 版的 BISS（基本 IGMPv3 Snooping 支持）。

- IGMP 第 3 版使用新的成员报告消息，而一些比较古老的 IGMP snooping 设备有可能无法识别这些消息；
- IGMPv3 可以与 ISM 和 SSM 配合使用。在 ISM 中，exclude 和 include 模式的报告都可以使用；而在 SSM 中，最后一跳路由器只支持 include 模式的报告，而 exclude 模式的报告会被忽略；
- 不能用 Inspur 3850 和 Inspur 6650 设备建立混合设备堆栈。

IGMP Snooping 的限制条件

下面是 IGMP snooping 的限制条件：

- 设备只支持基于目的组播 IP 地址的 IGMPv3 snooping，不支持基于源 IP 地址或代理报告的 snooping；
- 运行 IGMP 过滤或组播 VLAN 注册（VMR）的设备不支持 IGMPv3 加入消息和离开消息；
- 只有组播查询有 IGMPv1 和 IGMPv2 报告时，设备才支持 IGMP 报告抑制。如果查询只包含了 IGMPv3 报告，那么设备是不支持这项特性的；
- 只有在运行 IGMPv2 的主机上，才支持对 IGMP 离开时间进行配置。IGMPv2 是设备的默认模式。
- 网络的实际离开延迟往往就是配置的离开时间，但离开时间也有可能与配置的时间有所出入。具体情况取决于设备的 CPU 负载条件、网络延迟，以及通过这个接口发送的流

量多少；

- IGMP 限流（IGMP throttling）操作限制只能应用于二层端口。用户可以在逻辑 EtherChannel 接口的接口配置模式下输入命令 **ip igmp max-groups action replace**，但不能在属于 EtherChannel 端口组的端口上使用这条命令；
- 在最大组限制被设置为默认值(即无最大组限制)时，输入命令 **ip igmp max-groups action {deny | replace}**不会产生任何效果；
- 如果用户在一个接口将组播条目添加到转发表之后，配置了限流（throttling）操作并且设置了最大组限制，那么这些转发表条目就会过期或者被删除，具体做法取决于限流的操作。

关于 IGMP 的信息

互联网组管理协议的角色

IGMP 用于动态注册局域网中一个组播组中的各个主机。在接口上启用 PIM 的同时也会启用 IGMP。IGMP 可以提供一种自动控制和限制（包含特殊组播查询器和主机的）网络中组播流量的方式。

- 查询器是一类网络设备（如路由器），这类设备会发送查询消息来发现哪些网络设备是给定组播组的成员；
- 主机是接收方（也包括路由器），主机会发送报告消息（来对查询消息进行响应）来向查询器通告主机的成员身份。主机会使用 IGMP 消息来加入和离开组播组。

主机会通过向本地组播设备发送 IGMP 消息的方式，来识别组的成员设备。通过 IGMP，设备可以侦听 IGMP 消息，并且周期性地发送查询消息来发现在特定子网中，哪些组播组是活跃的（active），哪些组是不活跃的（inactive）。

IGMP 组播地址

IP 足拨片流量会使用组播地址，也就是 D 类 IP 地址。D 类地址的前 4 位为 1110。因此，主机组地址的取值范围就是从 224.0.0.0 到 239.255.255.255 之间。

范围在 224.0.0.0 到 224.0.0.255 之间的组播地址是保留给路由协议和其他网络控制流量使用的组播地址。组 224.0.0.0 不会被分配给其他组使用。

IGMP 数据包会使用 IP 组播组地址进行传输，即：

- IGMP 一般查询消息会发送给地址 224.0.0.1（子网中的所有系统）；
- IGMP 特定组查询消息会被发送给这台设备正在查询的那个组的 IP 地址；
- IGMP 组成员报告会被发送给这台设备正在报告的那个组的 IP 地址；
- IGMPv2 离开组消息会被发送给地址 224.0.0.2（子网中的所有设备）；
- IGMPv3 成员报告会被发送给地址 224.0.0.22，所有启用了 ICMPv3 的组播设备都必须侦听这个地址。

IGMP 版本

设备支持 IGMP 第 1 版、IGMP 第 2 版和 IGMP 第 3 版。这些版本可以在设备上进行互操作。例如，如果用户启用了 IGMP snooping，查询器的版本为 IGMPv2，而这台设备又从主机那里接收到了 IGMPv3 报告，那么这台设备可以将这个 IGMPv3 转发给组播路由器。

IGMPv3 设备可以从运行特定源组播特性的设备那里接收到消息，也可以向它转发消息。

IGMP 第 1 版

IGMP 第 1 版（IGMPv1）主要用于查询-响应模型，它可以让组播路由器和多层设备发现本地子网中有哪些组播组是活动的（也就是这个组播组中有一台或多台对其感兴趣的主机）。IGMPv1 有其他进程可以让主机加入和离开一个组播组。要想了解详细信息，可以参见 RFC 1112。

IGMP 第 2 版

IGMPv2 对 IGMP 的功能进行了扩展，提供了一些诸如 IGMP 离开进程这样的特性，因此 IGMPv2 减少了离开延迟、特定组查询和显式最大查询响应时间。IGMPv2 还增加了路由器的功能，让路由器可以不依赖组播协议就选举出 IGMP 查询器。要想了解详细信息，可以参见 RFC 2236。

注释： ICMP 第 2 版是设备的默认版本。

IGMP 第 3 版

设备支持 IGMP 第 3 版。

IGMPv3 设备支持基本 IGMPv3 Snooping 支持（BISS），这种特性包括支持 IGMPv1 和 IGMPv2 的 snooping 特性，以及支持 IGMPv3 的成员报告消息。当网络中包含 IGMPv3 主机时，BISS 可以限制组播流量的泛洪。它可以将流量限制在与针对 IGMPv2 或 IGMPv1 主机运行 IGMP snooping 特性时几乎相同的端口。

IGMPv3 设备可以从运行特定源组播特性的设备那里接收到消息，也可以向它转发消息。

IGMPv3 主机信令

IGMPv3 是 IETF 标准追踪协议的第 3 个版本。在这一版中，主机会将成员身份发送给组播组的最后一跳设备。IGMPv3 引入了主机发送组成员身份的功能，这让网络可以针对源进行过滤。主机可以通过信令表示自己希望从除了某些特定源之外（即 EXCLUDE 模式）所有向某个组发送流量的源那里接收流量，也可以标识自己希望只从某些特定向组发送流量的源（即 INCLUDE 模式）那里接收流量。

IGMPv3 可以与 ISM 和 SSM 配合使用。在 ISM 中，EXCLUDE 和 INCLUDE 模式的报告都可以使用；而在 SSM 中，最后一跳路由器只支持 INCLUDE 模式的报告，而 EXCLUDE 模式的报告会被忽略

IGMP 版本的差异

根据 IETF（互联网工程任务组） RFC（请求评论）文档的定义，IGMP 有 3 个版本。IGMPv2 在 IGMPv1 的基础上进行了强化，让主机可以通过发送信令表达自己希望离开组播组；IGMPv3 又在 IGMPv2 的基础上进行了升级，让主机可以侦听仅从某些源 IP 地址发送出来的组播流量。

表 32：IGMP 版本

IGMP 版本	描述
IGMPv1	提供基本的查询-响应机制，让组播设备可以判断哪些组播组是活动的，同

	时提供了让主机加入和离开组播组的进程。RFC 1112 定义了 IGMPv1 是如何将 IP 组播转发扩展到主机的
IGMPv2	对 IGMP 进行了扩展，支持诸如 IGMP 离开进程、特定组查询和显式最大响应时间。IGMPv2 也给设备添加了不依赖组播协议选举 IGMP 查询器的功能。IGMPv2 定义在 RFC 2236 中

注释： 在默认情况下，在接口上启用 PIM 的同时，设备也会启用 IGMPv2。IGMPv2 在设计上希望可以尽量向后兼容 IGMPv1。为了做到向后兼容，RFC 2236 定义了特殊的互操作规则。如果网络中包含了传统的 IGMPv1 主机，那么用户就需要熟悉这些互操作规则。要想详细了解关于 IGMPv1 和 IGMPv2 互操作的信息，可以参见 RFC 2236，互联网组管理协议第 2 版。

运行 IGMPv1 的设备

IGMPv1 设备会向“所有主机”组播地址 224.0.0.1 发送 IGMP 查询消息，来请求包含活动组播接收方的组播组。组播接收方也会发送 IGMP 报告，来通告自己对接收某个组播流感兴趣。主机可以异步发送报告，也可以对设备发送的 IGMP 查询消息进行响应。如果同一个组播组中有超过过一台组播接收方，那么其中只有一台设备可以发送 IGMP 报告消息，其他主机会抑制自己的报告消息。

在 IGMPv1 中，不存在 IGMP 查询器选举。如果同一个网段中拥有多台设备，所有设备都会周期性第发送 IGMP 查询消息。IGMPv1 也没有提供主机离开组的特殊机制。如果主机对接收某个组的数据流已经不感兴趣，它们只是会不再回复设备发送的 IGMP 查询数据包。而设备还是会继续发送查询数据包。如果连续 3 个 IGMP 查询消息没有得到响应，这个组就会超时，设备也就不会在这个网络中给这个组发送组播数据包。如果一段时间之后，主机又想接收组播数据包，它只需要向设备发送一条信的 IGMP 加入消息，设备就会重新开始向其转发组播数据包了。

如果局域网中有多台设备，就会选举出一台指定路由器（DR）以避免不同设备给相连的主机发送重复的组播流量。PIM 设备会按照一个选举流程来选择 DR。拥有最高 IP 地址的 PIM 设备会最终称为 DR。

DR 负责执行下列任务：

- DR 负责向汇聚点（RP）发送 PIM 注册和 PIM 加入与修剪消息，以向其通告主机的组成员身份；
- 发送 IGMP 主机-查询消息；
- 默认每 60 秒发送一次主机-查询消息，以保证主机和网络中的 IGMP 管理流量比例不高。

运行 IGMPv2 的设备

IGMPv2 提高了 IGMPv1 的查询消息功能。

IGMPv2 中的查询与成员关系报告消息与 IGMPv1 消息相同，只有两处例外：

- IGMPv2 查询消息分为两类：一般查询消息（相当于 IGMPv1 查询消息）和特定组查询消息；
- IGMPv1 成员关系报告与 IGMPv2 成员关系报告使用的 IGMP 类型代码不同。

IGMPv2 还借助下列功能对 IGMP 进行了强化：

- 查询器选举进程：让 IGMPv2 设备可以不借助组播路由协议来选举 IGMP 查询器；
- 最大响应时间字段：查询消息中的这个信字段可以让 IGMP 查询器指定最大的查询响应时间。这个字段可以让用户调节查询-响应进程，来控制流量的突发，并具体调节离开组的延迟时间；
- 特定组查询消息：可以让 IGMP 查询器针对特定的组进行查询，而不是针对所有组进行查询；
- 离开组消息：让主机可以通告网络中的设备它们希望离开某个组。

在 IGMPv1 环境中，DR 和 IGMP 查询器往往是同一台设备。而在 IGMPv2 环境中，这两类设备的功能已经解耦。DR 和 IGMP 查询器会根据不同的标准进行选择，因此它们也有可能是同一个子网中的不同设备。DR 是子网中 IP 地址最高的设备，而 IGMP 查询器则是 IP 地址最低的设备。

查询消息的作用是按照下面的方式选举 IGMP 查询方：

1. 当 IGMPv2 设备启动时，它们会分别向全系统组播地址 224.0.0.1 发送一条一般查询消息，该消息的源 IP 地址字段填写的是设备自己接口的地址；
2. 当一台 IGMPv2 设备接收到一条一般查询消息时，这台设备会将消息中的源 IP 地址与自己的接口地址进行比较。子网中 IP 地址最低的设备会被选举为 IGMP 查询器；
3. 所有设备（但不包括查询器）启动查询计时器，每当设备从 IGMP 查询器那里接收到一条一般查询消息时，查询计时器就会重置。如果查询计时器超时，设备就会认为这台 IGMP 查询器出现了故障，因此设备会再次启动选举进程，来选举信的 IGMP 查询器。

在默认情况下，计时器的时间是查询时间间隔的 2 倍。

IGMP 加入与离开进程

IGMP 加入进程

当一台主机想要加入一个组播组中，它会主动给它想要加入的这个组播组发送一条或几条的成员报告。IGMP 加入进程对于 IGMPv1 和 IGMPv2 主机来说都是相同的。

在 IGMPv3 中，主机加入进程是这样操作的：

- 当主机想要加入一个组时，它会向 224.0.0.22 发送一条 IGMPv3 成员报告，报告的 EXCLUDE 列表为空；
- 当主机想要计入一条特定的信道，它会向 224.0.0.22 发送一条 IGMPv3 成员报告，报告中的 INCLUDE 列表中包含特定源的地址；
- 当主机想要加入一个组，但不想接收某些源的信息时，它会向 224.0.0.22 发送一条 IGMPv3 成员报告，报告的 EXCLUDE 列表中包含这些要排除的源。

注释： 如果局域网中一部分 IGMPv3 主机想要排除某个源，其他主机却希望包含这个源，那么设备会在局域网中发送这个源的流量（也就是说，包含的优先级高于排除）。

IGMP 离开进程

主机离开一个组的方式取决于 IGMP 的操作版本。

IGMPv1 的离开进程

在 IGMPv1 协议中，没有离开组消息向子网中的设备通告一台主机已经不希望继续接收某个组的组播流量了。这台主机知识不会继续处理该组播组的流量，同时也不会响应 IGMP 成员报告对这个组所作的 IGMP 查询。因此，只有当 IGMPv1 设备已经接收到不同成员关系报告时，它们才能发现这个子网中已经没有了这个组的活动接收方。为了简化这个过程，IGMPv1 设备对子网中的 IGMP 组配备了一个倒计时计时器。每当设备接收到某个组的成员关系报告时，它就会重置计时器。对于 IGMPv1 设备来说，超时时间间隔往往是查询时间间隔的 3 倍（即 3 分钟）。这样的超时时间间隔表示，在所有组都离开组播组之后，设备最多可能会在未来 3 分钟继续向子网中转发组播流量。

IGMPv2 的离开进程

IGMPv2 包含了离开组消息，这种消息可以让主机表达它希望停止接收某个组的组播流量。当一台 IGMPv2 主机离开组播组时，如果它是这个组中最后一台会响应成员报告查询消息的主机，那么它就会向所有设备组播组地址（224.0.0.2）发送一条离开组消息。

IGMPv3 的离开进程

IGMPv3 强化了离开进程，它可以让主机在 IGMPv3 成员关系报告中，通过包含或排除某些源、组或信道的方式，来表达自己不再希望继续接收某个组、某个源或者某条信道中流量。

IGMP Snooping

二层设备可以使用 IGMP snooping 来限制组播流量的泛洪，这种技术可以动态对二层接口进行配置，让组播流量只通过那些连接了 IP 组播设备的接口进行发送。顾名思义，IGMP snooping 要求局域网设备对主机与路由器之间传输的 IGMP 流量进行窥探（snoop），以追踪组播组和成员端口。当设备从一台主机那里接收到某个组播组的一条 IGMP 报告，这台设备就会将主机端口号添加到转发表条目中。当它从一台主机那里接收到一条 IGMP 离开组消息，它就会将主机端口从表中删除。如果设备长期没有从组播客户端那里接收到 IGMP 成员报告消息，它也会周期性地删除条目。

注释： 要想了解关于 IP 组播和 IGMP 的详细信息，可以参见 RFC 1112 和 RFC 2236。

组播路由器（有可能是一台使用 IP services 特性集的主用设备）会向所有 VLAN 周期性发送一般查询消息。所有对组播流量感兴趣的主机都会发送加入请求，并且被添加到转发表条目当中。设备会给在 IGMP snooping IP 组播转发表中给每个接受到了 IGMP 加入请求消息的组，针对各个 VLAN 分别创建一个条目。

设备支持基于 IP 组播组（而不是基于 MAC 地址组）的桥接。对于基于组播 MAC 地址的组，如果用户配置的 IP 地址会转换（别名）为一个之前配置的 MAC 地址或者某个保留的组播 MAC 地址（224.0.0.xxx 范围内的地址），那么这条命令也不会生效。因为设备会使用 IP 组播组，没有地址别名的问题。

通过 IGMP snooping 学习到的 IP 组播组是动态的。但用户可以使用全局配置命令 **ip igmp snooping vlan** *vlan-id* **static** *ip_address* **interface** *interface-id* 来静态配置组播组。如果用户给一个组播组地址静态设置组成员，那么用户所作的配置会优越 IGMP snooping 自动生成的操作。组播组成员列表中会包含用户定义的设置，和通过 IGMP snooping 学习到的设置。

用户可以在没有组播接口的子网中配置 IGMP snooping 查询器来支持 IGMP snooping，因为组播流量不需要进行路由。

如果端口生成树、端口组或 VLAN ID 发生了变更，那么通过 VLAN 中这个端口学习到的那些 IGMP snooping 组播组就会被设备删除。

在这一节中，我们会介绍 IGMP snooping 的特征。

加入一个组播组

当一台与设备相连的主机希望加入某个 IP 组播组时，如果这台设备是一台 IGMPv2 客户端的话，它会主动发送一条 IGMP 加入消息，标识自己想要加入的 IP 组播组。此外，当设备从路由器那里接收到一般查询消息时，它会将查询消息从这个 VLAN 中的所有端口转发出去。如果 IGMPv1 或 IGMPv2 主机希望加入组播组，会用向设备发送加入消息的方式作出响应。如果设备上没有这个组的组播转发表条目，那么设备的 CPU 就会给这个组创建一个组播转发表条目。CPU 也会将接收到加入消息的那个接口一起添加到转发表条目当中。这个接口所连接的主机会接收到发送给这个组播组的组播流量。

图 22：初始 IGMP 加入消息

Router A	路由器 A
IGMP report	IGMP 报告
Forwarding table	转发表
Host 1	主机 1

Host 2	主机 2
Host 3	主机 3
Host 4	主机 4

路由器 A 向设备发送了一条一般查询消息，设备将这条查询消息从端口 2 到端口 5 转发了出去，所有这些端口都是同一个 VLAN 的成员端口。主机 1 希望加入组播组 224.1.2.3，因此用组播的形式向组中发送了一条 IGMP 成员关系报告（IGMP 加入消息）。设备 CPU 使用 IGMP 报告中的信息建立了一个转发表条目，其中包含了连接主机 1 和连接路由器的端口的端口号。

表 33：IGMP Snooping 转发表

目的地址	数据包类型	端口
224.1.2.3	IGMP	1、2

设备硬件可以将 IGMP 信息的数据包从发送给这个组播组的其他数据包中区分出来。表中的信息让交换引擎能够将以 224.1.2.3 这个组播 IP 地址作为目的地址，同时又不是 IGMP 数据包的数据发送给路由器，以及加入到组中的主机。

如果另一台主机（比如主机 4）主动给同一个组发送了一个 IGMP 加入消息，CPU 会接收到这个消息，并且将主机 4 所连接的端口号添加到转发表中。由于转发表只会将 IGMP 消息转交给 CPU，因此这个消息不会通过设备的其他端口进行泛洪。所有已知的组播流量都会被转发给这个组，而不会转发给 CPU。

图 23：加入组播组的第二台主机

Router A	路由器 A
IGMP report	IGMP 报告
Forwarding table	转发表
Host 1	主机 1
Host 2	主机 2
Host 3	主机 3
Host 4	主机 4

表 34：更新后的 IGMP Snooping 转发表

目的地址	数据包类型	端口
224.1.2.3	IGMP	1、2、5

离开组播组

路由器会周期性地发送一般查询消息，而设备会将这些查询消息通过 VLAN 中的所有端口转发出去。感兴趣主机会对查询消息作出响应。只要 VLAN 中还有一台主机希望接收组播流量，那么路由器就会继续将组播流量转发给这个 VLAN。设备只会把组播组流量转发给转发表中（IGMP Snooping）针对这个 IP 组播组所列出的那些主机。

当主机希望离开组播组时，它们可以自行安静地离开，也可以发送一条离开消息。当设备接收到主机发送的离开消息时，它会发送一条特定组查询消息，以便了解是否还有其他与这个接口相连的设备仍然对接收这个组播组的流量感兴趣。接下来，设备会对转发表中的这个 MAC 组进行更新，确保只有对接收这个组的组播流量感兴趣的主机才会列出在转发表中。如果路由器没有从 VLAN 中接收到报告，那么路由器就会从自己的 IGMP 缓存中，对这个 VLAN 移除该组。

直接离开（Immediate Leave）

设备可以使用 IGMP Snooping 直接离开（Immediate Leave）特性将一个发送了离开消息的接口从转发表中移除，同时又不需要向该接口发送特定组查询消息。在这个离开消息所指定的组播组中，这个 VLAN 接口会从组播树中被修剪掉。直接离开特定可以确保交换网络中所有

主机都获得最佳的带宽分配，即使网络中同时使用了多个组播组。
只有 IGMP 第 2 版主机支持直接离开特性。IGMPv2 是设备的默认版本。
注释： 用户只应该在每个端口只连接了一台主机的 VLAN 上启用直接离开特性。如果在多台主机连接在同一个端口的 VLAN 中启用了直接离开特性，那么设备可能会在不经意间被设备丢弃。

IGMP 可配置的离开计时器

设备会在发送特定组查询消息，以判断主机是否仍然对接收一个特定组播组的流量感兴趣之后等待一段时间，这个时间长度是可以进行配置的。IGMP 离开响应时间的取值范围是从 100 到 32767 毫秒。

IGMP 报告抑制

注释： 只有组播查询有 IGMPv1 和 IGMPv2 报告时，设备才支持 IGMP 报告抑制。如果查询只包含了 IGMPv3 报告，那么设备是不支持这项特性的。

使用 IGMP 报告抑制特性之后，设备针对每个组播路由器查询消息只会向组播设备转发 1 个 IGMP 报告。在启用了 IGMP 报告抑制之后（默认即启用），设备会将所有主机发给组的第 1 个 IGMP 报告发送给组中的所有组播路由器。设备不会将这个组其后的 IGMP 报告发送给组播路由器。这个特性可以防止设备将重复报告发送给组播设备。
如果组播路由器查询消息中只包含了 IGMPv1 和 IGMPv2 报告的请求消息，那么设备就只会将所有主机发给组的第 1 个 IGMPv1 或 IGMPv2 报告发送给组中的所有组播路由器。
如果组播请求消息也包含了请求 IGMPv3 报告，那么设备就会将所有 IGMPv1、IGMPv2 和 IGMPv3 报告转给组中的组播设备。
如果禁用了 IGMP 报告抑制特性，那么所有 IGMP 组播消息都会被发送给组播路由器。

IGMP Snooping 与设备堆栈

IGMP snooping 功能可以跨越堆栈中的设备生效。也就是说，1 台设备的 IGMP 控制信息会被分发给堆栈中的所有设备。无论 IGMP 组播数据是通过哪个堆栈成员设备进入堆栈的，数据最终都会到达对这个组的流量感兴趣的主机。
如果堆栈中有一台设备发生了故障，或者从堆栈中被移除了，那么只有这台设备连接的组播组成员无法接收到组播数据。堆栈中其他设备上连接的组播组其他成员还是可以继续接收到组播数据流。但是但主用设备被移除时，二层和三层组播组（IP 组播路由）可能会多花一点时间才能完成收敛。

IGMP 过滤与限流

在有些环境中，例如在都会区或多用户居住单元（MDU）的部署环境中，用户可能希望控制一些组播组的成员用户所连接的端口。用户可以基于一些订阅或服务规划类型，来限制组播服务（如 IP/TV）的分发。用户可能还希望限制设备端口所连用户能够加入多少个组播组。
通过 IGMP 过滤特性，用户可以以端口为单元过滤组播加入消息，用户可以配置 IP 组播配置文件，并且将它与设备端口进行关联。IGMP 配置文件中可能会包含一个或多个组播组，并且指定用户是否允许访问这个或这些组播组。如果用户将一个不允许访问某个组播组的 IGMP 配置文件应用到了设备的某个端口，那么设备就会丢弃请求 IP 组播流量的 IGMP 加入报告，这个端口也不允许接收该组的 IP 组播流量。如果过滤的操作是允许访问这个组播组，那么这个端口接收到的 IGMP 报告就会按照正常的流程进行转发。用户也可以设置一个二层接口能够加入的最大 IGMP 组数量。
IGMP 过滤只会控制特定组查询和成员报告，包括加入和离开报告。它不会控制一般 IGMP 查询。IGMP 过滤与设备转发 IP 组播流量的功能无关。过滤特性与是否使用 CGMP 或 MVR 来转发组播流量的方式相同。
IGMP 过滤只能应用于设备动态学习到 IP 组播组地址，不能应用于静态配置。

通过 IGMP 限流（IGMP throttling）特性，用户可以限制一个二层接口可以加入的最大 IGMP 组数量。如果设置了最大 IGMP 组数量，IGMP snooping 转发表中就只会包含最大数量的条目。用户可以当接口接收到 IGMP 加入报告时，是丢弃这个 IGMP 报告，还是用接收到的 IGMP 报告来替换到随机选择的组播条目。

注释： 运行 IGMP 过滤的设备上是不支持 IGMPv3 加入与离开消息的。

默认的 IGMP 配置

这张表显示了设备的默认 IGMP 配置。

表 35：默认的 IGMP 配置

特性	**默认设置**
多层设备充当组播组的成员	没有定义组成员设备
访问组播组	接口上可以访问所有组
IGMP 版本	所有接口上皆为第 2 版
IGMP 主机-查询消息时间间隔	所有接口上皆为 60 秒
IGMP 查询超时时间	所有接口上皆为 60 秒
IGMP 最大查询响应时间	所有接口上皆为 10 秒
多层设备充当静态连接成员	禁用

默认 IGMP Snooping 配置

这张表显示了设备默认的 IGMP snooping 配置

表 36：默认的 IGMP Snooping 配置

特性	**默认设置**
IGMP Snooping	在全局和每个 VLAN 上启用
组播路由器	未配置
IGMP Snooping 直接离开特性	禁用
静态组	未配置
TCN①-泛洪查询数	2
TCN 查询请求	禁用
IGMP snooping 查询器	禁用
IGMP 报告抑制	启用

① TCN=拓扑变更通告（Topology Change Notification）——原注

默认 IGMP 过滤与限流配置

这张表显示了设备上默认的 IGMP 过滤与限流配置。

表 37：默认的 IGMP 过滤配置

特性	**默认设置**
IGMP 过滤器	未应用
IGMP 组最大数量	没有设置最大数量 **注释：**当转发表中的组达到最大数量时，默认 IGMP 限流就会拒绝 IGMP 报告
IGMP 配置文件	未定义
IGMP 配置文件操作	拒绝这个范围的地址

如何配置 IGMP

将设备配置为组成员（CLI）

用户可以将设备配置为组播组的成员，并且发现网络中组播的可达性。如果用户管理的所有具有组播功能的路由器及多层设备都是一个组播组的成员，那么如果向这个组发起 ping 测试，所有设备都会进行响应。当设备自己是组成员时，设备就会响应发送给那个组的 ICMP echo 请求数据包。另一个例子是软件中提供的组播追踪路由工具。

注意： 执行这项操作可能会影响 CPU 的性能，因为 CPU 会接收到所有去往这个组地址的数据流量。

这项配置是可选的。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. ip igmp join-group *group-address*

5. end

6. show ip igmp interface [*interface-id*]

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	指定希望启用组播路由的三层接口，并进入接口配置模式。 设置的接口必须是下面两类之一： • 路由端口：通过在接口配置模式下输入命令 **no switchport**，设置为三层端口的物理端口。用户还需要在这个接口上启用 IP PIM 稀疏-密集模式，并将这个接口作为静态连接的组成员加入 IGMP 静态组当中； • SVI：使用全局配置命令 **interface vlan** *vlan-id* 创建的 VLAN 接口。用户还需要在这个 VLAN 上启用 IP PIM 稀疏-密集模式，并将这个 VLAN 作为静态连接的组成员加入 IGMP 静态组当中，然后在这个 VLAN、IGMP 静态组和物理接

		口上启用 IGMP snooping。 用户必须给这些接口分配 IP 地址
步骤 4	**ip igmp join-group** *group-address* 示例： `Device(config-if)#` **ip igmp join-group 225.2.2.2**	配置设备加入一个组播组。 在默认情况下，设备上没有定义组成员身份。 在 *group-address* 部分用点分十进制设置 IP 地址
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show ip igmp interface** [*interface-id*] 示例： `Device#` **show ip igmp interface**	查看配置的条目
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

控制对 IP 组播组的访问（CLI）

设备会通过发送 IGMP 主机-查询消息来寻找哪些组播组在直连本地网络上有成员设备。接下来，设备会将所有目的地为组播组的数据包都转发给这些组成员。用户可以在每个接口上配置一个过滤器（filter），来限制这个子网中的主机可以加入的组播组。
要在接口上限制加入的数量，可以在接口上配置一个与 IGMP 配置文件相关联的配置文件。这项配置是可选的。

总步骤

1. enable
2. configure terminal
3. ip igmp profile
4. permit
5. exit
6. interface *interface-id*
7. ip igmp filter *filter_number*
8. end
9. show ip igmp interface [*interface-id*]
10. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式

步骤 3	**ip igmp profile**
	
	示例：
	`Device(config)# ip igmp profile 10`
	`Device(config-igmp-profile)# ?`
		输入 IGMP 过滤器配置文件编号，取值范围是从 1 到 4294967295。
		要想了解更多关于配置 IGMP 过滤器配置文件的信息，可以参阅配置 IGMP 配置文件

步骤 4	**permit** 示例： `Device(config-igmp-profile)# permit 229.9.9.0`	输入 IGMP 配置文件的配置行为。设备支持配置下列 IGMP 配置文件的配置行为： • **deny**：匹配的 IP 地址会被拒绝； • **exit**：从 IGMP 配置文件的配置模式中离开； • **no**：取消一条命令或将其恢复默认值； • **permit**：匹配的 IP 地址会被放行； • **range**：向这个集中添加一个范围
步骤 5	**exit** 示例： `Device(config-igmp-profile)# exit`	返回全局配置模式
步骤 6	**interface** *interface-id* 示例： `Device(config)# interface gigabitethernet 1/0/1`	指定要配置的三层接口，并进入接口配置模式
步骤 7	**ip igmp filter** *filter_number* 示例： `Device(config-if)# ip igmp filter 10`	设置 IGMP 过滤器配置文件的编号。 要想了解更多关于应用 IGMP 过滤器配置文件的信息，可以参阅应用 IGMP 配置文件（CLI）
步骤 8	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 9	**show ip igmp interface** [*interface-id*]	查看配置的条目

	示例： `Device#` **show ip igmp interface**	
步骤 10	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

修改 IGMP 版本（CLI）

在默认情况下，交换机会使用 IGMP 第 2 版，这个版本的协议可以提供诸如 IGMP 查询超时和最大查询响应时间等特性。

子网中的所有系统都必须支持相同的版本。交换机不会自动检测版本 1 的系统，并且将自己的 IGMP 版本切换为版本 1。用户可以在子网中同时使用第 1 版和第 2 版的 IGMP，因为运行第 2 版协议的路由器或交换机能够完美兼容使用 IGMPv1 的主机。

如果主机不支持第 2 版 IGMP，就配置交换机来运行 IGMPv1，

这项配置是可选的。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. ip igmp version {**1** | **2** | **3** }

5. end

6. show ip igmp interface [*interface-id*]

7. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	指定要配置的三层接口，并进入接口配置模式
步骤 4	**ip igmp version** {**1**	**2**	**3** }	设置交换机使用的 IGMP 版本。

	示例： `Device(config-if)#` **ip igmp version 2**	**注释**：如果将版本修改为版本 1，用户就不能再输入接口配置命令 **ip igmp query-interval** 或 **ip igmp query-max-response-time**。 要回到默认设置，可以使用接口配置命令 **no ip igmp version**。
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show ip igmp interface** [*interface-id*] 示例： `Device#` **show ip igmp interface**	查看配置的条目
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

修改 IGMP 主机-查询消息时间间隔（CLI）

设备会周期性地发送 IGMP 主机-查询消息，来发现自己直连网络上连接了哪些组播组。这些消息会被发送给全主机组播组（224.0.0.1），且这些消息的生存时间值（TTL）为 1。设备会发送主机-查询消息来更新自己掌握的网络中的组成员关系。如果在发送了一些数量的查询消息之后，软件发现某个组播组没有本地成员主机，那么软件就会停止从远端向本地网络的转发这个组的组播数据包，并且会向源发送一条上游修剪消息。

设备会在局域网（子网）中选举出一台 PIM 指定路由器（DR）。指定路由器负责向局域网中的所有主机发送 IGMP 主机-查询消息。在稀疏模式中，指定路由器也会向 RP 路由器发送 PIM 注册消息和 PIM 加入消息。如果使用 IGMPv2，那么 DR 就是 IP 地址最高的路由器或者多层设备。如果使用 IGMPv1，那么 DR 就会根据局域网中运行的组播路由协议进行选择。

这项配置是可选的。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. ip igmp query-interval *seconds*

5. end

6. show ip igmp interface [*interface-id*]

7. copy running-config startup-config

具体步骤

	命令或操作	目的

步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)# interface gigabitethernet 1/0/1`	指定希望启用组播路由的三层接口，并进入接口配置模式。 设置的接口必须是下面两类之一： • 路由端口：通过在接口配置模式下输入命令 **no switchport**，设置为三层端口的物理端口。用户还需要在这个接口上启用 IP PIM 稀疏-密集模式，并将这个接口作为静态连接的组成员加入 IGMP 静态组当中； • SVI：使用全局配置命令 **interface vlan** *vlan-id* 创建的 VLAN 接口。用户还需要在这个 VLAN 上启用 IP PIM 稀疏-密集模式，并将这个 VLAN 作为静态连接的组成员加入 IGMP 静态组当中，然后在这个 VLAN、IGMP 静态组和物理接口上启用 IGMP snooping。 用户必须给这些接口分配 IP 地址
步骤 4	**ip igmp query-interval** *seconds* 示例： `Device(config-if)# ip igmp query-interval 75`	配置指定路由器（DR）发送 IGMP 主机-查询消息的频率。 在默认情况下，指定路由器每 60 秒发送一次 IGMP 主机-查询消息，这是为了保证主机和网络中的 IGMP 管理流量保持在一个非常低的比例。 取值范围是从 1 到 65535
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show ip igmp interface** [*interface-id*] 示例： `Device# show ip igmp interface`	查看配置的条目
步骤 7	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中

给 IGMPv2 修改 IGMP 查询超时时间

如果用户使用的是 IGMPv2，那么用户可以设置设备接替接口查询器前的时间周期。在默认情况下，设备会等待两倍的查询间隔周期，而查询间隔周期可以通过接口配置命令 **ip igmp query-interval** 进行设置。在这段时间之后，如果设备没有接收到查询消息，它就会成为查询器。

这项配置是可选的。

总步骤

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip igmp querier-timeout** *seconds*
5. **end**
6. **show ip igmp interface** [*interface-id*]
7. **copy running-config startup-config**

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	指定希望启用组播路由的三层接口，并进入接口配置模式。 设置的接口必须是下面两类之一： • 路由端口：通过在接口配置模式下输入命令 **no switchport**，设置为三层端口的物理端口。用户还需要在这个接口上启用 IP PIM 稀疏-密集模式，并将这个接口作为静态连接的组成员加入 IGMP 静态组当中； • SVI：使用全局配置命令 **interface vlan** *vlan-id* 创建的 VLAN 接口。用户还需要在这个 VLAN 上启用 IP PIM 稀疏-密集模式，并将这个 VLAN 作为静态连接的组成员加入 IGMP 静态组当中，然后在这个 VLAN、IGMP 静态组和物理接口上启用 IGMP snooping。 用户必须给这些接口分配 IP 地址
步骤 4	**ip igmp querier-timeout** *seconds*	设置 IGMP 查询超时时间。 默认值为 60 秒（查询间隔时间的 2 倍）。取值范围是从 60 到 300

	示例： `Device(config-if)#` **ip igmp querier-timeout 120**	
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show ip igmp interface** [*interface-id*] 示例： `Device#` **show ip igmp interface**	查看配置的条目
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

给 IGMPv2 修改最大查询响应时间（CLI）

如果使用 IGMPv2，用户可以修改 IGMP 查询消息中通告的最大查询响应时间。最大查询响应时间可以让设备快速检测出自己在局域网中已经没有直连的组成员设备了。降低这个参数的取值可以让设备更快对组进行修剪。

这项配置是可选的。

总步骤

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip igmp query-max-response-time** *seconds*
5. **end**
6. **show ip igmp interface** [*interface-id*]
7. **copy running-config startup-config**

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式

步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	指定希望启用组播路由的三层接口，并进入接口配置模式。 设置的接口必须是下面两类之一： • 路由端口：通过在接口配置模式下输入命令 **no switchport**，设置为三层端口的物理端口。用户还需要在这个接口上启用 IP PIM 稀疏-密集模式，并将这个接口作为静态连接的组成员加入 IGMP 静态组当中； • SVI：使用全局配置命令 **interface vlan** *vlan-id* 创建的 VLAN 接口。用户还需要在这个 VLAN 上启用 IP PIM 稀疏-密集模式，并将这个 VLAN 作为静态连接的组成员加入 IGMP 静态组当中，然后在这个 VLAN、IGMP 静态组和物理接口上启用 IGMP snooping。 用户必须给这些接口分配 IP 地址
步骤 4	**ip igmp query-max-response-time** *seconds* 示例： `Device(config-if)#` **ip igmp query-max-response-time 15**	修改 IGMP 查询消息中通告的最大查询响应时间。默认值为 10 秒。取值范围是从 1 到 25
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show ip igmp interface** [*interface-id*] 示例： `Device#` **show ip igmp interface**	查看配置的条目
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

将设备配置为静态连接的成员（CLI）

有时，局域网段中已经没有了某个组的成员，或者只有一台不能使用 IGMP 通告其组成员身份的主机。不过，用户可能希望将组播流量发送给那个网段。下列命令的作用是将组播流量推送给网段：

- **ip igmp join-group**：设备会接受组播数据包，也会转发组播数据包。接受组播数据包可

以防止设备执行快速交换；

- **ip igmp static-group**：设备不接受组播数据包，它只会对组播数据包进行转发。这种方式会启用快速交换。出站接口会添加到 IGMP 缓存中，但设备自身并不是组的成员，证据是组播路由条目中没有 L（本地）标记。

这项配置是可选的。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. ip igmp static-group *group-address*

5. end

6. show ip igmp interface [*interface-id*]

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	指定希望启用组播路由的三层接口，并进入接口配置模式。 设置的接口必须是下面两类之一： • 路由端口：通过在接口配置模式下输入命令 **no switchport**，设置为三层端口的物理端口。用户还需要在这个接口上启用 IP PIM 稀疏-密集模式，并将这个接口作为静态连接的组成员加入 IGMP 静态组当中； • SVI：使用全局配置命令 **interface vlan** *vlan-id* 创建的 VLAN 接口。用户还需要在这个 VLAN 上启用 IP PIM 稀疏-密集模式，并将这个 VLAN 作为静态连接的组成员加入 IGMP 静态组当中，然后在这个 VLAN、IGMP 静态组和物理接口上启用 IGMP snooping。 用户必须给这些接口分配 IP 地址
步骤 4	**ip igmp static-group** *group-address* 示例： `Device(config-if)#` **ip igmp static-group 239.100.100.101**	将设备配置为一个组静态连接的成员。 这项特性默认是禁用的

步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show ip igmp interface** [*interface-id*] 示例： `Device#` **show ip igmp interface**	查看配置的条目
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 IGMP 配置文件（CLI）

用户可以按照下面的步骤来创建 IGMP 配置文件：
这项任务是可选的。

总步骤

1. enable
2. configure terminal
3. ip igmp profile *profile number*
4. permit | **deny**
5. range *ip multicast address*
6. end
7. show ip igmp profile *profile number*
8. show running-config
9. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip igmp profile** *profile number* 示例：	给正在配置的配置文件分配一个编号，并进入 IGMP 配置文件配置模式。配置文件的编号范围是从 1 到 4294967295。在 IGMP 配置文件配置模式

	Device(config)# **ip igmp profile 3**	中，用户可以使用下面这些命令来创建配置文件： • **deny**：让匹配的 IP 地址被拒绝。这是默认操作； • **exit**：从 IGMP 配置文件的配置模式中离开； • **no**：取消一条命令或将其恢复默认值； • **permit**：让匹配的 IP 地址被放行； • **range**：给这个配置文件设置一个 IP 地址范围。用户可以输入一个 IP 地址，或者一个 IP 地址范围的起点与终点。 设备在默认状态下没有配置 IGMP 配置文件。 **注释**：要删除配置文件，需要输入全局配置命令 **no ip igmp profile** *profile number*
步骤 4	**permit** | **deny** 示例： Device(config-igmp-profile)# **permit**	（可选）设置允许或拒绝访问这个 IP 组播地址。如果用户不进行配置，那么配置文件默认会拒绝访问
步骤 5	**range** *ip multicast address* 示例： Device(config-igmp-profile)# **range 229.9.9.0**	输入要进行访问控制的 IP 组播地址或 IP 组播地址范围。如果用户输入的是一个范围，那就要先输入最低的 IP 组播地址，然后键入空格，最后输入最高的 IP 组播地址。 用户可以反复使用 **range** 命令输入多个地址范围。 **注释**：要删除一个 IP 地址或者一个 IP 地址范围，需要输入 IGMP 配置文件配置命令 **no range** *ip multicast address*
步骤 6	**end** 示例： Device(config)# **end**	返回特权 EXEC 模式
步骤 7	**show ip igmp profile** *profile number* 示例： Device# **show ip igmp profile 3**	查看配置文件的配置
步骤 8	**show running-config** 示例： Device# **show running-config**	查看配置的条目
步骤 9	**copy running-config startup-config** 示例： Device# **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

应用 IGMP 配置文件（CLI）

要想按照 IGMP 配置文件中定义的方式来控制访问，用户需要将配置文件应用到响应的接口上。用户只能将 IGMP 配置文件应用到二层 Access 端口上，而不能将 IGMP 配置文件应用到路由端口或 SVI 上。用户不能将配置文件应用到属于一个 EtherChannel 端口组的端口上。用户可以将配置文件应用到多个接口上，但每个接口上只能应用一个配置文件。
用户可以按照下面的步骤将一个 IGMP 配置文件应用到交换机端口上：

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. ip igmp filter *profile number*

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	指定一个物理接口并进入接口配置模式。这个接口必须是一个二层端口，且不属于 EtherChannel 端口组
步骤 4	**ip igmp filter** *profile number* 示例： `Device(config-if)#` **ip igmp filter 321**	将指定的 IGMP 配置文件应用到接口。配置文件编号的取值范围是从 1 到 4294967295。 **注释：**要从接口上移除配置文件，需要输如接口配置命令 **igmp filter** *profile number*
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show ip igmp interface** [*interface-id*] 示例：	查看配置的条目

	`Device#` **show ip igmp interface**	
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

设置 IGMP 组的最大数量（CLI）

用户可以按照下面的步骤来设置一个二层接口可以加入的最大 IGMP 组数量：

在开始前

这个限制条件只适用于二层端口。用户不能在路由端口或 SVI 接口上设置 IGMP 组的最大数量。用户也可以在逻辑 EtherChannel 接口上使用这条命令，但不能在属于 EtherChannel 端口组的端口上使用这条命令。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. ip igmp max-groups *number*

5. end

6. show running-config interface *interface-id*

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	指定要配置的物理接口并进入接口配置模式。这个接口可以是一个不属于 EtherChannel 组的二层端口，也可以是一个 EtherChannel 接口
步骤 4	**ip igmp max-groups** *number* 示例： `Device(config-if)#` **ip igmp max-groups 20**	设置这个接口可以加入的最大 IGMP 组数量。取值范围是从 0 到 4294967294。默认状态是没有设置最大值。 **注释：**设备支持的最大 4096 个二层 IGMP 组和 2048 个三层 IGMP 组

步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show ip igmp interface** [*interface-id*] 示例： `Device#` **show ip igmp interface**	查看配置的条目
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 IGMP 限流操作（CLI）

在设置了二层接口可以加入的最大 IGMP 组数量之后，用户可以配置一个接口，让它用接收 IGMP 报告的组来取代当前的组。

用户可以按照下面的步骤来配置当转发表中的条目达到最大值时，设备执行的限流操作。

总步骤

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip igmp max-groups action** {**deny** | **replace**}
5. **end**
6. **show running-config interface** *interface-id*
7. **copy running-config startup-config**

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface**	指定要配置的物理接口并进入接口配置模式。这个接口可以是一个不属于 EtherChannel 组的二层端口，也可以是一个 EtherChannel 接口。这个接口不能是 trunk 端口

	gigabitethernet 1/0/1	
步骤 4	**ip igmp max-groups action** {**deny** | **replace**} 示例： `Device(config-if)#` **ip igmp max-groups action replace**	当接口接收到一个 IGMP 报告，而转发表中的条目又到达了最大值时，这条命令可以设置设备执行的限流操作： • **deny**：丢弃这个报告。如果配置的是这个限流操作，那么之前转发表中的条目不会被移除，但会过期。在这些条目过期之后，若转发表中的条目达到最大数量，设备就会丢弃这个接口此后接收到的 IGMP 报告； • **replace**：用接收到 IGMP 报告的新组替换当前的组。如果配置的是这个限流操作，那么之前转发表中的条目就会被移除。在转发表中的条目达到最大数量时，设备会用接收到的 IGMP 报告替换一个随机选择的条目。 要防止设备移除转发表中的条目，可以在接口向转发表中添加条目之前，先配置 IGMP 限流的操作。 **注释**：要还原默认的丢弃报告这种操作方式，可以使用接口配置命令 **no ip igmp max-groups action**
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show ip igmp interface** [*interface-id*] 示例： `Device#` **show ip igmp interface**	查看配置的条目
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置设备使其在没有直连 IGMP 主机时依然转发组播流量

用户可以根据需求执行下面的操作，让设备在没有直连 IGMP 主机时依然转发组播流量。

总步骤

1. enable

2. configure terminal

3. interface *type number*

4. 执行下列步骤之一：

• **ip igmp join-group** *group-address*

• **ip igmp static-group** {***** | *group-address* [**source** *source-address*]}

5. end

6. show ip igmp interface [*interface-type interface-number*]

7. copy running-config startup-config

8. show running-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *type number* 示例： `Device(config)# interface gigabitethernet 1`	进入接口配置模式。 • 对于 *type* 和 *number* 参数，指定与主机直连的接口

步骤 4	执行下列操作之一：
	
	• **ip igmp join-group** *group-address*
	• **ip igmp static-group** {***** | *group-address*
	[**source** *source-address*]}
	
	示例：
	`Device(config-if)# ip igmp join-group 225.2.2.2`
	示例：
	`Device(config-if)# ip igmp static-group 225.2.2.2`
		第 1 个示例显示了如何在设备上配置一个接口，让其加入指定的组。
		• 通过这种方式，设备会接受组播数据包，也会转发组播数据包。接受组播数据包可以防止设备执行快速交换。
		第 2 个示例显示了如何在接口上配置静态组成员条目。
		• 通过这种方式，设备本身不接受数据包，它只会对它们进行转发。因此，这种方式会启用快速交换。出站接口会添加到 IGMP 缓存中，但设备自身并不是组的成员，证据是组播路由条目中没有 L（本地）标记。

步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show ip igmp interface** [*interface-type interface-number*] 示例： `Device# show ip igmp`	（可选）显示这个接口与组播有关的信息

	interface	
步骤 7	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中
步骤 8	**show running-config** 示例： `Device# show running-config`	查看配置的条目

使用 IGMP 扩展的访问列表来控制对 SSM 网络的访问

用户可以根据需求执行下面的操作，使用 IGMP 扩展的访问列表来基于源地址、组地址或基于这两项参数，来控制对 SSM 网络的访问。

总步骤

1. enable

2. configure terminal

3. ip multicast-routing [**distributed**]

4. ip pim ssm {**default** | **range** *access-list*}

5. ip access-list extended *access-list* -name

6. deny igmp *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*]
[**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

7. permit igmp *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

8. exit

9. interface type number

10. ip igmp access-group *access-list*

11. ip pim sparse-mode

12.在所有需要对SSM信道成员身份进行控制的接口上重复步骤1到步骤11

13. ip igmp version 3

14.在所有面向主机的接口上补充步骤13

15. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device# configure terminal`	
步骤 3	**ip multicast-routing** [**distributed**] 示例： `Device(config)# ip multicast-routing distributed`	启用 IP 组播路由。 • 对于 IPv4 组播，需要输入关键字 **distributed**
步骤 4	**ip pim ssm** {**default** | **range** *access-list*} 示例： `Device(config)# ip pim ssm default`	配置 SSM 服务。 • 使用关键字 **default** 会将 SSM 范围访问列表定义为 232/8； • 关键字 **range** 的作用是设置定义 SSM 范围的标准 IP 访问列表的编号或名称
步骤 5	**ip access-list extended** *access-list* -name 示例： `Device(config)# ip access-list extended mygroup`	设置扩展命名 IP 访问列表
步骤 6	**deny igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] 示例： `Device(config-ext-nacl)# deny igmp host 10.1.2.3 any`	（可选）过滤 IGMP 报告中的特定源地址或组地址，以限制子网中的主机成为(S,G)信道的成员。 • 重复这 1 步来限制子网中的主机称为其他(S,G)信道的成员。（这些源应该比后面的 **permit** 语句更加具体，因为任何没有明确放行的源或组都会被拒绝）； • 切记，访问列表最后有一条隐式的 **deny** 语句； • 这个示例显示了如何创建一条 deny 语句，来过滤源 10.1.2.3 加入所有组，这条语句可以有效拒绝这个源
步骤 7	**permit igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] 示例： `Device(config-ext-nacl)#`	（可选）允许 IGMP 报告中的特定源地址或组地址通过 IP 访问列表。 • 一个访问列表中至少要配置 1 个 **permit** 语句； • 重复这 1 步来允许其他源通过 IP 访问列表； • 这个示例显示了如何允许向没有被之前的 **deny** 语句拒绝的源和组建立组成员关系

	`permit igmp any any`	
步骤 8	**exit** 示例： `Device(config-ext-nacl)# exit`	离开当前的配置模式，并返回全局配置模式
步骤 9	**interface** *type number* 示例： `Device(config)# interface ethernet 0`	选择一个与可以启用 IGMPv3 的主机相连的接口
步骤 10	**ip igmp access-group** *access-list* 示例： `Device(config-if)# ip igmp access-group mygroup`	将特定访问列表应用于 IGMP 报告
步骤 11	**ip pim sparse-mode** 示例： `Device(config-if)# ip pim sparse-mode`	在接口上启用 PIM-SM **注释**：在这里必须使用稀疏模式
步骤 12	在所有需要对 SSM 信道成员执行访问控制的接口上重复步骤 1 到步骤 11	——
步骤 13	**ip igmp version 3** 示例： `Device(config-if)# ip igmp version 3`	在这个接口上启用 IGMPv3。IGMP 默认的版本是 IGMP 第 2 版。SSM 需要使用 IGMPv3
步骤 14	在所有面向主机的接口上重复步骤 13	——
步骤 15	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

如何配置 IGMP Snooping

启用 IGMP Snooping

总步骤

1. enable

2. configure terminal

3. ip igmp snooping

4. bridge-domain *bridge-id*

5. ip igmp snooping

6. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip igmp snooping** 示例： `Device(config)# ip igmp snooping`	在 IGMP snooping 被禁用后，使用这条命令在全局启用 IGMP snooping
步骤 4	**bridge-domain bridge-id** 示例： `Device(config)# bridge-domain 100`	（可选）进入桥域配置模式
步骤 5	**ip igmp snooping** 示例： `Device(config-bdomain)# ip igmp snooping`	（可选）在正在配置的桥域接口上启用 IGMP snooping • 只有当特定敲域上曾经显式禁用了 IGMP snooping 时，才需要这行这一步操作
步骤 6	**end** 示例： `Device(config-bdomain)# end`	返回特权 EXEC 模式

在一个 VLAN 接口上禁用或启用 IGMP Snooping（CLI）

用户可以按照下面的步骤在一个 VLAN 接口上启用 IGMP snooping：

总步骤

1. enable

2. configure terminal

3. ip igmp snooping vlan *vlan-id*

4. end

5. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip igmp snooping vlan** *vlan-id* 示例： `Device(config)# ip igmp snooping vlan 7`	在 VLAN 接口上启用 IGMP snooping。VLAN ID 范围是从 1 到 1001，从 1006 到 4094。 用户必须先全局启用 IGMP snooping，然后才能启用 VLAN snooping **注释**：要在一个 VLAN 接口上禁用 IGMP snooping，可以对这个 VLAN 编号使用全局配置命令 **no ip igmp snooping vlan** *vlan-id*
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中

设置 snooping 方式（CLI）

在转发表中，每个二层组播条目都有对应的具有组播功能的路由器端口。设备会通过下面两种方法之一来学习端口：

- 对 IGMP 查询消息、协议独立组播（PIM）数据包执行 snooping，；
- 使用全局配置命令 **ip igmp snooping mrouter** 静态连接到一个组播路由器端口

用户可以从特权 EXEC 模式中，通过下列步骤修改 VLAN 接口访问组播路由器的方式：

总步骤

1. enable

2. configure terminal

3. ip igmp snooping vlan *vlan-id* **mrouter interface** {**GigabitEthernet** | **Port-Channel** | **TenGigabitEthernet**}

4. end

5. show ip igmp snooping

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密码

	示例： `Device> enable`	
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip igmp snooping vlan** *vlan-id* **mrouter interface** {**GigabitEthernet** | **Port-Channel** | **TenGigabitEthernet**} 示例： `Device(config)# ip igmp snooping vlan 1 mrouter interface GigabitEthernet1/0/3`	在 VLAN 上启用 IGMP snooping。VLAN ID 的范围是从 1 到 1001，从 1006 到 4094
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show ip igmp snooping** 示例： `Device# show ip igmp snooping`	查看所做的配置
步骤 6	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中

配置组播路由器端口（CLI）

用户可以执行下面的步骤在设备上添加一个组播路由器端口（启用一条去往组播路由器的静态连接）。

注释： 只有设备端口上支持配置去往组播路由器的静态连接。

总步骤

1. **enable**
2. **configure terminal**
3. **ip igmp snooping vlan** *vlan-id* **mrouter interface** *interface-id*
4. **end**
5. **show ip igmp snooping mrouter** [**vlan** *vlan-id*]
6. **copy running-config startup-config**

具体步骤

	命令或操作	**目的**

步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip igmp snooping vlan** *vlan-id* **mrouter interface** {**GigabitEthernet** | **Port-Channel** | **TenGigabitEthernet**} 示例： `Device(config)# ip igmp snooping vlan 1 mrouter interface GigabitEthernet1/0/3`	设置组播路由器 VLAN ID 以及连接组播路由器的接口。 • VLAN ID 的范围是从 1 到 1001，从 1006 到 4094； • 接口可以是物理接口或 port channel。后者的取值范围是从 1 到 128。 **注释**：要从VLAN中移除一个组播路由器端口，需要输入全局配置命令**no ip igmp snooping vlan** *vlan-id* **mrouter interface** *interface-id*
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show ip igmp snooping mrouter** [**vlan** *vlan-id*] 示例： `Device# show ip igmp snooping mrouter vlan 5`	查看 VLAN 接口上是否启用了 IGMP Snooping
步骤 6	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中

对主机进行配置让它静态加入一个组（CLI）

主机或二层端口往往会动态加入组播组当中，但用户也可以在接口上静态配置主机。

用户可以按照下面的步骤将一个二层端口作为组成员添加到一个组播组中：

总步骤

1. enable

2. configure terminal

3. ip igmp snooping vlan *vlan-id* **static** *ip_address* **interface** *interface-id*

4. end

5. show ip igmp snooping groups

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip igmp snooping vlan** *vlan-id* **static** *ip_address* **interface** *interface-id* 示例： `Device(config)# ip igmp snooping vlan 105`	静态将一个二层端口配置为一个组播组的成员： • *vlan-id* 为组播组的 VLAN ID。取值范围是从 1 到 1001，从 1006 到 4094； • ip-address 为组 IP 地址； • interface-id 为成员端口。这个接口可以是物理接口也可以是 port channel（1-128）。 **注释**：要从组播组中移除这个二层端口，需要输入全局配置命令**no ip igmp snooping vlan** *vlan-id* **static** *mac-address* **interface** *interface-id*
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show ip igmp snooping groups** 示例： `Device# show ip igmp snooping groups`	查看成员端口与 IP 地址
步骤 6	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中

启用 IGMP 直接离开（CLI）

当用户启用 IGMP 直接离开特性时，那么当设备在端口上检测到一条 IGMPv2 离开消息时，它就会立刻移除这个端口。只有当该 VLAN 中每个端口都只连接了一台接收方时，用户才应该在这个 VLAN 中使用直接离开特性。

注释： 只有 IGMP 第 2 版主机才支持直接离开特性。IGMPv2 是设备默认的 IGMP 版本。

总步骤

1. enable

2. configure terminal

3. ip igmp snooping vlan *vlan-id* **immediate-leave**

4. end

5. show ip igmp snooping vlan *vlan-id*

6. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip igmp snooping vlan** *vlan-id* **immediate-leave** 示例： `Device(config)#` **ip igmp snooping vlan 21 immediate-leave**	在 VLAN 接口上启用 IGMP 直接离开特性。 **注释：**要在一个VLAN上禁用IGMP直接离开特性，需要输入全局配置命令**no ip igmp snooping vlan** *vlan-id* **immediate-leave**
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show ip igmp snooping vlan** *vlan-id* 示例： `Device#` **show ip igmp snooping vlan 21**	查看 VLAN 接口上是否启用了直接离开（Immediate Leave）特性
步骤 6	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 IGMP 离开计时器（CLI）

用户可以在全局配置离开时间，也可以针对各个 VLAN 分别配置离开时间。用户可以按照下面的步骤启用并配置 IGMP 离开计时器：

总步骤

1. enable

2. configure terminal

3. ip igmp snooping last-member-query-interval *time*

4. ip igmp snooping vlan *vlan-id* **last-member-query-interval** *time*

5. end

6. show ip igmp snooping

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip igmp snooping last-member-query-interval** *time* 示例： `Device(config)#` **ip igmp snooping last-member-query-interval 1000**	在全局配置 IGMP 离开计时器。取值范围是 100 到 32767 毫秒。 **注释：**要在全局将IGMP离开计时器重置为默认设置，需要输入全局配置命令**no ip igmp snooping last-member-query-interval**
步骤 4	**ip igmp snooping vlan** *vlan-id* **last-member-query-interval** *time* 示例： `Device(config)#` **ip igmp snooping vlan 210 last-member-query-interval 1000**	（可选）在 VLAN 接口上配置 IGMP 离开时间。取值范围是 100 到 32767 毫秒。 **注释：**在 VLAN 上配置的离开时间会优于全局配置的计时器。 **注释：**要从一个特定VLAN中移除用户所配置的IGMP离开时间设置，需要输入全局配置命令**no ip igmp snooping vlan** *vlan-id* **last-member-query-interval**
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show ip igmp snooping** 示例： `Device#` **show ip igmp snooping**	（可选）显示用户配置的 IGMP 离开时间
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

相关主题

配置 IGMP 稳健性（Robustness）变量（CLI）

用户可以使用下面的步骤在设备上配置 IGMP 稳健性变量。

稳健性变量（robustness variable）是 IGMP snooping 在计算 IGMP 消息时会使用的一个整数。稳健性变量可以让用户按照自己的需求来调整丢包的几率。

总步骤

1. enable

2. configure terminal

3. ip igmp snooping robustness-variable *count*

4. ip igmp snooping vlan *vlan-id* **robustness-variable** *count*

5. end

6. show ip igmp snooping

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip igmp snooping robustness-variable** *count* 示例： `Device(config)# ip igmp snooping robustness-variable 3`	配置IGMP稳健性变量。取值范围是1到3倍。 稳健性变量的推荐配置值为2。使用这条命令可以将IGMP snooping稳健性变量值由默认值（2）修改为用户设置的数值
步骤 4	**ip igmp snooping vlan** *vlan-id* **robustness-variable** *count* 示例： `Device(config)#ip igmp snooping vlan 100 robustness-variable 3`	（可选）在 VLAN 接口上配置稳健性变量。取值范围是 1 到 3 倍。稳健性变量的推荐配置值为 2。 **注释：**在 VLAN 上配置的稳健性变量会优于全局配置的值。
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show ip igmp snooping** 示例： `Device# show ip igmp snooping`	（可选）显示用户配置的 IIGMP 稳健性变量数

步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 IGMP 最后成员查询计数（CLI）

要设置当设备接收到 IGMP 特定组或特定组-源离开消息时，会发送几次 IGMP 特定组或特定组-源查询消息，需要使用下面的命令。

总步骤

1. enable

2. configure terminal

3. ip igmp snooping last-member-query-count *count*

4. ip igmp snooping vlan *vlan-id* **last-member-query-count** *count*

5. end

6. show ip igmp snooping

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip igmp snooping last-member-query-count** *count* 示例： `Device(config)#` **ip igmp snooping last-member-query-count 3**	配置IGMP最后成员查询数。取值范围是1到7条消息，默认值为2条消息
步骤 4	**ip igmp snooping vlan** *vlan-id* **last-member-query-count** *count* 示例： `Device(config)#`**ip igmp snooping vlan 100 last-member-query-count 3**	（可选）在 VLAN 接口上配置 IGMP 最后成员查询数。取值范围是 1 到 7 条消息。 **注释：**在 VLAN 上配置的最后成员查询数会优于全局配置的值。
步骤 5	**end**	返回特权 EXEC 模式

	示例： `Device(config)# end`	
步骤 6	**show ip igmp snooping** 示例： `Device#` **show ip igmp snooping**	（可选）显示用户配置的 IIGMP 稳健性变量数
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置与 TCN 相关的命令

控制 TCN 事件后的组播泛洪次数

用户可以配置在出现拓扑变更通告（TCN）事件之后，设备泛洪一般查询的数量。如果用户将 TCN 泛洪查询数量设置为 1，那么在接收到 1 条一般查询消息之后，泛洪就会停止。如果设置为 7，那么在接收到 7 跳一般查询消息之前，泛洪都会持续。在 TCN 事件中，设备会根据接收到的一般查询消息来重新学习组。

TCN 事件有很多种情形，比如客户端的位置发生了变化，而接收方连接在同一个之前处于阻塞状态，目前则正在转发数据的端口；再比如端口没有发送离开消息就直接关闭。

用户可以使用下面的步骤来配置 TCN 泛洪查询数：

总步骤

1. enable

2. configure terminal

3. ip igmp snooping tcn flood query count *count*

4. end

5. show ip igmp snooping

6. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip igmp snooping tcn flood query count** *count* 示例： `Device(config)#` **ip igmp**	设置组播流量泛洪的IGMP一般查询消息数量。 取值范围是从1到10。默认的泛洪查询数为2。 **注释**：要还原到默认的泛洪查询数，可以输入全局配置命令**no ip igmp snooping tcn flood query count**

	snooping tcn flood **query count 3**	
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show ip igmp snooping** 示例： `Device#` **show ip igmp snooping**	查看 TCN 的设置
步骤 6	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

从泛洪默认中恢复

当拓扑出现变化时，生成树的根会使用组播地址 0.0.0.0 发送一条特殊的 IGMP 离开消息（也称为全局离开消息）。但用户可以让设备无论是否是生成树的根，都发送全局离开消息。当路由器接收到这种特殊的离开消息时，它会立刻发送一般查询消息，其目的是加速 TCN 事件期间泛洪模式的进程。如果设备是生成树的根，那么无论是否进行下面的配置，设备都会发送离开消息。

用户可以按照下面的步骤启用发送离开消息的操作。

总步骤

1. enable

2. configure terminal

3. ip igmp snooping tcn query solicit

4. end

5. show ip igmp snooping

6. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip igmp snooping tcn query solicit**	发送IGMP离开消息（全局离开）以加速网络从TCN事件导致的泛洪模式中恢复过来。在默认情况下，查询请求（query solicitation）消息是禁用的。

	示例： `Device(config)#` **ip igmp snooping tcn query solicit**	**注释**：要还原到默认的查询请求操作，可以输入全局配置命令**no ip igmp snooping tcn query solicit**
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show ip igmp snooping** 示例： `Device#` **show ip igmp snooping**	查看 TCN 的设置
步骤 6	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

禁用 TCN 事件中的组播泛洪（CLI）

当设备接收到一个 TCN 时，它会从所有端口泛洪组播流量，直至自己接收到 2 个一般查询消息位置。如果设备有很多端口连接了订阅不同组播组的主机，那么这种泛洪操作可能会导致链路超载，并由此引发丢包。用户可以按照下面的步骤来控制 TCN 泛洪：

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. no ip igmp snooping tcn flood

5. end

6. show ip igmp snooping

7. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例：	指定要配置的接口，并接入接口的配置模式

	Device(config)# **interface gigabitethernet 1/0/1**	
步骤 4	**no ip igmp snooping tcn flood** 示例： Device(config-if)# **no ip igmp snooping tcn flood**	禁用生成树TCN事件过程中的组播流量泛洪。 在默认情况下，接口上会启用组播泛洪操作。 **注释**：要重新启用组播泛洪操作，可以输入接口配置命令**ip igmp snooping tcn flood**
步骤 5	**end** 示例： Device(config)# end	返回特权 EXEC 模式
步骤 6	**show ip igmp snooping** 示例： Device# **show ip igmp snooping**	查看 TCN 的设置
步骤 7	**copy running-config startup-config** 示例： Device# **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 IGMP Snooping 查询器（CLI）

用户可以按照下面的步骤在 VLAN 中启用 IGMP Snooping 查询器：

总步骤

1. enable

2. configure terminal

3. ip igmp snooping querier

4. ip igmp snooping querier address *ip_address*

5. ip igmp snooping querier query-interval *interval-count*

6. ip igmp snooping querier tcn query [**count** *count* | **interval** *interval*]

7. ip igmp snooping querier timer expiry *timeout*

8. ip igmp snooping querier version *version*

9. end

10. show ip igmp snooping vlan *vlan-id*

11. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： Device> **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device#` **configure terminal**	
步骤 3	**ip igmp snooping querier** 示例： `Device(config)#` **ip igmp snooping querier**	启用 IGMP snooping 查询器
步骤 4	**ip igmp snooping querier address** *ip_address* 示例： `Device(config)#` **ip igmp snooping querier address 172.16.24.1**	（可选）给 IGMP snooping 查询器设置 IP 地址。如果不设置 IP 地址，查询器就会尝试使用给 IGMP 查询器配置的全局 IP 地址。 **注释：**如果 IGMP snooping 查询器无法在设备上找到一个 IP 地址，它就不会生成 IGMP 一般查询
步骤 5	**ip igmp snooping querier query-interval** *interval-count* 示例： `Device(config)#` **ip igmp snooping querier query-interval 30**	（可选）设置 IGMP 查询器之间的时间间隔。取值范围是从 1 到 18000 秒
步骤 6	**ip igmp snooping querier tcn query** [**count** *count* | **interval** *interval*] 示例： `Device(config)#` **ip igmp snooping querier tcn query interval 20**	（可选）设置拓扑变更通告（TCN）查询消息之间的事件。其中参数 *count* 参数的取值范围是 1 到 10。参数 *interval* 的取值范围是 1 到 255 秒
步骤 7	**ip igmp snooping querier timer expiry** *timeout* 示例： `Device(config)#` **ip igmp snooping querier timer expiry 180**	（可选）设置 IGMP 查询器超时前经历的时间长度。取值范围是 60 到 300 秒
步骤 8	**ip igmp snooping querier version** *version* 示例： `Device(config)#` **ip igmp snooping querier version 2**	（可选）选择查询器特性使用的 IGMP 版本。选择 1 或 2

步骤 9	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 10	**show ip igmp snooping vlan** *vlan-id* 示例： `Device#` **show ip igmp snooping vlan 30**	（可选）查看 VLAN 接口上是否已经启用了 IGMP snooping 查询器。VLAN ID 范围是从 1 到 1001，从 1006 到 4094
步骤 11	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

禁用 IGMP 报告抑制（CLI）

用户可以按照下面的步骤禁用 IGMP 报告抑制特性。

总步骤

1. enable

2. configure terminal

3. no ip igmp snooping report-suppression

4. end

5. show ip igmp snooping

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**no ip igmp snooping report-suppression** 示例： `Device(config)#` **no ip igmp snooping report-suppression**	禁用IGMP报告抑制。如果禁用了报告抑制特性，那么所有IGMP报告就都会被转发给组播路由器。 IGMP报告抑制默认是启用的。 当IGMP报告抑制特性启用时，设备只会针对每条组播路由器查询消息转发1个IGMP报告。 **注释**：要重新启用IGMP报告抑制特性，可以输入全局配置命令**ip igmp snooping report-suppression**
步骤 4	**end**	返回特权 EXEC 模式

	示例： `Device(config)# end`	
步骤 5	**show ip igmp snooping** 示例： `Device#` **show ip igmp snooping**	查看 IGMP 报告抑制是否已经禁用
步骤 6	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

监控 IGMP

用户可以显示特定的统计数据，包括 IP 路由表、缓存和数据库中的信息。

注释： 这个版本不支持针对每条路由显示统计数据。

用户可以查看信息，以了解资源的使用，并解决网络中的问题。用户也可以查看关于节点可达性的信息，并且查看设备的数据包会按照什么路由路径在网络中转发。

用户可以使用下表中的任何一条特权 EXEC 命令来显示各类路由统计信息。

表 38：显示系统与网络统计数据的命令

命令	目的
ping [*group-name* | *group-address*]	向组播组地址发送一条 ICMP Echo 请求消息
show ip igmp filter	显示 IGMP 过滤器信息
show ip igmp groups [*type-number* | *detail*]	显示设备直连的组播组，以及通过 IGMP 学习到的组播组
show ip igmp interface [*type number*]	显示某个接口上与组播相关的信息
show ip igmp membership [*name/group address* | **all** | **tracked**]	显示用来执行转发的 IGMP 成员身份信息
show ip igmp profile [*profile_number*]	显示 IGMP 配置文件信息
show ip igmp ssm-mapping [*hostname/IP address*]	显示 IGMP SSM 映射信息
show ip igmp static-group {**class-map** [**interface** [*type*]]	显示静态组信息
show ip igmp vrf	根据用户输入的名称显示用户选择的 VPN 路由转发实例

监控 IGMP Snooping 信息

用户可以查看动态学习到或静态配置的路由器端口及 VLAN 接口的 IGMP snooping 信息，也可以查看某个配置了 IGMP snooping 的 VLAN 中的 MAC 地址条目。

表 39：查看 IGMP Snooping 信息的命令

命令	目的
show ip igmp snooping detail	显示操作状态信息。
show ip igmp snooping groups [**count** | [**vlan** *vlan-id* [*A.B.C.D* | **count**]]	显示设备上或关于某个特定参数的组播表信息： • **count**：显示组的总数； • **vlan**：根据 VLAN ID 显示组信息
show ip igmp snooping mrouter [**vlan** *vlan-id*]	显示动态学习和手动配置的组播路由器接口上的信息。 **注释**：在启用 IGMP snooping 时，设备会自动学习到组播路由器连接的接口。这些就是动态学习到的接口。 （可选）输入 **vlan** *vlan-id* 来显示关于某一个 VLAN 的信息
show ip igmp snooping querier [**detail** | **vlan** *vlan-id*]	显示 VLAN 中最近接收到 IGMP 查询消息的 IP 地址和接收端口，以及相关信息。 （可选）输入 **detail** 让设备显示 VLAN 中详细的 IGMP 查询器信息； （可选）输入 **vlan** *vlan-id* 让设备显示关于某一个 VLAN 的信息
show ip igmp snooping [**vlan** *vlan-id* [**detail**]]	显示设备上所有 VLAN 的 snooping 配置信息，或显示设备上某个额特定 VLAN 的 snooping 配置信息。 （可选）输入 **vlan** *vlan-id* 让设备显示关于某一个 VLAN 的信息。VLAN 的取值范围是从 1 到 1001，从 1006 到 4094

监控 IGMP 过滤与限流的配置

用户可以查看 IGMP 配置文件特征信息，也可以查看设备上所有接口或某个特定接口上的 IGMP 配置文件及最大组配置。用户还可以查看设备上所有接口或某个特定接口上的 IGMP 限流配置。

命令	目的
show ip igmp profile [*profile number*]	显示特定的 IGMP 配置文件或设备上定义的所有 IGMP 配置文件
show running-config [**interface** *interface-id*]	显示特定接口或者设备上所有接口的配置，包括一个接口可以加入的 IGMP 最大组数量（如配置），以及应用在接口上的 IGMP 配置文件

IGMP 的配置示例

示例：将设备子配置为一个组的成员

这个示例显示了如何配置让设备加入组播组 225.2.2.2：

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip igmp join-group 255.2.2.2
Device(config-if)#
```

示例：控制对组播组的访问

要限制接口加入的数量，可以让接口按照 IGMP 配置文件实施过滤。

```
Device# configure terminal
Device(config)# ip igmp profile 10
Device(config-igmp-profile)# ?
IGMP profile configuration commands:
deny matching addresses are denied
exit Exit from igmp profile configuration mode
no Negate a command or set its defaults
permit matching addresses are permitted
range add a range to the set
Device(config-igmp-profile)# range 172.16.5.1
Device(config-igmp-profile)# exit
Device(config)#
Device(config)# interface gigabitEthernet 2/0/10
Device(config-if)# ip igmp filter 10
```

示例：配置 IGMP Snooping

这个示例显示了如何向一台组播路由器启用一条静态路由：

```
Device# configure terminal
Device(config)# ip igmp snooping vlan 200 mrouter interface gigabitethernet1/0/2
Device(config)# end
```

这个示例显示了如何在一个端口上静态配置一台主机：

```
Device# configure terminal
Device(config)# ip igmp snooping vlan 105 static 224.2.4.12 interface gigabitethernet1/0/1
Device(config)# end
```

这个示例显示了如何在 VLAN 130 上启用 IGMP 直接离开特性：

```
Device# configure terminal
Device(config)# ip igmp snooping vlan 130 immediate-leave
Device(config)# end
```

这个示例显示了如何将 IGMP snooping 查询器最大响应事件设置为 25 秒：

```
Device# configure terminal
Device(config)# ip igmp snooping querier query-interval 25
Device(config)# end
```

这个示例显示了如何将 IGMP snooping 查询器超时时间设置为 60 秒：

```
Device# configure terminal
Device(config)# ip igmp snooping querier timer expiry 60
Device(config)# end
```

这个示例显示了如何将 IGMP snooping 查询器特性设置为第 2 版：

```
Device# configure terminal
Device(config)# no ip igmp snooping querier version 2
Device(config)# end
```

示例：配置 IGMP 配置文件

这个示例显示了 rue 好创建 IGMP profile 4（编号 4 的配置文件），来放行去往一个 IP 组播地址的访问连接，以及如何查看相关的配置。如果操作是拒绝（这是默认操作），那么这项操作应该可以通过命令 **show ip igmp profile** 的输出信息显示出来。

```
Device(config)# ip igmp profile 4
Device(config-igmp-profile)# permit
Device(config-igmp-profile)# range 229.9.9.0
Device(config-igmp-profile)# end
Device# show ip igmp profile 4
IGMPProfile 4
permit
range 229.9.9.0 229.9.9.0
```

示例：应用 IGMP 配置文件

这个示例显示了如何将 IGMP profile 4 应用到一个端口：

```
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# ip igmp filter 4
Device(config-if)# end
```

示例：设置 IGMP 组最大数量

这个示例显示了如何将一个端口可以加入的 IGMP 组数量限制为 25 个：

```
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# ip igmp max-groups 25
Device(config-if)# end
```

示例：将接口配置为一个路由端口

这个示例显示了如何将设备上的一个接口配置为路由端口。在很多 IP 组播路由配置流程中，用户都需要执行这项操作，这个过程中需要运行命令 **no switchport**。

```
Device configure terminal
Device(config)# interface GigabitEthernet1/0/9
Device(config-if)# description interface to be use as routed port
Device(config-if)# no switchport
Device(config-if)# ip address 20.20.20.1 255.255.255.0
Device(config-if)# ip pim sparse-dense-mode
Device(config-if)# ip igmp join-group 224.1.2.3 source 15.15.15.2
Device(config-if)# end
Device# configure terminal
Device# show run interface gigabitEthernet 1/0/9
Current configuration : 166 bytes
!
interface GigabitEthernet1/0/9
no switchport
ip address 20.20.20.1 255.255.255.0
ip pim sparse-dense-mode
ip igmp static-group 224.1.2.3 source 15.15.15.2
end
```

示例：将接口配置为一个 SVI

这个示例显示了如何将设备上的一个接口配置为 SVI。在很多 IP 组播路由配置流程中，用户都需要执行这项操作，这个过程中需要运行命令 **no switchport**。

```
Device(config)# interface vlan 150
Device(config-if)# ip address 20.20.20.1 255.255.255.0
Device(config-if)# ip pim sparse-dense-mode
Device(config-if)# ip igmp join-group 224.1.2.3 source 15.15.15.2
Device(config-if)# end
Device# configure terminal
Device(config)# ip igmp snooping vlan 20 static 224.1.2.3
interface gigabitEthernet 1/0/9
Device# show run interface vlan 150
Current configuration : 137 bytes
!
interface Vlan150
ip address 20.20.20.1 255.255.255.0
ip pim sparse-dense-mode
ip igmp static-group 224.1.2.3 source 15.15.15.2
end
```

示例：配置设备使其在没有直连 IGMP 主机时转发组播流量

下面的示例显示了如何使用命令 **ip igmp join-group** 来配置设备，让它在没有直连 IGMP 主机的情况下转发组播流量。通过这种方式，设备可以接受组播数据包，并且转发组播数据包。接受组播数据包可以防止设备执行快速转发。

在这个示例中，用户对设备上的快速以太网接口 0/0/0 进行了配置，让它加入组 225.2.2.2：

```
interface FastEthernet0/0/0
ip igmp join-group 225.2.2.2
```

下面的示例显示了如何使用命令 **ip igmp static-group** 来配置设备，让它在没有直连 IGMP 主机的情况下转发组播流量。通过这种方式，设备本身不会接受组播数据包，但是会转发组播数据包。因此，这种方式允许设备执行快速转发。出站接口会添加到 IGMP 缓存中，但设备自身并不是组的成员，证据是组播路由条目中没有 L（本地）标记。

在这个示例中，用户在设备的快速以太网接口 0/1/0 上给组 255.2.2.2 配置了静态组成员：

```
interface FastEthernet0/1/0
ip igmp static-group 225.2.2.2
```

使用 IGMP 扩展访问列表来控制对 SSM 网络的访问

这一节包含了关于使用 IGMP 扩展访问列表来控制访问 SSM 网络的示例。

注释： 切记，访问列表是非常灵活的：用户可以在一个访问列表中设计出各式各样 permit 语句和 deny 语句的组合，达到过滤组播流量的效果。这一节中的示例仅提供其中的一些方法，来向读者展示操作方法。

示例：拒绝所有组 G 的状态

下面的示例显示了如何拒绝所有组 G 的状态。在这个示例中，用户对快速以太网接口 0/0/0 进行了配置，过滤了 IGMPv3 报告中所有源去往 SSM 组 223.2.2.2 的访问，这样可以有效地阻塞这个组。

```
ip access-list extended test1
deny igmp any host 232.2.2.2
permit igmp any any
!
interface FastEthernet0/0/0
ip igmp access-group test1
```

示例：拒绝所有源 S 的状态

下面的示例显示了如何拒绝所有源 S 的状态。在这个示例中，用户对吉比特以太网接口 1/1/0 进行了配置，过滤了 IGMPv3 报告中所有源为 10.2.1.32 的访问，这样可以有效地阻塞这个源。

```
ip access-list extended test2
deny igmp host 10.2.1.32 any
permit igmp any any
!
interface GigabitEthernet1/1/0
ip igmp access-group test2
```

示例：允许所有组 G 的状态

下面的示例显示了如何允许所有组 G 的状态。在这个示例中，用户对吉比特以太网接口 1/2/0 进行了配置，接受了 IGMPv3 报告中所有源去往 SSM 组 223.1.1.10 的访问，这样可以有效地接受这个组。

```
ip access-list extended test3
permit igmp any host 232.1.1.10
!
interface GigabitEthernet1/2/0
ip igmp access-group test3
```

示例：允许所有源 S 的状态

下面的示例显示了如何允许所有源 S 的状态。在这个示例中，用户对吉比特以太网接口 1/2 进行了配置，接受了 IGMPv3 报告中所有源为 10.6.23.32 的访问，这样可以有效地接受这个源。

```
ip access-list extended test4
permit igmp host 10.6.23.32 any
!
interface GigabitEthernet1/2/0
ip igmp access-group test4
```

示例：过滤源 S 对组 G 的访问

下面的示例显示了如何过滤特定源 S 访问组 G。在这个示例中，用户对吉比特以太网接口 0/3/0 进行了配置，让该接口在 IGMPv3 报告中过滤源 232.2.2.2 访问 SSM 组 232.2.30.30。

```
ip access-list extended test5
deny igmp host 10.4.4.4 host 232.2.30.30
permit igmp any any
!
interface GigabitEthernet0/3/0
ip igmp access-group test5
*Mar 1 00:48:28.640: %USB_CONSOLE-6-MEDIA_USB: Console media-type is USB.
```

其他参考资料

相关文档

相关主题	文档名
如需了解本章所述命令的完整语法结构及使用信息	《IP 组播路由命令参考手册（Inspur 6650 交换机）》
Inspur INOS 命令	《Inspur INOS 主命令列表，所有版本》
Inspur INOS IP SLA 命令	《Inspur INOS IP 组播命令参考手册》

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com

标准与 RFC

标准/RFC	标题
RFC 1112	IP 组播转发的主机扩展

RFC 2236	互联网组管理协议，第 2 版
RFC 3376	互联网组管理协议，第 3 版

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

配置接口特征的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

配置 IGMP 代理

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

IGMP 代理的前提条件

- IGMP UDL 上的所有设备都使用相同的子网地址。如果 UDL 上的所有设备无法使用相同的子网地址，那么用户就需要在上游路由器上配置辅助地址，来匹配所连各个下游路由器的全部子网；
- 需要启用 IP 组播并配置 PIM 接口；

注释： 用户在给 PIM 接口配置 IGMP 代理时，需要使用下面的指导方针

- 当这个接口工作在稀疏模式区域时，就应该使用稀疏模式（PIM-SM），并运行 RP、BSR 或包含 Auto-RP 侦听器功能的 Auto-RP；
- 当这个接口运行在稀疏-密集模式区域时，就应该使用 PIM 稀疏-密集模式，并运行不包含 Auto-RP 侦听器功能的 Auto-RP；
- 当这个接口运行在密集模式区域时，就应该使用 PIM 密集模式，并参与密集模式的区域；
- 如果这个接口正在从密集模式区域接收源流量，而这个密集模式区域又需要能够访问处于稀疏模式区域中的接收方，就应该配置包含代理注册功能的 PIM-DM。

关于 IGMP 代理的信息

IGMP 代理可以让不与下游路由器直连的 UDLR（单向链路路由）环境中的主机，加入一个源位于上游网络的组播组。

下图显示了一个双 UDLR 环境的示例拓扑：

- 传统 UDL 路由环境：包含直连接收方的 UDL 设备；
- IGMP 代理环境：不包含直连接收方的 UDL 设备。

注释： 上游和下游路由器上需要部署 IGMP UDL。

注释： 虽然下面的情况和示例在配置中使用的都是路由器，但在这里也可以使用交换机。

Internet Link	Internet 链路
Router A	路由器 A
Unidirectional Link	单向链路
Router B	路由器 B
LAN B	局域网 B
Local net	局域网络
Router C	路由器 C
User 2	用户 2
User 1	用户 1

环境 1——传统 UDLR 环境（UDL 设备有直连接收方）

在环境 1 中，不需要使用 IGMP 代理机制。在这个环境中，下面是事件发生的顺序：

1 用户 2 发送一条 IGMP 成员身份报告，请求组 G 中的消息；

2 路由器 B 接收到这条 IGMP 成员身份报告，针对局域网 B 添加了一条组 G 的转发条目，并将这条的 IGMP 报告代理发送给路由器 A，也就是 UDLR 的上游设备；

3 IGMP 报告穿越 Internet 执行代理转发。

4 路由器 A 接收到了 IGMP 代理消息，并在单向链路上维护一个转发条目。

环境 2——IGMP 代理环境（UDL 设备没有直连接收方）

在环境 2 中，需要使用 IGMP 代理机制，才能让那些不与下游设备直连的主机加入源在上游网络的组播组。在这个环境中，下面是事件发生的顺序：

1　用户 1 发送一条 IGMP 成员身份报告，请求组 G 中的消息；

2　路由器 C 逐跳向 RP（路由器 B）发送一条 PIM 加入消息；

3　路由器 B 接收到 PIM 加入消息，并针对局域网 B 添加了一条组 G 的转发条目；

4　路由器 B 周期性地校验自己的组播路由表，并且将 IGMP 成员身份报告通过 Internet 链路代理转发给其上游 UDL 设备；

5　路由器 A 在单向链路上创建并维护一个转发条目。

在企业网络中，人们常常需要通过卫星接收 IP 组播流量，并且将组播流量在整个网络中进行转发。单单依靠 UDLR（单向链路路由），环境 2 是无法实现的，因为接收方主机必须与下游路由器（也就是路由器 B）直连。IGMP 代理机制可以让路由器在组播转发表中创建(*,G)条目的 IGMP 报告，因而克服了这种限制。所以，用户要想让这个环境能够正常工作，就必须（使用命令 **ip igmp mroute-proxy**）对代理的(*,G)组播静态路由条目启用 IGMP 报告转发，并在连接（拥有潜在成员的）PIM 网络的接口上（使用命令 **ip igmp proxy-service**）启用组播路由代理服务。

注释：　由于 PIM 消息不会向上游转发，因此每个下游网络和上游网络都有一个独立的域。

如何配置 IGMP 代理

配置上游 UDL 设备来执行 IGMP UDLR

用户可以按照下面的步骤配置 UDL 设备，来实现 IGMP UDLR。

总步骤

1. enable

2. configure terminal

3. interface *type number*

4. ip igmp unidirectional-link

5. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *type number* 示例： `Device(config)# interface gigabitethernet 1/0/0`	进入接口配置模式。 • 对于 *type* 和 *number* 参数，指定上游设备上要用来充当 UDL 的接口

步骤 4	**ip igmp unidirectional-link** 示例： `Device(config-if)# ip igmp unidirectional-link`	在接口上配置 IGMP，让其成为执行 IGMP UDLR 的单向链路
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置下游 UDL 设备通过支持 IGMP 代理来执行 IGMP UDLR

用户可以按照下面的步骤下游 UDL 设备，通过支持 IGMP 代理来执行 IGMP UDLR。

总步骤

1. enable

2. configure terminal

3. interface *type number*

4. ip igmp unidirectional-link

5. exit

6. interface *type number*

7. ip igmp mroute-proxy *type number*

8. exit

9. interface *type number*

10. ip igmp helper-address udl *interface-type interface-number*

11. ip igmp proxy-service

12. end

13. show ip igmp interface

14. show ip igmp udlr

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *type number* 示例： `Device(config)# interface gigabitethernet 0/0/0`	进入接口配置模式。 • 对于 *type* 和 *number* 参数，在要执行 IGMP UDLR 的下游设备上指定充当 UDL 的接口

步骤 4	**ip igmp unidirectional-link** 示例： `Device(config-if)# ip igmp unidirectional-link`	在接口上配置 IGMP，让其成为执行 IGMP UDLR 的单向链路
步骤 5	**exit** 示例： `Device(config-if)# exit`	离开接口配置模式并返回全局配置模式
步骤 6	**interface** *type number* 示例： `Device(config)# interface gigabitethernet 1/0/0`	进入接口配置模式。 • 对于 *type* 和 *number* 参数，选择一个面向这些非直连主机的接口
步骤 7	**ip igmp mroute-proxy** *type number* 示例： `Device(config-if)# ip igmp mroute-proxy loopback 0`	对代理的(*,G)组播静态路由（mroute）条目启用 IGMP 报告转发。 • 执行这一步的目的，是针对所有组播转发表中的(*,G)条目，启用向代理服务接口转发 IGMP 报告； • 在这个示例中，用户在吉比特以太网接口 1/0/0 上配置了命令 **ip igmp mroute-proxy**，请求针对所有组，都将组播路由表中所有要转发给吉比特以太网接口 1/0/0 的 IGMP 报告，发送给 loopback 0 接口
步骤 8	**exit** 示例： `Device(config-if)# exit`	离开接口配置模式并返回全局配置模式
步骤 9	**interface** *type number* 示例： `Device(config)# interface loopback 0`	进入特定接口的接口配置模式。 • 在本例中，我们指定了 loopback 0 接口
步骤 10	**ip igmp helper-address udl** *interface-type interface-number* 示例： `Device(config-if)# ip igmp helper-address udl gigabitethernet 0/0/0`	针对 UDLR 配置 IGMP 帮助。 • 执行这一步的目的，是让下游设备将从主机那里接收到的 IGMP 报告，找到与（使用 *interface-type* 和 *interface-number* 参数指定的那个）UDL 相连的上游设备； • 在示例拓扑中，用户在下游设备的 loopback 0 接口上配置了 IGMP helper。用户通过配置 Loopback 0 接口来将主机发送的 IGMP 报告发送给吉比特以太网接口 0/0/0 所连接的上游设备
步骤	**ip igmp proxy-service**	启用组播路由代理服务。

11	示例： `Device(config-if)# ip igmp proxy-service`	• 当用户启用了组播代理服务时，设备会根据 IGMP 查询时间间隔周期性地查看静态组播路由表，校验与（步骤 7 中的）命令 **ip igmp mroute-proxy** 所配置的接口相匹配的(*,G)转发条目。如果出现匹配，就会创建一条 IGMP 报告，这个接口也会收到这条报告； **注释：**命令 **ip igmp proxy-service** 需要与命令 **ip igmp helper-address** 一起使用。 • 在本例中，用户在 loopback 0 接口上配置了 **ip igmp proxy-service** 命令，其目的是针对所有通过命令 **ip igmp mroute-proxy** 注册在接口上的组，都将发往它们的组播流量，从这个环回接口转发出去
步 骤 12	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步 骤 13	**show ip igmp interface** 示例： `Device# show ip igmp interface`	（可选）查看一个接口上与组播相关的信息
步 骤 14	**show ip igmp udlr** 示例： `Device# show ip igmp udlr`	（可选）查看配置了 UDL 帮助地址（helper address）的接口上，直连组播组的 UDLR 信息

IGMP 代理的配置示例

示例：IGMP 代理的配置

下面的示例显示了如何配置上游 UDL 设备来执行 IGMP UDLR，以及如何配置下游 UDL 设备通过支持 IGMP 代理来执行 IGMP UDLR。

上游设备的配置

```
interface gigabitethernet 0/0/0
ip address 10.1.1.1 255.255.255.0
ip pim dense-mode
!
interface gigabitethernet 1/0/0
ip address 10.2.1.1 255.255.255.0
ip pim dense-mode
ip igmp unidirectional-link
!
```

```
interface gigabitethernet 2/0/0
ip address 10.3.1.1 255.255.255.0
```

下游设备的配置

```
ip pim rp-address 10.5.1.1 5
access-list 5 permit 239.0.0.0 0.255.255.255
!
interface loopback 0
ip address 10.7.1.1 255.255.255.0
ip pim dense-mode
ip igmp helper-address udl ethernet 0
ip igmp proxy-service
!
interface gigabitethernet 0/0/0
ip address 10.2.1.2 255.255.255.0
ip pim dense-mode
ip igmp unidirectional-link
!
interface gigabitethernet 1/0/0
ip address 10.5.1.1 255.255.255.0
ip pim sparse-mode
ip igmp mroute-proxy loopback 0
!
interface gigabitethernet 2/0/0
ip address 10.6.1.1 255.255.255.0
```

其他参考资料

下面的内容提供了与自定义 IGMP 相关的参考资料。

相关文档

相关主题	文档名
Inspur INOS 命令	《Inspur INOS 主命令列表，所有版本》
Inspur INOS IP SLA 命令	《Inspur INOS IP 组播命令参考手册》
IP 组播技术相关知识的概述	“IP 组播技术概述”部分文档
IP 组播的基本概念、配置任务与示例	“配置基本 IP 组播”或“在 IPv6 网络中配置 IP 组播”部分文档

标准与 RFC

标准/RFC	标题
RFC 1112	IP 组播转发的主机扩展
RFC 2236	互联网组管理协议，第 2 版
RFC 3376	互联网组管理协议，第 3 版

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用	http://www.icntnetworks.com

户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	

IGMP 代理的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

在交换型以太网中限制 IP 组播

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

在交换型以太网中限制 IP 组播的前提条件

在参考这部分文档进行配置之前，用户应该首先熟悉“IP 组播技术概述”那部分文中介绍的概念。

如何在交换型以太网中限制 IP 组播

二层交换机的默认操作，是将所有组播流量从交换机目的局域网中的所有端口转发出去。这种行为会降低交换机的效率，而交换机应该对流量进行限制，只从需要接收这些数据的端口转发流量。这就要求交换机上有一种限制机制，能够减少不必要的组播流量，提升交换机的性能。

Inspur 组管理协议（CGMP），路由器组管理协议（RGMP）和 IGMP snooping 可以在二层交换环境中有效地限制 IP 组播。

CGMP 和 IGMP snooping 应该应用于那些包含终端用户或者接收方客户端的子网当中；

RGMP 应该用于那些只包含路由器的路由网段，如一个折叠式骨干网中；

RGMP 和 CGMP 无法实现互操作。但互联网组管理协议（IGMP）可以与 CGMP 和 RGMP snooping 实现互操作。

在 Inspur 交换机针对 IP 组播部署 CGMP

CGMP 是一项由 Inspur 开发的协议，这项协议主要部署在与 Inspur 交换机相连的设备上，来执行类似于 IGMP 的任务。对于那些无法区分 IP 组播数据包和 IGMP 报告消息的 Inspur 交换机来说，使用 CGMP 是十分必要的，因为这两种数据包在 MAC 层面都会编址相同的组地址。交换机可以区分 IGMP 数据包，但需要使用交换机上的软件，而这样会严重影响交换机的性能。

用户必须在组播设备和二层交换机上配置 CGMP。这样做的结果是，通过 CGMP，组播流量就只会通过那些连接了感兴趣接收方的交换机端口转发出去。所有其他那些没有显式请求流量的端口都不会接收到这些组播数据，除非这些端口连接了一台组播路由器。组播路由器端口必须接收到每个 IP 组播数据包。

如果使用了 CGMP，那么当主机加入一个组播组的时候，它就会用组播的形式主动将 IGMP 成员身份报告发给目标组。IGMP 报告会通过交换机转发给路由器，执行正常的 IGMP 处理。路由器（路由器上必须在这个接口上启用 CGMP）接收到 IGMP 报告之后，会按照常规的方式对其进行处理，但同时还会创建一条 CGMP 加入消息，并将其发送给交换机。加入消息中会包含终端工作站的 MAC 地址，以及所加入组的 MAC 地址。

交换机接收到这个 CGMP 加入消息时，会将这个端口添加到该组播组的 CAM（内容可编址内存）表中。所有此后去往这个组的流量此后都会通过这个端口来转发给主机。

二层交换机的设计方式是，同一个物理端口可以对应多个目的 MAC 地址。这种设计方式可以让交换机连接到一个分层网络，让多个组播目的地址可以通过一个端口转发出去。

设备端口也会被添加到这个组播组的条目中。组播设备必须侦听每个组中的所有组播流量，因为 IGMP 控制消息也会使用组播流量进行发送。其余组播流量则会使用 CGMP 在 CAM 表中添加的新增条目来进行转发。

IGMP Snooping

IGMP snooping 是一种运行在二层局域网交换机上的 IP 组播限制机制。IGMP snooping 需要局域网交换机对主机与路由器之间相互发送的 IGMP 数据包中，一些三层信息（IGMP 加入/离开消息）进行分析（或者说“窥探”）。当交换机从一台主机那里接收到一个特定组播组的

IGMP 主机报告时，交换机会将主机的端口号添加到相关的组播表条目中。当交换机从一台主机那里接收到 IGMP 离开组消息时，交换机会移除这台主机对应的条目。
由于 IGMP 控制消息也会采用组播数据包的方式进行发送，所以它们无法在二层与组播数据进行区分。运行 IGMP snooping 的交换机会检查每个组播数据包，以判断其中是否包含相关的 IGMP 控制信息。如果用户在 CPU 速率较低的低端交换机上实施了 IGMP snooping，那么当数据速率交稿时，这台交换机的性能有可能会受到严重影响。这个问题的解决方案是在那些配备了能够用硬件执行 IGMP 校验的 ASIC（专用集成电路）的高端交换机上实施 IGMP snooping。对于没有配备特殊硬件的低端交互来说，实施 CGMP 是更理想的选择。

路由器端口组管理协议（RGMP）

CGMP 和 IGMP snooping 都是 IP 组播限制机制，这类机制应该在连接了活动接收方的路由网络中实施。这两种协议都依赖主机和路由器之间发送的 IGMP 控制消息来判断那些交换机端口与感兴趣接收方相连。
交换型以太网骨干网络往往是由很多台连接到同一台交换机的路由器所组成，这种网络中往往没有主机。鉴于路由器不会生成 IGMP 主机报告，因此 CGMP 和 IGMP snooping 也就无法限制组播流量，这些流量会被通过 VLAN 中的每个端口进行泛洪。但路由器会创建 PIM（协议独立组播）消息在三层加入和修剪组播流量。
路由器端口组管理协议（RGMP）是一种用于纯路由器网段的 IP 组播限制机制。RGMP 必须在路由器和二层交换机上启用。组播路由器会通过向组发送 RGMP 加入消息的方式，标识自己对接受该组的数据流感兴趣。接下来，交换机会将对应的端口添加到这个组的转发表中，这种方式与交换机处理 CGMP 加入消息的方式类似。IP 组播数据只会转发给感兴趣的路由器端口。当路由器已经不对这些数据流感兴趣时，路由器会发送一条 RGMP 离开消息，而交换机也会移除相应的转发条目。
如果任何路由器没有启用 RGMP，它们都会继续接收所有组播数据。

如何在交换型以太网中限制 IP 组播

配置交换机来执行 IP 组播

如果组播网络中部署了交换机，应该查询相应的交换机文档来了解配置 IP 组播的信息。

配置 IGMP Snooping

路由器上不需要执行任何配置。用户应该查询相应的交换机文档来了解如何在交换机上启用 IGMP snooping，并且按照文档中提供的建议执行操作。

启用 CGMP

CGMP 协议用于与设备直连的 Inspur 交换机上，执行类似于 IGMP 的操作。对于那些无法区分 IP 组播数据包和 IGMP 报告消息的 Inspur 交换机来说，使用 CGMP 是十分必要的，因为

这两种数据包在 MAC 层面都会编址相同的组地址。交换机可以区分 IGMP 数据包。

注释： CGMP 只应该在 802、ATM 媒介，抑或 LANE over ATM 上启用。

CGMP 只应该在与 Inspur 交换机直连的设备上启用。

总步骤

1. enable

2. configure terminal

3. interface *type number*

4. ip cgmp [**proxy** | **router-only**]

5. end

6. clear ip cgmp [*interface-type interface-number*]

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *type number* 示例： `Device(config)# interface ethernet 1`	选择与可以启用IGMPv3的主机直连的接口
步骤 4	**ip cgmp** [**proxy** | **router-only**] 示例： `Device(config-if)# ip cgmp proxy`	在与 Inspur 5000 系列交换机相连的设备接口上启用 CGMP。 使用关键字 proxy 可以启用 CGMP 代理功能。在启用之后，所有不具有 CGMP 功能的设备都会由代理路由器进行通告。代理路由器会针对网络中存在其他不具备 CGMP 功能的设备发送通告消息。在消息中，代理路由器会将这些设备的 MAC 地址和组地址设置为 0000.0000.0000
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**clear ip cgmp** [*interface-type interface-number*] 示例： `Device# clear ip cgmp`	（可选）从 Inspur 交换机的缓存中清楚所有组条目

在二层交换型以太网中配置 IP 组播

用户可以按照下面的步骤使用 RGMP 在二层交换型以太网中配置 IP 组播

总步骤

1. enable

2. configure terminal

3. interface *type number*

4. ip rgmp

5. end

6. debug ip rgmp

7. show ip igmp interface

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *type number* 示例： `Device(config)# interface ethernet 1`	选择与主机相连的接口
步骤 4	**ip rgmp** 示例： `Device(config-if)# ip rgmp`	在以太网、快速以太网和吉比特以太网接口上启用 RGMP
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**debug ip rgmp** 示例： `Device# debug ip rgmp`	（可选）记录由 RGMP 设备发送的调试消息
步骤 7	**show ip igmp interface** 示例： `Device# show ip igmp interface`	（可选）显示一个接口上与组播相关的信息

在交换型以太网中限制 IP 组播的配置示例

示例：配置 CGMP

下面的案例适用于组播源和组播接收方处于同一个 VLAN 中的这种基本网络环境。用户的需求是交换机限制组播转发，让组播只通过那些请求了组播流量的端口进行转发。
在网络中，一台 4908-L3 路由器连接到了 Inspur 4003 设备的 3/1 端口，该端口属于 VLAN 50。下面的配置应用在了 GigabitEthernet1 接口上。注意，用户没有配置 **ip multicast-routing** 这条命令，因为路由器不会通过自己的接口来路由组播流量。

```
interface GigabitEthernet1
ip address 192.168.50.11 255.255.255.0
ip pim dense-mode
ip cgmp
```

示例：配置 RGMP

下面的示例显示了如何在路由器上配置 RGMP：

```
ip multicast-routing
ip pim sparse-mode
interface ethernet 0
ip rgmp
```

其他参考资料

下面的内容提供了与在交换型以太网中限制 IP 组播的相关参考资料。

相关文档

相关主题	文档名
Inspur INOS 命令	《Inspur INOS 主命令列表，所有版本》
Inspur INOS IP SLA 命令	《Inspur INOS IP 组播命令参考手册》
IGMP snooping	《IP 组播：IGMP 配置指南》中的 IGMP snooping 部分
RGMP	《IP 组播：IGMP 配置指南》中的配置路由器端口组管理协议部分

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产	http://www.icntnetworks.com

品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	

在交换型以太网中限制 IP 组播的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

配置 PIM

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

PIM 的前提条件

在开始配置 PIM 之前，要确定准备使用的 PIM 模式。PIM 模式与用户希望网络支持的应用有关。具体的知道方针为：

- 总地来说，如果应用本质上是一对多或多对多应用，那就可以成功使用 PIM-SM 模式；
- 如果希望让一对多的应用实现最优的性能，最适合的模式是 SSM，但需要设备支持 IGMP 第 3 版。

在配置 PIM 末节路由之前，要判断是否满足下列条件：

- 用户必须在末节路由器和中心路由器上都配置了 IP 组播路由。此外，用户也必须在末

节路由器的上行接口上配置 PIM 模式（密集模式、稀疏模式或稀疏-密集模式皆可）；

- 用户还必须在设备上配置 EIGRP（高级内部网关路由协议）末节路由或者 OSPF（最短路径优先）末节路由；
- PIM 末节路由器不会对在分布层路由器之间穿越的流量进行路由。单播（EIGRP）末节路由会强制执行这项操作。用户必须配置单播末节路由来帮助 PIM 末节路由器来执行这项操作。

注释： 要想进一步了解关于 EIGRP 或 OSPF 配置的信息，可以参阅《Inspur 6650 路由配置指南，3E 版》。

PIM 的限制条件

下面这 1 条是配置 PIM 的限制条件：

- 如果设备运行的是 LAN Base 特性集，那么这台设备上就不支持 PIM。

PIMv1 与 PIMv2 的互操作性

要避免在设备上错误配置组播路由，可以阅读这一节中内容。

Inspur 可以提供 PIM 第 1 版和第 2 版之间的互操作性和过渡，虽然这里会有一些小小的问题。

用户可以渐进性地升级到 PIMv2。在同一个网络中，用户可以在不同的路由器和多次交换机上配置 PIMv1 和 PIMv2。在共享介质网络内部，所有路由器和多层交换机上都必须运行同一个版本的 PIM 协议。因此，如果 PIMv2 设备检测到了一台 PIMv1 设备，那么 PIMv2 设备会就将自己运行的级别降至 PIMv1，直到所有的第 1 版设备都关闭或升级到第 2 版为止。

PIMv2 使用 BSR 发现每个组前缀的 RP 集信息，并将信息通告给 PIM 域中的所有路由器和多层交换机。PIMv1 如果与 Auto-RP 特性一起使用，则可以执行和 PIMv2 BSR 相同的任务。但 Auto-RP 是一个单独的协议，与 PIMv1 是相互独立的，它是 Inspur 私有的协议。PIMv2 则是 IETF 的标准协议。

注释： 我们推荐用户使用 PIMv2。BRS 可以与 Inspur 路由器和多层设备上的 Auto-RP 实现互操作。

当 PIMv2 设备与 PIMv1 设备互操作时，Auto-RP 应该已经部署好了。PIMv2 BSR 同时也会充当 Auto-RP 映射代理，这台设备会自动将 Auto-RP 选举出来的 RP 通告出去。也就是说，Auto-RP 会在这个组中的每台路由器或多层设备上设置其自己的单个 RP。并不是域中所有路由器和设备都会使用 PIMv2 散列功能来选择多 RP。

在 PIMv1 和 PIMv2 混合的网络域中，密集模式组并不需要专门进行什么配置；它们可以自动实现互操作。

在 PIMv1 和 PIMv2 混合的网络域中，也有可能部署稀疏模式组，因为 PIMv1 中的 Auto-RP 特性可以与 PIMv2 的 RP 特性进行互操作。虽然所有 PIMv2 设备也可以使用 PIMv1，但我们推荐将 RP 升级到 PIMv2。如果希望轻松过渡到 PIMv2，我们推荐：

- 在整个网络域中使用 Auto-RP；
- 在整个网络域中配置稀疏-密集模式。

如果 PIMv1 域中还没有配置好 Auto-RP，请先配置 Auto-RP。

配置 PIM 末节路由的限制条件

- IP Services 镜像包含了对组播路由的完整支持；
- 二层接入域中只允许部署直连的组播（IGMP）接收方和组播源。接入域中不支持 PIM 协议；
- 在使用 PIM 末节路由的网络中，唯一允许向用户路由 IP 流量的方式，是让流量穿过一台配置了 PIM 末节路由的设备；
- 不支持冗余的 PIM 末节路由器拓扑。PIM 末节特性只支持非冗余的接入路由器拓扑；
- 运行 IP Base 和 IP Services 特性集的设备都支持 PIM 末节路由。

配置 Auto-RP 和 BSR 的限制条件

考虑到网络的配置，配置 Auto-RP 和 BSR 存在下面的限制条件：

配置 Auto-RP 的限制条件

- 运行 LAN Base 特性集的设备不支持 Auto-RP；
- 如果配置稀疏模式或稀疏-密集模式的 PIM，但没有配置 Auto-RP，那么用户必须手动配置一台 RP；
- 如果用户将一些路由接口配置在稀疏模式下，同时用户在所有设备上都给 Auto-RP 组手动配置了一个 RP 地址，那么用户仍然可以使用 Auto-RP。
- 如果用户将一些路由接口配置在稀疏模式下，同时又输入了全局配置命令 **ip pim autorp listener**，那么即使所有设备上用户都没有给 Auto-RP 组手动配置 RP 地址，仍然可以使用 Auto-RP。

配置 BSR 的限制条件

下面是（在配置自己的网络时）配置 BSR 的限制条件：

- 将候选 BSR 配置为 Auto-RP 的 RP 映射代理；
- 对于通过 Auto-RP 通告的组前缀，PIMv2 BSR 机制不应该通告这组前缀范围的子集。在 PIMv1 和 PIMv2 混合的网络域中，同样的组前缀要部署备份 RP。这可以防止 PIMv2 DR 根据 RP 映射数据库的最长匹配查找原则，而从那些 PIMv1 DR 中选择出一个不同的 RP。

配置 Auto-RP 和 BSR 的限制条件与指导方针

下面是（在配置自己的网络时）配置 Auto-RP 和 BSR 的限制条件：

- 如果这个网络是全部由 Inspur 路由器和多层设备组成的，那么用户可以使用 Auto-RP 或者 BSR；
- 如果网络中包含了非 Inspur 的路由器，那就必须使用 BSR；
- 如果网络中同时部署了 Inspur PIMv1 和 PIMv2 路由器与多层设备，以及非 Inspur 路由器，那用户就必须同时使用 Auto-RP 和 BSR。如果网络中包含了其他厂商的路由器，那就要在 Inspur PIMv2 设备上配置 Auto-RP 映射代理和 BSR。要确保 BSR 和非 Inspur PIMv2 设备之间的路径上没有 PIMv1 设备；

注释： 使用 PIMv2 有两种方法。可以在网络中仅使用 PIMv2，也可以将部署了混合 PIM 版本的环境迁移为纯 PIMv2 的环境。

- 因为自举（bootstrap）消息是逐跳发送的，所以 PIMv1 可以防止这些消息到达网络中的所有路由器和多层设备。因此，如果网络中有一台 PIMv1 设备，同时网络中又只有 Inspur 路由器和多层设备，那就最好使用 Auto-RP；
- 如果网络中包含了非 Inspur 设备，应该在 Inspur PIMv2 路由器或多层设备上配置 Auto-

RP 映射代理和 BSR。确保 BSR 和非 Inspur PIMv2 设备之间的路径上没有 PIMv1 设备；

- 如果网络中有非 Inspur 的 PIMv2 路由器需要与 PIMv1 路由器和多层设备进行互操作，那就需要同时配置 Auto-RP 和 BSR。我们推荐用一台 Inspur PIMv2 设备来同时作为 Auto-RP 映射代理和 BSR。

Auto-RP 增强的限制条件

不支持同时部署 Auto-RP 和自举路由器（BSR）。

关于 PIM 的信息

协议独立组播概述

协议独立组播（PIM）协议会维护当前接收方发起的成员关系的 IP 组播服务模式。PIM 并不依赖于某一种特定的单播路由协议；这是一项独立于 IP 路由协议的协议，但可以利用网络中使用的单播路由协议来创建单播路由表，可以利用的单播路由协议包括 EIGRP（增强型内部网关路由协议）、OSPF（最短路径优先）、BGP（边界网关协议）和静态路由。PIM 会使用单播路由协议来执行组播转发功能。

虽然 PIM 称为组播路由协议，但它其实是使用单播路由表来执行逆向路径转发（RPF）校验功能的，并不会建立一个完全独立的组播路由表。PIM 与其他路由协议不同，它不会发送和接收路由器之间的路由更新。

PIM 定义在 RFC 4601 中：RFC 4601，协议独立组播——稀疏模式（PIM-SM）。

PIM 可以工作在密集模式或稀疏模式中。路由器也可以同时处理稀疏组和密集组（稀疏-密集模式）。模式决定了路由器会如何创建组播路由表，以及路由器如何转发它从直连 LAN 中接收到的组播数据包。

要想了解关于 PIM 转发（接口）模式的详细信息，可以阅读下面的内容：

PIM 密集模式

PIM 密集模式（PIM-DM）会使用推送模型将组播流量泛洪到网络中的每一个角落。这种推送模型是一种在接收方没有请求数据，即主动向接收方传输数据的方式。在有些部署环境（即网络中的每个子网都有有效接收方的环境）中，这种模式效率很高。

在密集模式中，路由器会认为所有其他路由器都希望向一个组转发组播数据包。如果一台路由器接收到了一个组播数据包，但它没有直连成员或 PIM 邻居，那么它就会向反方向发送一条修剪消息。这样后面的组播数据包就不会再泛洪给这台被修剪叶网络中的路由器了。

PIM 会建立基于源的组播分发树。

PIM-DM 起初会在整个网络中泛洪组播流量。没有下游邻居的路由器修剪掉那些自己不需要的流量。这个进程每 3 秒就会重复一次。

路由器会通过泛洪和修剪机制来收数据流，并以此汇集状态信息。这些数据流中包含源和组信息，这样下游路由器就可以建立自己的组播转发表了。PIM-DM 只支持源树，也即(S,G)条目，它不能用来建立共享分发树。

注释： 密集模式不常使用，我们也不推荐使用这种模式。因此，我们不会在文档中介绍它的配置方法。

PIM 稀疏模式

PIM 稀疏模式（PIM-SM）使用了推送模型来传输组播流量。只有包含活动接收方的那些显式请求了数据网段，才会接收到这些数据。

稀疏模式的接口与密集模式接口不同，只有当下游路由器接收到了周期性发送的加入消息，或者当接口上有直连的成员时，稀疏模式接口才会被添加到组播路由表中。在转发来自一个局域网的流量时，如果组知道 RP，那么设备就会执行稀疏模式的操作。在这种情况下，设备就会封装数据包，并且将数据包发送给 RP。如果不知道 RP，那么数据包就会按照密集模式的方式进行泛洪。如果来自于一个特定源的组播流量足够多，那么接收方的第一跳路由器有可能会向源发送加入消息，来建立基于源的分发树。

PIM-SM 会在共享树中转发数据包，以此来分发关于活动源的信息。因为 PIM-SM 会使用共享树（至少一开始会使用共享树），所以它需要使用汇集点（RP）。RP 必须由管理员在网络中进行配置。详见汇集点了解详细信息。

在稀疏模式下，路由器会认为其他路由器都不希望给一个组转发组播数据包，除非这台路由器显式请求了该流量。当主机加入一个组播组时，直连的路由器就会向 RP 发送 PIM 加入消息。RP 会对组播组进行追踪。发送组播数据包的主机会通过自己的第一跳路由器在 RP 上进行注册。接下啦，RP 会向源发送加入消息。此时，数据包就会在共享树中进行转发了。如果特定源发送的组播流量足够多，那么主机的第一跳路由器有可能会向源发送加入消息，来建立基于源的分发树。

源会在 RP 上进行注册，然后将数据沿着共享树向下转发给接收方。当边缘路由器通过 RP 接收到源在共享树中发送的数据包时，边缘路由器就会学习到这个源。接下来，边缘路由器就会向源发送 PIM (S,G)加入消息。每台逆向路径中的路由器都会将 RP 地址的单播路由度量值，与源地址的度量值进行比较。如果源地址的度量值更优，它就会向源转发一条 PIM (S,G)加入消息。如果 RP 的度量值相同或者更优，它就会向与 RP 相同的方向发送一条 PIM (S,G)加入消息。在这种情况下，可以认为共享树和源树是一致的。

如果共享树不是源和接收方之间的最优路径，路由器就会动态创建一个源树，并且停止沿着共享树发送流量。这是软件的默认操作方式。网络管理员可以通过命令 **ip pim spt-threshold infinity** 来强制流量保持在共享树中。

PIM-SM 可以很好地扩展到任意规模的网络中，也包含那些有 WAN 链路的网络。显式加入机制可以防止多余的流量被泛洪到 WAN 链路上。

组播源发现协议（MSDP）

组播源发现协议（MSDP）用于 PIM-SM 环境中的域间源发现。每个 PIM 管理域都有自己的 RP。为了让一个域中的 RP 将新的源发送给其他域中的 RP，人们才会使用 MSDP。

当一个域中的 RP 接收到了一条去往新源的 PIM 注册消息，那么如果配置了 MSDP，这个 RP 就会向另一个域中的所有 MSDP 对等体发送一条新的源-活动（SA）消息。每台中间 MSDP 对等体都会泛洪这个来自于 RP 的 SA 消息。MSDP 对等体都会将这条 SA 消息添加到自己的 MSDP sa 缓存中。如果其他域中的 RP 对 SA 消息中的组有（用一条出站接口列表非空的(*,G)条目标识的）加入请求消息，那么这个组就会对该域感兴趣，而 RP 也会向源发出一条(S,G)的加入消息。

稀疏-密集模式

如果用户在一个接口上配置了稀疏模式或者密集模式，那么接口就会工作在稀疏或密集状态下。但有些环境可能需要一个域中的 PIM 对一些组工作在稀疏模式下，而对另一些组工作在密集模式下。

另一种启用纯密集模式或纯稀疏模式的方法，是启用稀疏-密集模式。在这种环境中，如果组是密集模式，这个接口就会工作在密集模式下，如果组是稀疏模式，这个接口就会工作在稀疏模式下。如果接口工作在稀疏-密集模式下，而用户又希望将这个组视为稀疏组，那就

必须配置一台 RP。
如果用户配置了稀疏-密集模式，那么接口工作在或稀疏或密集状态下的方式就会针对路由器作为成员的组。
稀疏-密集模式的另一个好处在于，Auto-RP 信息可以在密集模式中进行分发；不过，用户组的组播组可以用于稀疏模式的方式。因此，用户没有必要在叶路由器上配置默认 RP。
当接口按照密集模式执行操作时，它就会出现在组播路由表的出站接口列表中，前提是下列两个条件之一为真：

- 接口连接了成员设备或 DVMRP 邻居；
- 接口仍有 PIM 邻居和组未被修剪。

当接口按照稀疏模式执行操作时，它就会出现在组播路由表的出站接口列表中，前提是下列两个条件之一为真：

- 接口连接了成员设备或 DVMRP 邻居；
- 接口的一台 PIM 邻居已经接收到了一条显式的加入消息。

PIM 版本
PIMv2 包含了下列针对 PIMv1 的优化：

- 每个组播组有一台活动的汇集点（RP），同时可以有多个备份 RP。在 PIMv1 中，这个 RP 会与同一组中的多台活动 RP 进行比较；
- 自举路由器）BSR）会提供容错、自动 RP 发现与分发功能，这种功能可以让路由器和多层设备动态学习到组-RP 的映射；
- 稀疏模式和密集模式成为组的属性，而不是接口的属性；

注释： 我们强烈推荐用户使用稀疏-密集模式，而不推荐使用纯稀疏模式或纯密集模式。

- 对于组地址族来说，PIM 加入和修剪消息的编码方式拥有更加灵活；
- 用更灵活的 hello 数据包格式取代了查询数据包，对当前及未来的功能选项提供了扩展空间；
- 发送到 RP 的注册消息会显示这是边界路由器发送的消息还是指定路由器发送的消息；
- PIM 数据包不再封装在 IGMP 数据包内部，它们成为了独立的数据包。

PIM 末节路由

所有设备的软件版本都支持 PIM 末节路由特性，这种特性可以将路由流量移动到距离终端用户更近的位置，以此降低对资源的消耗。
PIM 末节路由特性支持在分布层和接入层之间路由组播。它支持两类 PIM 接口，即上行链路 PIM 接口和 PIM 被动接口。配置了 PIM 被动模式的路由接口不会传输和转发 PIM 控制流量，这类接口只会传输和转发 IGMP 流量。
在使用 PIM 末节路由的网络中，在使用 PIM 末节路由的网络中，唯一允许向用户路由 IP 流量的方式，是让流量穿过一台配置了 PIM 末节路由的设备。PIM 被动接口会与二层接入域（如 VLAN），或者连接其他二层设备的接口相连。二层接入域中只允许直连的组播（IGMP）接收方和源。PIM 被动接口不会发送和处理任何接收到的 PIM 控制数据包。
在使用 PIM 末节路由时，用户应该配置分布层路由器和远程路由器来使用 IP 组播路由，同时只将这些设备配置为 PIM 末节路由器。设备不会对在分布层路由器之间穿越的流量进行路由。用户也需要在设备上配置一个路由模式的上行链路端口。设备的上行链路端口不能使用 SVI。如果希望针对 SVI 上行链路端口执行 PIM，需要将软件升级到 IP Services 特性集。

注释： 在设备上配置 PIM 末节路由时，用户还必须配置 EIGRP 末节路由。

PIM 末节路由器不支持冗余拓扑。当有多台 PIM 路由器向一个接入域转发组播流量时，这就

形成了一个冗余拓扑。PIM 消息会被阻塞，PIM 被动接口也不支持指定路由器选举机制。PIM 末节特性只支持不包含冗余接入路由器的拓扑。在这种不包含冗余设计的拓扑中，PIM 被动接口会认为它是接入域的唯一接口和指定路由器。
在下面这张图中，设备 A 路由模式的上行链路端口 25 与路由器相连，设备在 VLAN 100 接口和连接主机 3 的端口都启用了 PIM 末节路由。这种配置可以让直连的主机接收到来自组播源 200.1.1.3 的流量。

图 24：PIM 末节路由器配置

Source 200.1.1.3	源 200.1.1.3
Router	路由器
Switch A	交换机 A
Host 3	主机 3
Host 2	主机 2
Host 1	主机 1
Port 25	端口 25
Port 20	端口 20

IGMP Helper

PIM 末节路由会将被路由的流量移动到距离终端用户更近的位置，达到减少网络流量的目的。用户可以通过在末节路由器上配置 IGMP helper 特性的方式，来达到减少网络流量的目的。用户可以在末节路由器（交换机）上配置接口配置命令 **ip igmp helper-address** *ip-address*，让交换机向下一跳接口发送报告。这样一来，没有与下游路由器直连的那些主机就可以加入源位于上游网络的组播组了。在配置了这种特性之后，从一台希望加入组播组的主机发来的 IGMP 数据包就会向上游转发给下一跳设备。当上游中心路由器接收到 helper IGMP 报告或离开消息时，它就会在该组的出站接口中添加或移除这个接口。

汇集点

汇集点（RP）是设备执行 PIM SM（协议独立组播 稀疏模式）操作时的一种角色。只有运行 PIM SM 的网络才需要部署 RP。在 PIM-SM 模型中，只有当一个网段中包含了显式请求组播数据的接收方时，设备才会向这个网段转发流量。这种转发组播数据的方式与 PIM 密集模式（DM）相对。在 PIM DM 模式中，组播流量最初会泛洪到网络的所有网段。那些没有下游邻居的路由器，或者没有直连接收方的路由器，会向回修剪掉这些不需要的流量。
RP 会充当源和接收方组播流量的“见面地点”。在 PIM-SM 网络中，源必须将流量发送给 RP。接下来，这些流量会沿着共享分发树转发给接收方。在默认情况下，当接收方的第一跳设备学习到源时，它会直接向源发送一条加入消息，创建出一个从源到接收方的基于源的分发树。这棵源树中并不包含 RP，除非 RP 位于源与接收方之间的最短路径中。
对于大多数情况，RP 在网络中的位置如何部署并不需要经过复杂的考量。在默认情况下，只有源和接收方建立新的会话时，才需要借助 RP。因此，RP 并不会因为处理流量而受到太多影响。在 PIM 第 2 版中，RP 执行的处理比 PIM 第 1 版要少，因为源只能周期性地注册在 RP 上，以创建状态。

相关主题

Auto-RP

在 PIM-SM 的第 1 个版本中，所有叶路由器（即与源或接收方直连的路由器）上都需要手动配置 RP 的 IP 地址。这种配置也称为静态 RP 配置。在小型网络中，配置静态 RP 相对比较简单。但是在大型、复杂的网络环境中，这就会变成一项费时费力的工作。

下面我们来介绍 PIM-SM 第 1 版，Inspur 实施了一个包含 Auto-RP 特性的 PIM-SM 版本。Auto-RP 会自动在 PIM 网络中分发组与 RP 的映射关系。Auto-RP 可以带来下列好处：

- 在网络中可以通过配置多个 RP 来服务不同的组；
- Auto-RP 支持通过不同的 RP 来分担负载，也可以根据组参与方的位置来部署 RP；
- Auto-RP 可以避免不连续的、手动配置的 RP，这种做法有可能会导致链接性问题。

用户可以使用多个 RP 来服务不同的组范围，或者相互充当备份 RP。要想让 Auto-RP 正常工作，用户需要将路由器指定为 RP 映射代理，而 RP 映射代理会接收从 RP 那里发来的 RP 通告消息，并且执行冲突仲裁。接下来，RP 映射代理会向所有其他路由器发送连续的组-RP 映射。因此，所有路由器都可以自动发现哪个 RP 负责自己支持的组。

注释： 如果在稀疏模式或稀疏-密集模式下配置 PIM，但又没有配置 Auto-RP，那就必须由用户静态配置 RP。

注释： 如果用户将一些路由接口配置在稀疏模式下，同时用户在所有设备上都给 Auto-RP 组手动配置了一个 RP 地址，那么用户仍然可以使用 Auto-RP。

要想让 Auto-RP 正常工作，用户需要将路由器指定为 RP 映射代理，而 RP 映射代理会接收从 RP 那里发来的 RP 通告消息，并且执行冲突仲裁。接下来，RP 映射代理会通过密集模式泛洪的方法，向所有其他路由器发送连续的组-RP 映射。因此，所有路由器都可以自动发现哪个 RP 负责自己支持的组。IANA（互联网数字分配机构）给 Auto-RP 分配了 2 个组地址，224.0.1.39 和 224.0.1.40。Auto-RP 的一大优势在于，如果用户要对指定的 RP 进行任何变更，都可以只配置在 RP 路由器上，而不必配置在叶路由器上。Auto-RP 的另一个优势是，Auto-RP 可以让 RP 地址覆盖整个网络域。用户可以通过定义 Auto-RP 通告的生存时间值（TTL）来扩展 RP 的作用范围。

配置 RP 的每种方式都有自己的优势与劣势，复杂程度也各有不同。在传统的 IP 组播网络环境中，我们推荐用户使用 Auto-RP 来配置 RP，因为这种方法配置和测试起来都比较容易，而且也更加稳定。配置 RP 的方法包括静态 RP、Auto-RP 和自举路由器。

PIM 网络中 Auto-RP 的角色

Auto-RP 可以在 PIM 网络中，自动分发组-RP 的映射关系。要想让 Auto-RP 正常工作，用户需要将路由器指定为 RP 映射代理，而 RP 映射代理会接收从 RP 那里发来的 RP 通告消息，并且执行冲突仲裁。接下来，RP 映射代理会通过密集模式泛洪的方法，向所有其他路由器发送连续的组-RP 映射。

因此，所有路由器都可以自动发现哪个 RP 负责自己支持的组。IANA（互联网数字分配机构）给 Auto-RP 分配了 2 个组地址，224.0.1.39 和 224.0.1.40。

映射代理会接收到从候选 RP 变为 RP 的意向通告。接下来，映射代理会通告 RP 选举的获胜方。这个通告是由其他映射代理独立判断出来的。

组播边界

管理的作用范围边界可以用来限制组播流量，让它们不会被转发到网络域或者子域之外。这种方法会使用一种特殊的组播地址范围，称为管理作用范围地址，来充当边界划分机制。如果用户在路由接口上配置了一个管理作用范围边界，那么组播地址在这个范围之内的组播流量不无法进入或离开这个接口了，这相当于给这个地址范围的组播流量提供一个防火墙。

注释： 组播边界和 TTL 门限会控制组播域的作用范围；但设备不支持 TTL 门限。用户应该使用组播边界来代替 TTL 门限值，来限制组播流量的转发，让其不会流入到网络域或子域之外。

下面显示，XYZ 公司在网络边界的所有路由接口上，都针对组播地址范围 239.0.0.0/8 设置了一个管理作用范围的边界集。这个边界可以防止范围在 239.0.0.0-239.255.255.255 之内的组播流量流入或流出这个网络。同样，工程部和市场部在网络边界上针对地址范围 239.128.0.0/16 部署了一个管理作用范围的边界。这个边界可以防止范围在 239.128.0.0-239.128.255.255 之内的组播流量流入或流出它们响应的网络。

图 25：管理作用范围的边界

Company XYZ	XYZ 公司
Engineering	工程部
Marketting	市场部

用户可以在路由接口上给组播组地址定义一个管理作用范围的边界。标准访问列表可以定义受影响的地址范围。在定义了边界之后，没有组播数据包可以穿越边界，无论是从哪个方向上穿越都不可以。边界可以让同一个组播组地址在不同的管理域中复用。

IANA 将组播地址范围 239.0.0.0 到 239.255.255.255 指定为了管理作用范围的地址。这个地址范围可以在不同机构管理的域中进行复用。这些地址可以认为是本地地址，而不是全局唯一的。

用户可以在管理作用范围边界配置关键字 **filter-autorp**，来检查并过滤 Auto-RP 发现和通告消息。对于所有被边界访问控制列表（ACL）拒绝的 Auto-RP 数据包，其中的 Auto-RP 组范围通告都会被移除。如果 ACL 允许 Auto-RP 组范围中的所有地址，那么 Auto-RP 组范围通过也会被允许，并且可以穿越边界。如果 ACL 不允许其中的任何地址，那么整个组范围都会被过滤，并且从 Auto-RP 消息中被移除，然后设备才会转发 Auto-RP 消息。

稀疏-密集模式中的 Auto-RP

Auto-RP 的前提条件是所有接口都必须使用接口配置命令 **ip pim sparse-dense-mode** 进行配置，让它工作在稀疏-密集模式下。配置在稀疏-密集模式下的接口会执行稀疏模式的操作，或者执行密集模式的操作，具体的操作方式取决于组播组工作的模式。如果组播组有一个已知的 RP，那么这个接口就会工作在稀疏模式下。如果组没有已知的 RP，那么接口默认会工作在密集模式下，数据也会通过这个接口进行泛洪。（用户可以防止密集模式回退；详见“配置基本 IP 组播”）

要想成功实施 Auto-RP，并且防止 224.0.1.39 和 224.0.1.40 之外的组工作在密集模式下，我们推荐配置一个“通吃 RP（sink RP）”（也称为最终 RP[RP of last resort]）。所谓通吃 RP 是一台静态配置的 RP，网络中有可能真地存在这个 RP，也有可能并不存在这个 RP。配置通吃 RP 并不会影响 Auto-RP 的操作，因为 Auto-RP 消息默认优于静态 RP 的配置。我们推荐给网络中的所有组播组都配置一个通吃 RP，因为未知或意料之外的源也有可能会成为活动源。如果用户没有配置 RP 来限制源的注册，那么这个组有可能会回退到密集模式的操作，并且开始泛洪数据。

Auto-RP 的优势

Auto-RP 使用 IP 组播来自动向 PIM 网络中的所有 Inspur 路由器和多层设备分发组-RP 的映射关系。Auto-RP 的优势包括：

- 在网络中可以通过配置多个 RP 来服务不同的组；
- Auto-RP 支持通过不同的 RP 来分担负载，也可以根据组参与方的位置来部署 RP；
- Auto-RP 可以避免不连续的、手动配置的 RP，这种做法有可能会导致链接性问题。

Auto-RP 在 PIM 网络中的优势

- Auto-RP 可以让用户在对指定 RP 进行任何变更时，都只在 RP 路由器上进行配置，而不需要在叶路由器上进行配置；
- Auto-RP 可以在域中扩展 RP 地址的作用范围。

PIMv2 自举路由器

PIMv2 自举路由器（BSR）是另一种向网络中所有 PIM 路由器和多层设备分发组-RP 映射信息的方式。它可以让用户不必再在网络中的每台路由器和设备上手动配置 RP 路由器。但是，除了使用 IP 组播分发组-RP 映射信息之外，BSR 会使用逐跳泛洪特殊 BSR 消息的方式，来分发映射信息。

BSR 是从一系列候选路由器和设备中选举出来的，这些设备上都由用户配置了 BSR 功能。选举机制与交换型局域网中的根网桥选举机制类似。BSR 选举会基于设备的 BSR 优先级来作为判断标准，而 BSR 优先级则包含在网络中逐跳发送的 BSR 消息中。每台 BSR 设备都会查看消息，但它只会将 BSR 优先级大于等于自己，同时 BSR IP 地址更高的消息从所有接口转发出去。BSR 就是通过这种方法选举出来的。

选举的 BSR 会用 TTL 值 1 发送 BSR 消息。相邻的 PIMv2 路由器或多层设备会接收到这个 BSR 消息，并且以 TTL 值 1 将它从所有接口转发出去（除了接收到该消息的那个接口）。通过这种方式，BSR 消息就可以逐跳在 PIM 域中进行发送了。由于 BSR 消息中包含当前 BSR 的 IP 地址，因此泛洪机制可以让候选 RP 自动学习到哪台设备是选举出来的 BSR。

候选 RP 会发送候选 RP 通告消息，其中显示了 BSR 负责的组范围，而 BSR 会将这个信息保存在自己的本地候选 RP 缓存当中。BSR 会周期性地在 BSR 消息中向域中其他 PIM 设备通告这个缓存的内容。这些消息会通过网络逐跳发送给所有路由器和设备，这些设备也会将 BSR 消息中的 RP 信息保存在自己的本地 RP 缓存当中。路由器和设备在每个组都会选举出相同的 RP，因为它们使用的 RP 散列算法都是相同的。

PIM 域边界

随着 IP 组播使用更加广泛，在一个 PIMv2 域与另一个 PIMv2 域之间划分边界的机会也就增加了。因为两个域很可能不会共享同一组 RP、BSR、候选 RP 和候选 BSR，因此用户需要限制 PIMv2 BSR 消息，防止它流入流出这个域。让消息留出域边界会给正常的 BSR 选举机制带来负面影响，导致设备跨越边界选举出一台 BSR，也会让候选 RP 通告混合在一起，这会导致设备在错误的域中选举 RP。

组播转发

转发组播流量需要通过启用了组播功能的路由器来完成。这些路由器会创建分发树来控制 IP 组播流量在网络中发送的路径，以便将流量转发给所有的接收方。

组播流量会通过转发树由源发送给组播组，因为转发树会连接组中的所有源和所有接收方。这棵树可能是由所有源共享的（即共享树），也可能是每个源独立建立的分发树（即源树）。共享树既有可能是单向的，也有可能是双向的。

在描述源树和共享树的结构之前，我们不妨解释一下组播路由表中的一些标记。这些标记包括：

- (S,G)：（组播组 G 的单播源，组播组 G）
- (*,G)：（组播组 G 的任意源，组播组 G）

(S,G)标记读作“S 逗号 G”[①]，这种树是在列举所有以 S 作为源的 IP 地址，以 G 作为组播组地址的最短路径树。

① 英文中确实读作“S 逗号 G”，但汉语中其实多读作“S 点 G”。——译者注

共享树(*,G)和源树(S,G)都是在源进行路由的。

组播分发源树

源数是最简单的组播分发树。源树的根在源主机，其枝干则组成了跨越网络最终到达接收方的生成树。由于这棵树使用的是网络中的最短路径，因此这棵树也称为最短路径树（SPT）。

下图显示了组 224.1.1.1 从源主机 A 扩展出来的 SPT 示例，这棵树连接了两个接收方，主机 B 和主机 C。

Source	源
Host A	主机 A
Notation: (S,G) S =source G =group	标记：(S,G) S=源 G=组
224.1.1.1 traffic	去往 224.1.1.1 的流量
Receiver	源
Host B	主机 B
Receiver	源
Host C	主机 C

用标准的表示法表示，图中示例的 SPT 可以表示为(192.168.1.1, 224.1.1.1)。

(S,G)表示法按时，每个源在向每个组发送流量时，都有一个专门的 SPT——事实也是如此。

组播分发共享树

与源树的根在组播源不同，共享树将公共的根放在了网络中一些选中的位置。这个共享根称为汇集点（RP）。

组播分发共享树显示了组 224.2.2.2 的共享树，这棵树的根位于路由器 D。这棵共享树是单向的。源流量会沿着一颗源树发送给 RP。接下来，流量会从 RP 沿着共享树被一路转发给所有的接收方（除非接收方位于源和 RP 之间，此时接收方可以直接接受服务）。

Source 1	源 1
Host A	主机 A
Notation: (*,G) * =all sources G =group	标记：(*,G) *=所有源 G=组
224.2.2.2 traffic	去往 224.2.2.2 的流量
Source 2	源 2
Rendezvous Point	汇集点
Receiver	源
Host B	主机 B
Receiver	源
Host C	主机 C

在这个示例中，组播流量从源，也就是主机 A 和主机 D 发送出来，一路来到根（路由器 D），然后沿着共享树发送给了两个接收方，主机 B 和主机 C。由于组播组中的所有源都使用了公共的共享树，所以将共享树用通配符表示为(*,G)，读作“星逗号 G[①]”。此时，*表示所有源，而 G 标识组播组。因此组播分发共享树所示的共享树应该写作(*, 224.2.2.2)。

① 汉语中多读作“星点 G”。——译者注

源树和共享树都是无环的。因此消息只有在树存在叶网络的时候才需要进行复制。组播组成员可以随时加入或离开。因此分发树必须动态进行更新。当一个叶网络的所有活动接收方都停止请求某个组播组的流量时，路由器就会从分发树中修剪掉这个叶，并且不再沿着该叶向下转发流量。如果叶网络中有一台接收方变成活动接收方，并且请求了组播流量，那么路由器就会动态修改分发树，并再次开始转发流量。

源树的优势

源树的优势是可以创建出源与接收方之间的最优路径。这一优势可以确保网络转发组播流量

的延迟是最低的。但这种优化是有代价的。路由器必须给每个源维护路径信息。如果一个网络中拥有成千上万的源和成千上万的组，那么这种管理方式很快就会给路由器的资源带来负担。当组播路由表条目过多时，设备内存就会遭到严重消耗，这是网络设计师一定要考虑清楚的一大因素。

共享树的优势

共享树的优势是将每台路由器中的状态数量减少到最小。这种优势可以让只使用共享树的网络对内存提出比较低的需求。共享树的劣势在于，源与接收方之间的路径很可能不是最优路径，这会给数据包转发引入延迟。例如，在上图中，主机 A（源 1）域主机 B（一台接收方）之间的最短路径应该是通过路由器 A 和路由器 C 进行转发。但由于我们将路由器 D 设置为了共享树的根，因此流量就必须穿越路由器 A、B、D 然后转发给 C。在实施纯共享树的环境时，网络设计者在考虑汇集点的部署位置时毋须谨慎。

在单播路由中，流量会通过网络沿着从源到目的主机的一条路径进行路由。单播路由器不会考虑源地址，它只会考虑目的地址，以及如何向目的地址转发流量。路由器会扫描自己的路由表来查找目的地址，然后将单播数据包的副本通过正确的接口，从向着目的的方向转发出去。

在组播转发环境中，源会向任意组中的主机转发流量，而主机是通过组播组地址表示的。组播路由器必须判断哪个方向是上游的方向（也就是通向源的方向），哪个方向是下游的方向（也就是通向接收方的方向）。如果有多条下游路径，那么路由器就会对数据包进行复制，并且将它沿着合适的下游路径（拥有最佳单播路由度量值的下游路径）进行转发，这也未必就是所有的路径。转发组播流量这种与其说是向着接收方转发，不如说是向着源反方向转发的方式，称为逆向路径转发（RPF）。我们会在下面对 RPF 进行描述。

PIM 共享树与源树

在默认情况下，组成员会通过一颗根在 RP 的数据分发树，接收到由发送方发送给组的数据。下图显示了共享分发树的这种类型。发送方发送的数据被发送给了 RP，以便转发给加入了这棵共享树的组成员。

图 26：共享树与源树（最短路径树）

Source	源
Source tree (Shortest path tree)	源树 （最短路径树）
Shared tree from RP	包含 RP 的共享树
Receiver	接收方
Router A	路由器 A
Router B	路由器 B
Router C	路由器 C

如果数据速率可以保证，共享树中的叶路由器（即没有任何下游连接的路由器）可以使用根在源的分发树。这种分发树称为最短路径树或源树。在默认情况下，软件会在从源那里接收到第一个数据包时，就构建出一棵源树。

这个进程描述了从共享树一到到源树的迁移过程：

1 接收方加入了一个组；叶路由器 C 向 RP 发送了一条加入消息；

2 RP 将与路由器 C 的链路添加到自己的出站接口列表中；

3 源发送数据；路由器 A 将数据封装到注册消息中，并将其发送给 RP；

4 RP 沿着共享树向路由器 C 发送数据，同时向源发送了一条加入消息。此时，数据可能已经两次到达了路由器 C，一次是封装的，一次是未封装的；

5 当未封装数据到达 RP 时，它会向路由器 A 发送一条注册停止消息；

6　在默认情况下，接受第一个数据时，路由器 C 会向源发送一条计入消息；
7　当路由器 C 在(S,G)接收到了数据，它会沿着共享树发送一条修剪消息；
8　RP 从(S,G)的出站接口中删除了与路由器 C 之间的链路。RP 向源发送一条修剪消息。
加入与修剪消息是发送给源和 RP 的。这些消息都是逐跳发送的。在去往源后 RP 的路径中，每台 PIM 设备都会对这类消息进行处理。注册与注册停止消息不是逐跳发送的。这些消息是由与源直连的指定路由器发送的，最终会被这个组的 RP 接收到。
组播源会使用共享树向组发送数据。用户可以配置 PIM 设备，让它一直使用共享树。
当第一个数据包到达最后一跳路由器时，就会发生从共享树到源树的变更。这种变更取决于用户使用全局配置命令 **ip pim spt-threshold** 所配置的门限值。
最短路径树需要的内存比源树多，但是可以减少延迟。用户可能希望推迟使用最短路径树。如果不想让叶路由器立刻切换到最短路径树，用户可以设置一个必须达到的门限值。
用户可以配置 PIM 叶路由器何时可以加入指定组的最短路径树。如果源发送的流量大于等于用户设置的流量，那么多层交换机就会向源发送一条加入消息，来构建源树（即最短路径树）。如果源发送的流量速率低于门限值，叶路由器就会回退到使用共享树，并且向源发送一条修剪消息。
用户可以使用组列表（一种标准访问列表）来设置将最短路径树门限值应用于哪个组。如果设置的值是 0，或者没有使用组列表，那么门限值就会应用于所有的组。

相关主题

延迟使用 PIM 最短路径树（CLI），第 x 叶

逆向路径转发

在单播路由中，流量会通过网络沿着从源到目的主机的一条路径进行路由。单播路由器不会考虑源地址，它只会考虑目的地址，以及如何向目的地址转发流量。路由器会扫描自己的路由表来查找目的地址，然后将单播数据包的副本通过正确的接口，从向着目的的方向转发出去。
在组播转发环境中，源会向任意组中的主机转发流量，而主机是通过组播组地址表示的。组播路由器必须判断哪个方向是上游的方向（也就是通向源的方向），哪个方向是下游的方向（也就是通向接收方的方向）。如果有多条下游路径，那么路由器就会对数据包进行复制，并且将它沿着合适的下游路径（拥有最佳单播路由度量值的下游路径）进行转发，这也未必就是所有的路径。转发组播流量这种与其说是向着接收方转发，不如说是向着源反方向转发的方式，称为逆向路径转发（RPF）。RPF 是一种用来转发组播数据报的算法。
协议独立组播（PIM）使用单播路由信息来沿着从接收方到源的方向，逆向创建分发树。而组播路由器则会从源向接收方沿着分发树转发数据包。RPF 是组播转发中的核心概念。它可以让路由器正确地沿着分发树转发组播流量。RPF 会使用当前的单播路由表来判断上游邻居和下游邻居。只有当组播数据包是在上游接口接收到时，路由器才会转发该组播数据包。RPF 校验可以确保分发树是无环的。

RPF 校验

当组播数据包到达路由器时，路由器就会对数据包执行 RPF 校验。如果 RPF 校验通过，数据包就会被转发，否则就会被丢弃。
对于沿着源树转发的流量，RPF 校验的流程如下：
1　路由器查看单播路由表中的源地址，判断数据包到达的接口，是为通过通向源的逆向路径；
2　如果数据包是通过通往源的接口到达路由器的，那么 RPF 校验就会通过，数据包会通过组播路由表条目中保存的出站接口转发出去；
3　如果步骤 2 的 RPF 校验失败，数据包就会被丢弃。

这张图显示了 RPF 校验失败的示例。

图 27：RPF 校验失败

Multicast Route Table	组播路由表
Network	网络
Interface	接口
Packet arrived on wrong interface. Discard packet	数据包到达接口有误 丢弃数据包
Multicast packet from source 151.10.3.21	从源 151.10.3.21 发来的组播数据包
RPF Check Fails	RPF 校验失败

如图所示，路由器从自己的串行接口 0（即 S0）接收到了从源 151.10.3.21 发来的组播数据包。路由器对单播路由表进行了校验，结果显示 S1 是路由器用来向 151.10.3.21 转发单播数据的接口。由于数据包到达的是接口 S0，因此这个数据包被丢弃了。

这张图显示了 RPF 校验成功的示例

图 28：RPF 校验成功

Multicast Route Table	组播路由表
Network	网络
Interface	接口
Packet arrived on correct interface.	数据包到达接口无误
Multicast packet from source 151.10.3.21	从源 151.10.3.21 发来的组播数据包
RPF Check Suceeds	RPF 校验成功

在这个示例中，组播数据包到了接口 S1。路由器参考单播路由表，发现 S1 是正确的接口。RFC 校验通过，数据包得到了转发。

PIM 会使用源树和以 RP 为根的共享树来转发数据包。RPF 校验对它们的操作是不同的：

- 如果 PIM 路由器或多层设备的组播路由表中有一棵源树状态（即一个(S,G)）条目，那么它会针对组播数据包的源 IP 地址执行 RPF 校验；
- 如果 PIM 路由器或多层设备有一条共享树状态（也就是没有明确的源树状态），那么它会对 RP 地址（当成员加入组时，路由器就已经知道 RP 的地址了）执行 RPF 校验。

DMVRP 和密集模式 PIM 仅用于源树和用于 RPF。

注释： 设备不支持 DVMRP

稀疏模式 PIM 使用 RPF 查询功能来判断它需要在哪里发送加入和修剪消息：

- (S,G)加入消息（源树状态）会被发送给源；
- (*,G)加入消息（共享树状态）会被发送给 RP。

默认的 PIM 路由配置

这张表显示了设备上的默认 PIM 路由配置、

表 41：默认的组播路由配置

特性	**默认设置**
组播路由	所有接口皆禁用
PIM 版本	第 2 版
PIM 模式	未定义模式
PIM 末节路由	未配置
PIM RP 地址	未配置

PIM 域边界	禁用
PIM 组播边界	无
候选 BSR	禁用
候选 RP	禁用
最短路径树门限速率	0kb/s
PIM 路由器查询消息时间间隔	30 秒

如何配置 PIM

启用 PIM 末节路由（CLI）

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. ip pim passive

5. end

6. show ip pim interface

7. show ip igmp groups detail

8. show ip mroute

9. show running-config

10. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	指定要启用 PIM 末节路由的接口，并进入接口配置模式。 指定的接口必须为下列接口之一： • 路由端口：通过接口配置命令 **no switchport** 被配置为了三层端口的物理端口。用户还应该在这个接口上启用 IP PIM 稀疏-密集模式，并让这个接口作为静态连接成员加入一个 IGMP 静态组中； • SVI：通过全局配置命令 **interface vlan** *vlan-id*

		创建的 VLAN 接口。用户还应该在这个 VLAN 上启用 IP PIM 稀疏-密集模式，让这个接口作为静态连接成员加入一个 IGMP 静态组中，并且在 VLAN、IGMP 静态组和物理接口上启用 IGMP snooping； 这些接口上必须配置 IP 地址。
步骤 4	**ip pim passive** 示例： `Device(config-if)# ip pim passive`	在接口上配置 PIM 末节特性
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show ip pim interface** 示例： `Device# show ip pim interface`	（可选）查看各个接口上启用的 PIM 末节
步骤 7	**show ip igmp groups detail** 示例： `Device# show ip igmp groups detail`	（可选）查看加入了特定组播源组的感兴趣客户端
步骤 8	**show ip mroute** 示例： `Device# show ip mroute`	（可选）查看 IP 组播路由表
步骤 9	**show running-config** 示例： `Device# show running-config`	查看之前所作的配置
步骤 10	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中

配置一个汇集点

如果接口工作在稀疏-密集模式，而用户又希望将这个组作为稀疏组处理时，就必须配置一个汇集点（RP）。用户可以使用下面的方式进行配置：

- 手动给组播组分配一个 RP；
- 使用一个独立的、Inspur 私有的、独立于 PIMv1 的协议进行分配，包括：
  - 在新网络中设置 Auto-RP；
  - 将 Auto-RP 添加到当前的稀疏模式云中；
  - 防止加入消息被发送给非法 RP；
  - 过滤入站方向的 RP 通告消息；
- 使用 IETF（互联网工程任务组）定义的标准协议，包括配置 PIMv2 BSR。

注释： 用户可以使用 Auto-RP、BSR，或者将这两者结合起来使用，具体做法取决于网络中运行的 PIM 版本和路由器的类型。如需进一步了解在网络中如何处理 PIM 的不同版本，可以参考 PIMv1 与 PIMv2 的互操作性。

手动给组播组分配一个 RP（CLI）

如果一个组的汇集点（RP）会通过一种动态机制（如 Auto-RP 或 BSR）学习到的，那么用户就不需要对 RP 执行下面的操作。

组播流量的发送方会通过从源第一跳路由器（指定路由器）那里接收到的注册消息学习到 RP，并且将它发送给 RP。组播数据包的接收方会通过 RP 使用发送显式加入消息的方式来加入组播组。

注释： RP 不是组播组的成员，它们只是组播源与组成员*见面的地点*。

用户可以给多个通过访问列表定义的组配置同一个 RP。如果一个组没有配置 RP，那么多层设备就会将这个组视为密集模式，并且按照密集模式 PIM 的方式来执行操作。

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. ip pim rp-address *ip-address* [*access-list-number*] [**override**]

4. access-list *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip pim rp-address** *ip-address* [*access-list-number*] [**override**] 示例： `Device(config)#` **ip pim rp-**	配置 PIM RP 的地址。 在默认情况下，设备上是没有配置 PIM RP 地址的。用户必须在所有路由器和多层设备（也包括 RP）上配置 RP 的地址。 **注释：**如果有一个组没有配置 RP，那么设备就会视该组为密集模式，使用密集模式 PIM 的方式进

	address 10.1.1.1 20 override	行操作。 PIM 设备可以同时为多个组的 RP。在每个 PIM 域中，一个 RP 地址只能使用一次。访问列表的条件会指定这台设备是哪些组的 RP。 • 在 *ip-address* 部分，用点分十进制的格式输入 RP 的单播地址； • （可选）在 *access-list-number* 部分，输入 IP 标准访问列表编号，范围是从 1 到 99。如果没有配置访问列表，那么 RP 就会用于所有组； • （可选）关键字 **override** 表示如果这条命令配置的 RP 与通过 Auto-RP 或 BSR 学习到的 RP 出现了冲突，那么以这条命令配置的 RP 为准
步骤 4	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] 示例： `Device(config)#` **access-list 25 permit 10.5.0.1 255.224.0.0**	创建一个标准访问列表，用户应根据需要重复配置这条命令。 • 在 *access-list-number* 部分，输入步骤 2 中配置的访问列表编号； • 关键字 **deny** 是指在条件匹配时，即拒绝访问； • 关键字 **permit** 是指在条件匹配时，即允许访问； • 在 *source* 部分，输入 RP 应该使用的组播组地址； • （可选）在 *source-wildcard* 部分，用点分十进制格式输入要应用于源的反掩码，配置反掩码应在要忽略的位取 0。 访问列表的最后永远包含一条隐式的全部拒绝语句
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** 示例： `Device#` **show running-config**	查看之前所作的配置
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

在新网络中设置 Auto-RP（CLI）

如果用户正在新网络中设置 Auto-RP，那就不需要默认 RP，因为用户要将所有接口都配置为稀疏-密集模式。

注释： 如果用户想要将一台 PIM 路由器配置为本地组的 RP，请忽略下面流程中的步骤 3。

总步骤

1. enable

2. show running-config

3. configure terminal

4. ip pim send-rp-announce *interface-id* **scope** *ttl* **group-list** *access-list-number* **interval** *seconds*

5. access-list *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]

6. ip pim send-rp-discovery scope *ttl*

7. end

8. show running-config

9. show ip pim rp mapping

10. show ip pim rp

11. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**show running-config** 示例： `Device#` **show running-config**	查看所有 PIM 设备上已经配置的默认 RP，以及稀疏模式网络中的 RP。这是用户之前使用全局配置命令 **ip pim rp-address** 配置的。 **注释**：对于稀疏-密集模式环境，不需要执行这一步。 用户选择的 RP 应该拥有良好的连通性，而且在整个网络中都是可用的。要将这个 RP 用于全局组（例如 224.x.x.x 和其他全局组）。不要重新配置这个 RP 服务的组地址范围。通过 Auto-RP 自动发现的 RP 会犹于静态配置的 RP。要给本地组准备一个第二 RP。
步骤 3	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 4	**ip pim send-rp-announce** *interface-id* **scope** *ttl* **group-list** *access-list-number* **interval** *seconds* 示例： `Device(config)#` **ip pim send-rp-announce gigabitethernet 1/0/5 scope 20 group-list 10 interval 120**	配置要成为本地组候选 RP 的 PIM 设备。 • 在 *interface-id* 部分，输入标识 RP 地址的接口类型与接口号。有效接口包括物理端口、port channel 和 VLAN； • 在 **scope** *ttl* 部分，设置生存时间的跳数。要把跳数设置得足够大，让 RP 通告消息可以到达网络中的所有映射代理。这个参数没有默认设置，取值范围是从 1 到 255； • 在 **group-list** *access-list-number* 部分，输入 IP 标准访问列表的编号，范围是从 1 到 99。如果没有配置访问列表，那么 RP 就会用于所有

		组； • 在 **interval** *seconds* 部分，设置通告消息的发送频率。默认值为 60 秒。取值范围是从 1 到 16383
步骤 5	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] 示例： `Device(config)#` **access-list 10 permit 10.10.0.0**	创建一个标准访问列表，用户应根据需要重复配置这条命令。 • 在 *access-list-number* 部分，输入步骤 2 中配置的访问列表编号； • 关键字 **deny** 是指在条件匹配时，即拒绝访问； • 关键字 **permit** 是指在条件匹配时，即允许访问； • 在 *source* 部分，输入 RP 应该使用的组播组地址； • （可选）在 *source-wildcard* 部分，用点分十进制格式输入要应用于源的反掩码，配置反掩码应在要忽略的位取 0。 **注释**：切记访问列表的最后永远包含一条隐式的全部拒绝语句
步骤 6	**ip pim send-rp-discovery scope** *ttl* 示例： `Device(config)#` **ip pim send-rp-discovery scope 50**	找一台连通性不太可能会受影响的设备，把它的角色设置为 RP 映射代理。 在 *ttl* 部分，设置生存时间的跳数，以限制 RP 发现数据包的转发范围。所有距离源在跳数范围之内的设备都会接收到这个 Auto-RP 发现消息。这些消息会告诉其他设备要使用那个组-RP 映射，以避免出现冲突（比如组-RP 范围重复）。这个值没有默认设置。取值范围是从 1 到 255
步骤 7	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 8	**show running-config** 示例： `Device#` **show running-config**	查看之前所作的配置
步骤 9	**show ip pim rp mapping** 示例： `Device#` **show ip pim rp mapping**	查看缓存在相关组播路由条目中的活动 RP
步骤 10	**show ip pim rp** 示例： `Device#` **show ip pim rp**	查看缓存在路由表中的信息
步骤	**copy running-config startup-**	（可选）将输入的条目保存到配置文件中

11	**config** 示例： `Device#` **copy running-config startup-config**	

将 Auto-RP 添加到当前的稀疏模式云中（CLI）

这部分包括了将 Auto-RP 部署到现成的稀疏模式云中时的操作建议，这种配置方式可以将对当前组播基础设备造成的影响降至最低。

这个流程是可选的。

总步骤

1. enable

2. show running-config

3. configure terminal

4. ip pim send-rp-announce *interface-id* **scope** *ttl* **group-list** *access-list-number* **interval** *seconds*

5. access-list *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]

6. ip pim send-rp-discovery scope *ttl*

7. end

8. show running-config

9. show ip pim rp mapping

10. show ip pim rp

11. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**show running-config** 示例： `Device#` **show running-config**	查看所有 PIM 设备上已经配置的默认 RP，以及稀疏模式网络中的 RP。这是用户之前使用全局配置命令 **ip pim rp-address** 配置的。 **注释：**对于稀疏-密集模式环境，不需要执行这一步。 用户选择的 RP 应该拥有良好的连通性，而且在整个网络中都是可用的。要将这个 RP 用于全局组（例如 224.x.x.x 和其他全局组）。不要重新配置这个 RP 服务的组地址范围。通过 Auto-RP 自动发现的 RP 会犹于静态配置的 RP。要给本地组准备一个第二 RP。
步骤 3	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 4	**ip pim send-rp-announce** *interface-id*	配置要成为本地组候选 RP 的 PIM 设备。 • 在 *interface-id* 部分，输入标识 RP 地址的接口

	scope *ttl* **group-list** *access-list-number* **interval** *seconds* 示例： `Device(config)#` **ip pim send-rp-announce gigabitethernet 1/0/5 scope 20 group-list 10 interval 120**	类型与接口号。有效接口包括物理端口、port channel 和 VLAN； • 在 **scope** *ttl* 部分，设置生存时间的跳数。要把跳数设置得足够大，让 RP 通告消息可以到达网络中的所有映射代理。这个参数没有默认设置，取值范围是从 1 到 255； • 在 **group-list** *access-list-number* 部分，输入 IP 标准访问列表的编号，范围是从 1 到 99。如果没有配置访问列表，那么 RP 就会用于所有组； • 在 **interval** *seconds* 部分，设置通告消息的发送频率。默认值为 60 秒。取值范围是从 1 到 16383
步骤 5	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] 示例： `Device(config)#` **access-list 10 permit 224.0.0.0 15.255.255.255**	创建一个标准访问列表，用户应根据需要重复配置这条命令。 • 在 *access-list-number* 部分，输入步骤 2 中配置的访问列表编号； • 关键字 **deny** 是指在条件匹配时，即拒绝访问； • 关键字 **permit** 是指在条件匹配时，即允许访问； • 在 *source* 部分，输入 RP 应该使用的组播组地址； • （可选）在 *source-wildcard* 部分，用点分十进制格式输入要应用于源的反掩码，配置反掩码应在要忽略的位取 0。 切记访问列表的最后永远包含一条隐式的全部拒绝语句
步骤 6	**ip pim send-rp-discovery scope** *ttl* 示例： `Device(config)#` **ip pim send-rp-discovery scope 50**	找一台连通性不太可能会受影响的设备，把它的角色设置为 RP 映射代理。 在 *ttl* 部分，设置生存时间的跳数，以限制 RP 发现数据包的转发范围。所有距离源在跳数范围之内的设备都会接收到这个 Auto-RP 发现消息。这些消息会告诉其他设备要使用那个组-RP 映射，以避免出现冲突（比如组-RP 范围重复）。这个值没有默认设置。取值范围是从 1 到 255。 **注释**：要移除这台设备的 RP 映射代理角色，可以使用全局配置命令 **no ip pim send-rp-discovery**
步骤 7	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 8	**show running-config** 示例：	查看之前所作的配置

	Device# **show running-config**	
步骤 9	**show ip pim rp mapping** 示例： Device# **show ip pim rp mapping**	查看缓存在相关组播路由条目中的活动 RP
步骤 10	**show ip pim rp** 示例： Device# **show ip pim rp**	查看缓存在路由表中的信息
步骤 11	**copy running-config startup-config** 示例： Device# **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

防止加入消息被发送给非法 RP

用户可以使用特权 EXEC 命令 **show running-config** 来判断整个网络中的设备上之前是否配置了命令 **ip pim accept-rp**。在任何没有配置命令 **ip pim accept-rp** 的设备上，这个问题可以以后再行解决。在那些已经配置了命令 **ip pim accept-rp** 的路由器或多层设备上，用户必须再次输入这条命令来接受信通告的 RP。

要接受所有 Auto-RP 通告的 RP，并且拒绝所有其他 RP，需要输入全局配置命令 **ip pim accept-rp auto-rp**。

这个流程是可选的。

过滤入站 RP 通告消息（CLI）

用户可以在映射代理上添加配置命令，以防止被人恶意配置的路由器伪装成候选 RP 给网络造成问题。

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. ip pim rp-announce-filter rp-list *access-list-number* **group-list** *access-list-number*

4. access-list *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： Device> **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device#` **configure terminal**	
步骤 3	**ip pim rp-announce-filter rp-list** *access-list-number* **group-list** *access-list-number* 示例： `Device(config)#` **ip pim rp-announce-filter rp-list 10 group-list 14**	过滤入站 RP 通告消息。 要在网络中的每台映射代理上都输入这条命令。如果不输入这条命令，所有入站 RP 通告消息默认都会被接受。 在 ***rp-list*** *access-list-numbe* 部分，输入一个候选 RP 地址的访问列表，这个候选 RP 地址是 **group-list** *access-list-number* 多定义的组范围内，可以接受的地址。如果没有配置这个变量，那么过滤策略就会应用于所有组播组。 如果使用了多台映射代理，那么所有映射代理上配置的过滤策略都要一致，这样才能确保在组-RP 映射信息中不会发生冲突
步骤 4	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] 示例： `Device(config)#` **access-list 10 permit 10.8.1.0 255.255.224.0**	创建一个标准访问列表，用户应根据需要重复配置这条命令。 • 在 *access-list-number* 部分，输入步骤 2 中配置的访问列表编号； • 关键字 **deny** 是指在条件匹配时，即拒绝访问； • 关键字 **permit** 是指在条件匹配时，即允许访问； • 创建一个访问列表来设置映射代理从哪台路由器和多层设备接受候选 RP 通告（rp-list ACL）； • 创建一个访问列表来设置要接受或拒绝来自哪些组播组的消息（group-list ACL）； • 在 *source* 部分，输入 RP 应该使用的组播组地址； • （可选）在 *source-wildcard* 部分，用点分十进制格式输入要应用于源的反掩码，配置反掩码应在要忽略的位取 0。 访问列表的最后永远包含一条隐式的全部拒绝语句
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** 示例： `Device#` **show running-config**	查看之前所作的配置
步骤 7	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

	示例： `Device#` **copy running-config startup-config**	

配置 PIMv2 BSR

配置 PIMv2 BSR 的过程中，可能会包含下列配置任务：

- 定义 PIM 域边界；
- 定义 IP 组播边界；
- 配置候选 BSR；
- 配置候选 RP。

定义 PIM 域边界（CLI）

用户可以执行下面的步骤来配置 PIM 域边界。这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. ip pim bsr-border

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	指定要配置的接口，并进入接口配置模式。 指定的接口必须为下列接口之一： • 路由端口：通过接口配置命令 **no switchport** 被配置为了三层端口的物理端口。用户还应该在这个接口上启用 IP PIM 稀疏-密集模式，并让这个接口作为静态连接成员加入一个 IGMP 静态组中； • SVI：通过全局配置命令 **interface vlan** *vlan-id* 创建的 VLAN 接口。用户还应该在这个 VLAN 上启用 IP PIM 稀疏-密集模式，让这个接口作为静态连接成员加入一个 IGMP 静态组中，并且在 VLAN、IGMP 静态组和物理接口上启用

		IGMP snooping； 这些接口上必须配置 IP 地址。
步骤 4	**ip pim bsr-border** 示例： `Device(config-if)# ip pim bsr-border`	给 PIM 域定义一台 PIM 自举消息边界。 在连接其他边界 PIM 域的接口上输入这条命令。这条命令会让设备在该接口上既不发送也不接收 PIMv2 BSR 消息。 **注释：**要移除 PIM 边界，可以使用接口配置命令 **no ip pim bsr-border**
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** 示例： `Device# show running-config`	查看之前所作的配置
步骤 7	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中

定义 IP 组播边界（CLI）

用户定义了一个组播边界来防止 Auto-RP 消息进入这个 PIM 域当中。用户可以创建一个访问列表来拒绝去往 224.0.1.39 和 224.0.1.40 的数据包，而这些数据包中会承载 Auto-RP 信息。这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. access-list *access-list-number* **deny** *source* [*source-wildcard*]

4. interface *interface-id*

5. ip multicast boundary *access-list-number*

6. end

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例：	进入全局配置模式

	Device# **configure terminal**	
步骤 3	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] 示例： Device(config)# **access-list 12 deny 224.0.1.39** **access-list 12 deny 224.0.1.40**	创建一个标准访问列表，用户应根据需要重复配置这条命令。 • 在 *access-list-number* 部分，输入步骤 2 中配置的访问列表编号； • 关键字 **deny** 是指在条件匹配时，即拒绝访问； • 关键字 **permit** 是指在条件匹配时，即允许访问； • 在 *source* 部分，输入 RP 应该使用的组播组地址； • （可选）在 *source-wildcard* 部分，用点分十进制格式输入要应用于源的反掩码，配置反掩码应在要忽略的位取 0。 **注释**：切记访问列表的最后永远包含一条隐式的全部拒绝语句
步骤 4	**interface** *interface-id* 示例： Device(config)# **interface gigabitethernet 1/0/1**	指定要配置的接口，并进入接口配置模式。 指定的接口必须为下列接口之一： • 路由端口：通过接口配置命令 **no switchport** 被配置为了三层端口的物理端口。用户还应该在这个接口上启用 IP PIM 稀疏-密集模式，并让这个接口作为静态连接成员加入一个 IGMP 静态组中； • SVI：通过全局配置命令 **interface vlan** *vlan-id* 创建的 VLAN 接口。用户还应该在这个 VLAN 上启用 IP PIM 稀疏-密集模式，让这个接口作为静态连接成员加入一个 IGMP 静态组中，并且在 VLAN、IGMP 静态组和物理接口上启用 IGMP snooping； 这些接口上必须配置 IP 地址。
步骤 5	**ip multicast boundary** *access-list-number* 示例： Device(config-if)# **ip multicast boundary 12**	配置边界，设置在步骤 2 中创建的访问列表
步骤 6	**end** 示例： Device(config)# end	返回特权 EXEC 模式
步骤 7	**show running-config** 示例： Device# **show running-config**	查看之前所作的配置
步骤 8	**copy running-config startup-**	（可选）将输入的条目保存到配置文件中

	config 示例： `Device#` **copy running-config startup-config**	

配置候选 BSR（CLI）

用户可以配置一个或多个候选 BSR。用来充当 BSR 的设备应该与其他设备之间拥有良好的连通性，而且 BSR 应该处于网络的骨干区域。

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. ip pim bsr-candidate *interface-id hash-mask-length* [*priority*]

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip pim bsr-candidate** *interface-id hash-mask-length* [*priority*] 示例： `Device(config)#` **ip pim bsr-candidate gigabitethernet 1/0/3 28 100**	配置设备，让他成为候选 BSR。 • 在 *interface-id* 部分，进入给 BSR 地址提取地址的那个这个设备接口。这个接口必须启用 PIM。有效接口包括物理端口、port channel 和 VLAN； • 在 *hash-mask-length* 部分，设置掩码长度（最大 32 位）。掩码要先与组地址执行与运算，然后再调用散列函数。所有散列参数相同的组都对应同一个 RP。例如，如果掩码为 24 位，那么就只有组地址的前 24 位会参与散列计算； • （可选）在 *priority* 部分，输入 0 到 255 之间的数字。优先级取值越大的设备越会优选为 BSR。如果优先级值相同，那么 IP 地址最高的设备会被选举为 BSR。默认值为 0
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

步骤 5	**show running-config** 示例： `Device#` **show running-config**	查看之前所作的配置
步骤 6	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置候选 RP（CLI）

用户可以配置一台或多台候选 RP。RP 和 BSR 一样应该与其他设备之间拥有良好的连通性，同时 RP 也应该处于网络的骨干区域。RP 既可以为整个 IP 组播地址空间提供服务，也可以仅为其中一部分空间提供服务。候选 RP 会向 BSR 发送候选 RP 通告。

这个流程是可选的。

在开始前

在判断哪台设备应该成为 RP 之前，应该考虑下面几项因素：

- 在一个由 Inspur 路由器和多层设备组成，且只使用了 Auto-RP\的网络中，任何设备都可以配置为 RP；
- 在只包含 Inspur PIMv2 路由器和多层设备，同时还有其他厂商路由器的网络中，任何设备都可以配置为 RP；
- 在由 Inspur PIMv1 路由器、Inspur PIMv2 路由器和其他厂商路由器组成的网络中，只将 Inspur PIMv2 路由器和多层设备配置为 RP

总步骤

1. enable

2. configure terminal

3. ip pim rp-candidate *interface-id* [**group-list** *access-list-number*]

4. access-list *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip pim rp-candidate** *interface-id* [**group-list** *access-list-number*]	配置设备，让他成为候选 RP。 • 在 *interface-id* 部分，选择其 IP 地址要通告为候选 RP 地址的那个接口。有效接口包括物理

	示例： `Device(config)# ip pim rp-candidate gigabitethernet 1/0/5 group-list 10`	端口、port channel 和 VLAN； • （可选）在 **group-list** *access-list-number* 部分，输入 IP 标准访问列表的编号，取值范围是从 1 到 99。如果没有设置组列表，那么这台设备就会称为所有组的候选 RP
步骤 4	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] 示例： `Device(config)# access-list 10 permit 239.0.0.0 0.255.255.255`	创建一个标准访问列表，用户应根据需要重复配置这条命令。 • 在 *access-list-number* 部分，输入步骤 2 中配置的访问列表编号； • 关键字 **deny** 是指在条件匹配时，即拒绝访问； • 关键字 **permit** 是指在条件匹配时，即允许访问； • 在 *source* 部分，输入 RP 应该使用的组播组地址； • （可选）在 *source-wildcard* 部分，用点分十进制格式输入要应用于源的反掩码，配置反掩码应在要忽略的位取 0。 访问列表的最后永远包含一条隐式的全部拒绝语句
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** 示例： `Device# show running-config`	查看之前所作的配置
步骤 7	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中

配置包含 Auto-RP 的稀疏模式（CLI）

在开始前

- 配置为稀疏-密集模式的接口会执行稀疏模式或密集模式的操作，具体执行哪种操作取决于组播组的操作模式。用户必须判断如何配置自己的接口；
- 在所有配置 Auto-RP 时需要进行配置的访问列表，都要在开始执行下面的配置任务之前配置好。

注释： 如果一个组没有已知 RP，而接口又配置为了稀疏-密集模式，那么这个接口就会工作在密集模式下，数据会通过这个接口进行泛洪。要想避免数据泛洪，就要配置 Auto-RP 侦

听器，然后再将接口配置为稀疏模式。

在配置 Auto-RP 时，用户必须要么配置 Auto-RP 侦听器特性（步骤 5），要么设置稀疏模式（步骤 7）或者设置稀疏-密集模式（步骤 8）；

在配置稀疏-密集模式时，密集模式故障切换可能会导致网络密集模式泛洪。要想避免这种情况，要在使用 PIM 稀疏模式的同时配置 Auto-RP 侦听器特性。

用户可以按照下面的流程来配置自动汇集点（Auto-RP）。Auto-RP 也可以和任意播 RP 一起使用。

总步骤

1. enable

2. configure terminal

3. ip multicast-routing [**distributed**]

4. 执行步骤5到步骤7，或者执行步骤6到步骤8

5. ip pim autorp listener

6. interface *type number*

7. ip pim sparse-mode

8. ip pim sparse-dense-mode

9. exit

10. 在所有 PIM 接口上重复步骤 1 到步骤 9 的操作

11. ip pim send-rp-announce {*interface-type interface-number* | *ip-address*} **scope** *ttl-value* [**group-list**

access-list] [**interval** *seconds*] [**bidir**]

12. ip pim send-rp-discovery [*interface-type interface-number*] **scope** *ttl-value* [**interval** *seconds*]

13. ip pim rp-announce-filter rp-list *access-list* **group-list** *access-list*

14. no ip pim dm-fallback

15. interface *type number*

16. ip multicast boundary *access-list* [**filter-autorp**]

17. end

18. show ip pim autorp

19. show ip pim rp [**mapping**] [*rp-address*]

20. show ip igmp groups [*group-name* | *group-address*| *interface-type interface-number*] [**detail**]

21. show ip mroute [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*]

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip multicast-routing** [**distributed**]	启用 IP 组播路由。 • 使用关键字 **distributed** 启用组播分布式交换

	示例： `Device(config)# ip multicast-routing`	
步骤 4	执行步骤 5 到步骤 7 或者执行步骤 6 和步骤 8	
步骤 5	**ip pim autorp listener** 示例： `Device(config)# ip pim autorp listener`	让去往两个 Auto-RP 组 224.0.1.39 和 224.1.1.40 的 IP 组播流量通过 PIM 密集模式泛洪的方式，通过 PIM 稀疏模式的接口。 • 如果要在步骤 8 中配置稀疏-密集模式，则应跳过这一步
步骤 6	**interface** *type number* 示例： `Device(config)# interface Gigabitethernet 1/0/0`	选择一个与启用了 PIM 的主机相连的接口
步骤 7	**ip pim sparse-mode** 示例： `Device(config-if)# ip pim sparse-mode`	在一个接口上启用 PIM 稀疏模式。如果在稀疏模式下配置 Auto-RP，那么用户也必须在下一步中配置 Auto-RP 侦听器。 • 如果要在步骤 8 中配置稀疏-密集模式，则应跳过这一步
步骤 8	**ip pim sparse-dense-mode** 示例： `Device(config-if)# ip pim sparse-dense-mode`	在一个接口上启用 PIM 稀疏-密集模式。 • 如果要在步骤 8 中配置稀疏-密集模式，则应跳过这一步
步骤 9	**exit** 示例： `Device(config-if)# exit`	离开接口配置模式并返回全局配置模式
步骤 10	在所有 PIM 接口上重复步骤 1 到步骤 9	
步骤 11	**ip pim send-rp-announce** {*interface-type interface-number* | *ip-address*} **scope** *ttl-value* [**group-list** *access-list*] [**interval** *seconds*] [**bidir**] 示例： `Device(config)# ip pim send-rp-announce loopback0 scope 31 group-`	将 RP 通过从所有启用了 PIM 的接口发送出去。 • 只在 RP 设备上执行这一步骤； • 使用和参数来定义用哪个 IP 地址来充当 RP 地址； • 使用参数来将直连的 IP 地址设置为 RP 地址； **注释**：如果用户在这条命令中配置了参数 *ip-address*，那么 RP 通告消息就会以该 IP 地址连接的接口作为源地址。（也就是说，RP 通告消息 IP 头部的源地址字段，是该接口的 IP 地址） • 在示例中，我们将接口的最大跳数设置为了 31 跳。设备希望标识为 RP 的 IP 地址，是 loopback 0 接口的 IP 地址。我们用编号为 5 的

	`list 5`	访问列表描述了让这台设备充当哪些组的 RP
步 骤 12	**ip pim send-rp-discovery scope** *ttl* 示例： `Device(config)# ip pim send-rp-discovery loopback 1 scope 31`	将这台设置配置为一台 RP 映射代理。 • 在充当 RP 映射代理的设备或同时充当 RP 和 RP 映射代理的设备上执行这一步操作。 **注释**：Auto-RP 可以让 RP 功能在一台设备上独立运行，同时让 RP 映射代理在一台或多台设备上运行。用户可以在一台结合了 RP 和 RP 映射代理功能的设备上，同时部署 RP 和 RP 映射代理。 • 使用可选参数 *interface-type* 和 *interface-number* 来定义将哪个 IP 地址用作 RP 映射代理的源地址； • 使用关键字 **scope** 和参数 *ttl-value* 来设置 Auto-RP 发现消息 IP 头部中的 TTL（生存事件）值； • 使用可选关键字 **interval** 和参数 *seconds* 来设置 Auto-RP 发现消息的发送时间间隔； **注释**：将 Auto-RP 发现消息的发送时间间隔降低到一个比默认值 60 秒更低的数值，会让设备更加频繁地泛洪组-RP 映射。在有些网络环境中，降低时间间隔的弊（产生更多管理控制数据包）大于利（对组-RP 映射的更新更加及时）。 • 这个示例所示为在 loopback 1 接口上将 Auto-RP 发现消息限制为 31 跳
步 骤 13	**ip pim rp-announce-filter rp-list** *access-list* **group-list** *access-list* 示例： `Device(config)# ip pim rp-announce-filter rp-list 1 group-list 2`	过滤候选 RP（C-RP）发送给 RP 映射代理的入站 RP 通告消息。 • 仅在 RP 映射代理上执行这条命令
步 骤 14	**no ip pim dm-fallback** 示例： `Device(config)# no ip pim dm-fallback`	（可选）防止 PIM 密集模式回退 • 如果所有接口都被配置为 PIM 稀疏模式，那就应当跳过这一步。 **注释**：如果所有接口都（用命令 **no ip pim dm-fallback**）配置为了 PIM 稀疏模式，那么设备默认就会按照命令 **no ip pim dm-fallback** 的方式执行操作
步 骤 15	**interface** *type number* 示例： `Device(config)# interface Gigabitethernet 1/0/0`	选择一个与启用了 PIM 的主机相连的接口
步 骤	**ip multicast boundary**	配置一个管理作用范围边界

16	*access-list* [**filter-autorp**] 示例： `Device(config-if)# ip multicast boundary 10 filter-autorp`	• 在与其他设备的边界上执行这一步； • 案例中没有显示访问列表； • 关键字为 **deny** 的访问列表条目会给匹配该条目的数据包创建出一个组播边界
步骤 17	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 18	**show ip pim autorp** 示例： `Device# show ip pim autorp`	（可选）显示 Auto-RP 的信息
步骤 19	**show ip pim rp** [**mapping**] [*rp-address*] 示例： `Device# show ip pim rp mapping`	（可选）显示网络中已知的 RP，同时显示设备学习到各个 RP 的途径
步骤 20	**show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**] 示例： `Device# show ip igmp groups`	（可选）显示拥有与设备直连的接收方，并且是通过 IGMP（互联网组管理协议）学习到的组播组。只有在输入这条命令时，网络中的接收方仍然是活动的时，这条命令的输出信息中才会显示该接收方的信息
步骤 21	**show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*] 示例： `Device# show ip mroute cbone-audio`	（可选）显示 IP 组播路由（mroute）的信息

延迟使用 PIM 最短路径树（CLI）

用户可以执行下面的步骤来配置一个流量的门限值，让设备只有在达到这个门限值时才会将组播路由从源树切换到最短路径树。
这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. access-list *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]

4. ip pim spt-threshold {*kbps* | **infinity**} [**group-list** *access-list-number*]

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] 示例： `Device(config)#` **access-list 12 deny 224.0.1.39** **access-list 12 deny 224.0.1.40**	创建一个标准访问列表，用户应根据需要重复配置这条命令。 • 在 *access-list-number* 部分，输入步骤 2 中配置的访问列表编号； • 关键字 **deny** 是指在条件匹配时，即拒绝访问； • 关键字 **permit** 是指在条件匹配时，即允许访问； • 在 *source* 部分，输入 RP 应该使用的组播组地址； • （可选）在 *source-wildcard* 部分，用点分十进制格式输入要应用于源的反掩码，配置反掩码应在要忽略的位取 0。 访问列表的最后永远包含一条隐式的全部拒绝语句
步骤 4	**ip pim spt-threshold** {*kbps* | **infinity**} [**group-list** *access-list-number*] 示例：	设置在切换到最短路径树（spt）之前，流量必须达到的门限值。 • 在 *kbps* 部分，设置流量速率（单位是 kb/s）。默认值为 0kbps。 • **注释**：虽然这个参数的取值范围是从 0 到 4294967，但由于设备硬件的限制，0kbps 是唯

	Device(config)# **ip pim spt-threshold infinity group-list 16**	一有效的参数。 • 如果希望指定组的所有源都使用共享树，而永远不会切换到源树，那就要设置 **infinity** 关键字。 • （可选）在部分，设置步骤 2 中创建的访问列表。如果将这个值设置为 0 或者没有使用组列表，那么这个门限值就会应用于所有组
步骤 5	**end** 示例： Device(config)# end	返回特权 EXEC 模式
步骤 6	**show running-config** 示例： Device# **show running-config**	查看之前所作的配置
步骤 7	**copy running-config startup-config** 示例： Device# **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

修改 PIM 路由器查询消息时间间隔（CLI）

PIM 路由器和多层设备会发送 PIM 路由器查询消息，在各个局域网段（即子网）中查询哪台设备是这里的指定路由器（DR）。DR 负责向直连局域网中的所有主机发送 IGMP 主机查询消息。

在 PIM DM 环境中，只有使用了 IGMPv1，DR 的存在才有一一。IGMPv1 没有 IGMP 查询器选举进程，因此选举出来的 DR 会执行 IGMP 查询器的工作。在 PIM SM 环境中，DR 是与组播源直连的设备。它会发送 PIM 注册消息来通过 RP，源发送的组播流量需要沿着共享树向下转发。此时，DR 是 IP 地址最大的那台设备。

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. ip pim query-interval *seconds*

5. end

6. show ip igmp interface [*interface-id*]

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密码

	示例： `Device> enable`	
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)# interface gigabitethernet 1/0/1`	指定要配置的接口，并进入接口配置模式。 指定的接口必须为下列接口之一： • 路由端口：通过接口配置命令 **no switchport** 被配置为了三层端口的物理端口。用户还应该在这个接口上启用 IP PIM 稀疏-密集模式，并让这个接口作为静态连接成员加入一个 IGMP 静态组中； • SVI：通过全局配置命令 **interface vlan** *vlan-id* 创建的 VLAN 接口。用户还应该在这个 VLAN 上启用 IP PIM 稀疏-密集模式，让这个接口作为静态连接成员加入一个 IGMP 静态组中，并且在 VLAN、IGMP 静态组和物理接口上启用 IGMP snooping； 这些接口上必须配置 IP 地址。
步骤 4	**ip pim query-interval** *seconds* 示例： `Device(config-if)# ip pim query-interval 45`	配置设备发送 PIM 路由器查询消息的频率。 默认值为 30 秒。取值范围是从 1 到 65535
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show ip igmp interface** [*interface-id*] 示例： `Device# show ip igmp interface`	查看之前所作的配置
步骤 7	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中

查看 PIM 的操作

在 PIM-SM 或 PIM-SSM 网络中查看 IP 组播的操作

当用户在 PIM-SM 网络环境中或 PIM-SSM 网络环境中查看 IP 组播操作时，一种比较有用的方法是从下一跳路由器上开始进行查看，然后沿着 STP 进行查看，一直查看到第一跳路由器。这种查看的目的是确保 IP 组播流量是按照正确的方式，在 IP 组播网络中进行路由的。用户可以执行下面这些可选的操作，来验证 PIM-SM 或 PIM-SSM 网络中的 IP 组播操作。当源和接收方的通信出现问题时，这些操作可以找出有问题的那一跳设备。

注释： 如果数据包没有到达预期的目的设备，用户可能会考虑禁用 IP 组播快速交换，这种机制会让路由器进入交换模式。如果在 IP 组播快速交换机制被禁用之后，数据包就能够到达既定的目的设备，那么这个问题就最有可能与 IP 组播快速交换有关。

在第一跳路由器上查看 IP 组播

在第一跳路由器上输入这些命令来查看第一跳路由器上的 IP 组播操作。

总步骤

1. **enable**
2. **show ip mroute** [*group-address*]
3. **show ip mroute active** [*kb/s*]

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码

步骤 2	**show ip mroute** [*group-address*]
	
	示例：
	`Device# show ip mroute 239.1.2.3`
	`(*, 239.1.2.3), 00:18:10/stopped, RP 172.16.0.1, flags: SPF`
	`Incoming interface: Serial1/0, RPF nbr 172.31.200.2`
	`Outgoing interface list: Null`
	`(10.0.0.1, 239.1.2.3), 00:18:10/00:03:22, flags: FT`
	`Incoming interface:`
		确认第一跳路由器上已经给组播路由设置了 F 标记

	GigabitEthernet0/0/0, RPF nbr 0.0.0.0 Outgoing interface list: Serial1/0, Forward/Sparse-Dense, 00:18:10/00:03:19	
步骤 3	**show ip mroute active** [*kb/s*] 示例： Device# **show ip mroute active** Active IP Multicast Sources - sending >= 4 kbps Group: 239.1.2.3, (?) Source: 10.0.0.1 (?) Rate: 20 pps/4 kbps(1sec), 4 kbps(last 30 secs), 4 kbps(life avg)	显示向组发送数据的活动组播源信息。这条命令的输出信息可以提供活动源发送组播数据包的速率。注释：在默认情况下，若在使用命令 **show ip mroute** 时输入 **active** 关键字，系统会显示那些向组发送数据的速率大等于 4kb/s 的活动源。要查看以低速（也就是速率低于 4kb/s）向组发送数据的活动源，需要在 *kb/s* 部分将数值设置为 1。将这个参数的数值设置为 1 之后，系统就会显示以大于等于 1kb/s 的速率向组发送数据的活动源，这可以有效地显示出所有可能的活动源所发送的流量

沿着 SPT 在路由器上查看 IP 组播

用户可以沿着 SPT，依次在各台路由器上输入下列命令，来查看 PIM-SM 网络或 PIM-SSM 网络中，SPT 沿线各个路由器上的 IP 组播操作。

总步骤

1. enable

2. show ip mroute [*group-address*]

3. show ip mroute active

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： Device> **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**show ip mroute** [*group-address*] 示例： Device# **show ip mroute 239.1.2.3** (*, 239.1.2.3), 00:17:56/00:03:02, RP 172.16.0.1, flags: S Incoming interface: Null, RPF nbr 0.0.0.0 Outgoing interface list: GigabitEthernet0/0/0,	对一个组或几个组，向着源的方向确认 RPF 邻居

	Forward/Sparse-Dense, 00:17:56/00:03:02 (10.0.0.1, 239.1.2.3), 00:15:34/00:03:28, flags: T Incoming interface: Serial1/0, RPF nbr 172.31.200.1 Outgoing interface list: GigabitEthernet0/0/0, Forward/Sparse-Dense, 00:15:34/00:03:02	
步骤 3	**show ip mroute active** [*kb/s*] 示例： Device# **show ip mroute active** Active IP Multicast Sources - sending >= 4 kbps Group: 239.1.2.3, (?) Source: 10.0.0.1 (?) Rate: 20 pps/4 kbps(1sec), 4 kbps(last 30 secs), 4 kbps(life avg)	显示向组发送数据的活动组播源信息。这条命令的输出信息可以提供活动源发送组播数据包的速率。**注释：**在默认情况下，若在使用命令 **show ip mroute** 时输入 **active** 关键字，系统会显示那些向组发送数据的速率大等于 4kb/s 的活动源。要查看以低速（也就是速率低于 4kb/s）向组发送数据的活动源，需要在 *kb/s* 部分将数值设置为 1。将这个参数的数值设置为 1 之后，系统就会显示以大于等于 1kb/s 的速率向组发送数据的活动源，这可以有效地显示出所有可能的活动源所发送的流量

在最后一跳路由器上查看 IP 组播的操作

用户可以在最后一跳路由器上输入下列命令，来查看这台路由器上的 IP 组播操作。

总步骤

1. **enable**
2. **show ip igmp groups**
3. **show ip pim rp mapping**
4. **show ip mroute**
5. **show ip interface** [*type number*]
6. **show ip mfib**
7. **show ip pim interface count**
8. **show ip mroute count**
9. **show ip mroute active** [*kb/s*]

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： Device> **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**show ip igmp groups**	在最后一跳路由器上查看 IGMP 成员身份。该信息可以证实这些组播组拥有与最后一跳路由器直连、

	示例： Device# **show ip igmp groups** IGMP Connected Group Membership Group Address Interface Uptime Expires Last Reporter 239.1.2.3 GigabitEthernet1/0/0 00:05:14 00:02:14 10.1.0.6 224.0.1.39 GigabitEthernet0/0/0 00:09:11 00:02:08 172.31.100.1	并且是通过 IGMP 学习到的接收方
步骤 3	**show ip pim rp mapping** 示例： Device# **show ip pim rp mapping** PIM Group-to-RP Mappings Group(s) 224.0.0.0/4 RP 172.16.0.1 (?), v2v1 Info source: 172.16.0.1 (?), elected via Auto-RP Uptime: 00:09:11, expires: 00:02:47	证实最后一跳路由器正确地创建了组-RP 映射。 注释：如果用户是在 PIM-SSM 网络中的最后一跳路由器，那就可以忽略这一步操作。命令 **show ip pim rp mapping** 在 PIM-SSM 网络中的路由器上是没有用的，因为 PIM-SSM 并不使用 RP。此外，如果配置正确的话，那么 PIM-SSM 组不会出现在命令 **show ip pim rp mapping** 的输出信息中
步骤 4	**show ip mroute** 示例： Device# **show ip mroute** (*, 239.1.2.3), 00:05:14/00:03:04, RP 172.16.0.1, flags: SJC Incoming interface: GigabitEthernet0/0/0, RPF nbr 172.31.100.1 Outgoing interface list: GigabitEthernet1/0, Forward/Sparse-Dense, 00:05:10/00:03:04 (10.0.0.1, 239.1.2.3),	在最后一跳路由器上查看其组播路由表是否正常

	00:02:49/00:03:29, flags: T Incoming interface: GigabitEthernet0/0/0, RPF nbr 172.31.100.1 Outgoing interface list: GigabitEthernet1/0, Forward/Sparse-Dense, 00:02:49/00:03:04 (*, 224.0.1.39), 00:10:05/stopped, RP 0.0.0.0, flags: DC Incoming interface: Null, RPF nbr 0.0.0.0 Outgoing interface list: GigabitEthernet1/0, Forward/Sparse-Dense, 00:05:15/00:00:00 GigabitEthernet0/0, Forward/Sparse-Dense, 00:10:05/00:00:00 (172.16.0.1, 224.0.1.39), 00:02:00/00:01:33, flags: PTX Incoming interface: GigabitEthernet0/0/0, RPF nbr 172.31.100.1	
步骤 5	**show ip interface** [*type number*] 示例： Device# **show ip interface GigabitEthernet 0/0/0** GigabitEthernet0/0 is up, line protocol is up Internet address is 172.31.100.2/24 Broadcast address is 255.255.255.255 Address determined by setup command MTU is 1500 bytes	在最后一跳路由器的出站接口上，查看是否为了实现最优性能而启用了组播快速交换。 **注释：**使用接口命令 **no ip mroute-cache** 可以禁用 IP 组播快速交换。如果禁用了 IP 组播快速交换，那么数据包就会通过进程交换的路径进行转发

	Helper address is not set Directed broadcast forwarding is disabled Multicast reserved groups joined: 224.0.0.1 224.0.0.22 224.0.0.13 224.0.0.5 224.0.0.6 Outgoing access list is not set Inbound access list is not set Proxy ARP is enabled Local Proxy ARP is disabled Security level is default Split horizon is enabled ICMP redirects are always sent ICMP unreachables are always sent ICMP mask replies are never sent IP fast switching is enabled IP fast switching on the same interface is disabled IP Flow switching is disabled IP CEF switching is disabled IP Fast switching turbo vector IP multicast fast switching is enabled IP multicast distributed fast switching is disabled IP route-cache flags are Fast Router Discovery is disabled IP output packet	

	accounting is disabled IP access violation accounting is disabled TCP/IP header compression is disabled RTP/IP header compression is disabled Policy routing is disabled Network address translation is disabled WCCP Redirect outbound is disabled WCCP Redirect inbound is disabled WCCP Redirect exclude is disabled BGP Policy Mapping is disabled	
步骤 6	**show ip mfib** 示例： Device# **show ip mfib**	显示 IP 组播转发信息库（MFIB）中的转发条目和接口
步骤 7	**show ip pim interface count** 示例： Device# **show ip pim interface count** State: * - Fast Switched, D - Distributed Fast Switched H - Hardware Switching Enabled Address Interface FS Mpackets In/Out 172.31.100.2 GigabitEthernet0/0/0 * 4122/0 10.1.0.1 GigabitEthernet1/0/0 * 0/3193	在最后一跳路由器上证实它正在转发组播流量
步骤 8	**show ip mroute count**	在最后一跳路由器上证实它正在转发组播流量

	示例： Device# **show ip mroute count** IP Multicast Statistics 6 routes using 4008 bytes of memory 3 groups, 1.00 average sources per group Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc) Group: 239.1.2.3, Source count: 1, Packets forwarded: 3165, Packets received: 3165 RP-tree: Forwarding: 0/0/0/0, Other: 0/0/0 Source: 10.0.0.1/32, Forwarding: 3165/20/28/4, Other: 0/0/0 Group: 224.0.1.39, Source count: 1, Packets forwarded: 21, Packets received: 120 Source: 172.16.0.1/32, Forwarding: 21/1/48/0, Other: 120/0/99 Group: 224.0.1.40, Source count: 1, Packets forwarded: 10, Packets received: 10 Source: 172.16.0.1/32, Forwarding: 10/1/48/0, Other:	
步骤 9	**show ip mroute active** [*kb/s*] 示例： Device# **show ip mroute active**	显示向组发送数据的活动组播源信息。这条命令的输出信息可以提供活动源发送组播数据包的速率。 注释：在默认情况下，若在使用命令 **show ip mroute** 时输入 **active** 关键字，系统会显示那些向组发送数

	Active IP Multicast Sources - sending >= 4 kbps Group: 239.1.2.3, (?) Source: 10.0.0.1 (?) Rate: 20 pps/4 kbps(1sec), 4 kbps(last 50 secs), 4 kbps(life avg)	据的速率大等于 4kb/s 的活动源。要查看以低速（也就是速率低于 4kb/s）向组发送数据的活动源，需要在 *kb/s* 部分将数值设置为 1。将这个参数的数值设置为 1 之后，系统就会显示以大于等于 1kb/s 的速率向组发送数据的活动源，这可以有效地显示出所有可能的活动源所发送的流量

使用启用了 PIM 的路由器来测试组播的可达性

如果用户管理的所有（启用了 PIM 的）路由器和接入服务器，都是一个组播组的成员，那么如果向这个组发起 ping 测试，所有路由器都会作出响应。因此，ping 是一个强大的管理和调试工具。

用户若要通过启用了 PIM 的路由器来测试 IP 组播的可达性，可以执行下面的操作。

配置路由器使其响应组播 ping

用户可以按照下面的步骤来配置路由器，使其响应组播 ping。用户可以在参与组播网络的路由器的所有接口上执行下面的操作：

总步骤

1. enable

2. configure terminal

3. interface *type number*

4. ip igmp join-group *group-address*

5. 在参与组播网络的路由器的每个接口上重复执行步骤3和步骤4

6. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *type number* 示例： `Device(config)#` **interface gigabitethernet 1/0/0**	进入接口配置模式。 在 *type* 和 *number* 部分，指定一个与主机直连或者面向主机的接口
步骤 4	**ip igmp join-group** *group-address*	（可选）配置路由器的一个接口，让其加入一个组播组。 为了执行这项示例的任务，用户需要在参与组播网

	示例： `Device(config-if)# ip igmp join-group 225.2.2.2`	络的路由器的所有接口上，针对 *group-address* 参数配置同一个组地址 注释：通过这种方式，路由器就会在接受组播数据包的同时，发送组播数据包。接受组播数据包可以防止路由器对其执行快速转发
步骤 5	在参与组播网络的路由器的每个接口上重复执行步骤 3 和步骤 4	——
步骤 6	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

向会响应组播 ping 的路由器发起 ping 测试

用户可以在一台路由器上执行下面的步骤，向那些会对组播 ping 作出响应的路由器发起 ping 测试。

总步骤

1. enable

2. ping *group-address*

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**ping** *group-address* 示例： `Device# ping 225.2.2.2`	向一个 IP 组播组地址发起 ping 测试。 如果能够 ping 通，标识该组地址当前工作正常

对 PIM 进行监控与排错

监控 PIM 信息

用户可以使用下表中的特权 EXEC 命令来监控 PIM 的配置。

表 42：PIM 监控命令

命令	目的
show ip pim all-vrfs tunnel [**tunnel** *tunnel_number* | **verbose**]	显示所有 VRF
show ip pim autorp	显示全局的 Auto-RP 信息
show ip pim boundary	显示用户配置在接口上的、管理作用范围的 IPv4 组播边界所过滤出来的组播路由信息
show ip pim interface	显示配置了 PIM（协议独立组播）的接口的

	相关信息
show ip pim neighbor	显示 PIM 邻居信息
show ip pim rp[*group-name* | *group-address*]	显示与稀疏模式组播组相关的 RP 路由器。所有软件版本都可以使用这条命令
show ip pim tunnel [**tunnel** | **verbose**]	显示关于 PIM（协议独立组播）隧道接口的信息
show ip pim vrf { **word** { **all-vrfs** | **autorp** | **boundary** | **bsr-router** | **interface** | **mdt** | **neighbor** | **rp** | **rp-hash** | **tunnel** } }	显示 VPN 路由/转发实例
show ip igmp groups detail	显示加入了特定组播源组的感兴趣客户端

监控 RP 映射与 BSR 信息

用户可以使用下表中的特权 EXEC 命令来查看组-RP 映射是否抑制。

表 43：RP 映射的监控命令

命令	目的
show ip pim rp [*hostname* or *IP address* | **mapping** [*hostname* or *IP address* | **elected** | **in-use**] | **metric** [*hostname* or *IP address*]]	显示所有可用的 RP 映射与度量值。这条命令的输出信息可以显示出设备是如何学习到 RP 的（是通过 BSR 还是通过 Auto-RP 机制）。 • （可选）在 *hostname* 部分，设置要显示 RP 的那个组的主机名 • （可选）在 *IP address* 部分，设置要显示 RP 的那个组的 IP 地址 • （可选）使用关键字 **mapping** 来显示 Inspur 设备了解的所有 RP 映射（包括静态配置的映射和通过 Auto-RP 学习到的映射） • （可选）使用关键字 **metric** 来显示 RP 的 RPF 度量值
show ip pim rp-hash *group*	显示特定组选得 RP。也就是说，在一台 PIMv2 路由器或者多层设备上，验证它的 RP 与 PIMv1 系统选择的 RP 相同。在 *group* 参数部分，应输入要查看其 RP 信息的那个组

用户可以使用下表中的特权 EXEC 命令来监控 BSR 信息。

表 44：BSR 的监控命令

命令	目的
show ip pim bsr	显示选举的 BSR 的信息
show ip pim bsr-router	显示关于 BSRv2 的信息

对 PIMv1 和 PIMv2 的互操作性问题进行排错

在对 PIMv1 和 PIMv2 之间的互操作性进行调试时，应该按顺序检查下面几项：

1　使用特权 EXEC 命令 **show ip pim rp-hash** 验证 RP 映射关系，确保所有系统在同一个组中都使用相同的 RP；

2　验证不同版本 DR 和 RP 的互操作性。确保 RP 与 DR 之间的交互是正常的（即使用注册停止消息进行响应，转发从注册消息中解封装的数据包）。

PIM 的配置示例

示例：启用 PIM 末节路由

在本例中，IP 组播路由已经启用。用户将交换机 A 的 PIM 上行链路端口 25 配置为了路由模式的上行链路端口，同时配置了命令 **spare-dense-mode**。在 VLAN 100 接口和接口 Gigabit Ethernet 20 上，用户启用了 PIM 末节路由：

```
Device(config)# ip multicast-routing distributed
Device(config)# interface GigabitEthernet3/0/25
Device(config-if)# no switchport
Device(config-if)# ip address 3.1.1.2 255.255.255.0
Device(config-if)# ip pim sparse-dense-mode
Device(config-if)# exit
Device(config)# interface vlan100
Device(config-if)# ip pim passive
Device(config-if)# exit
Device(config)# interface GigabitEthernet3/0/20
Device(config-if)# ip pim passive
Device(config-if)# exit
Device(config)# interface vlan100
Device(config-if)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip pim passive
Device(config-if)# exit
Device(config)# interface GigabitEthernet3/0/20
Device(config-if)# no switchport
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# ip pim passive
Device(config-if)# end
```

示例：查看 PIM 末节路由

要验证是否每个接口都启用了 PIM 末节，可以使用特权 EXEC 模式的命令 **show ip pim interface**：

```
Device# show ip pim interface
```

```
Address Interface Ver/ Nbr Query DR DR
Mode Count Intvl Prior
3.1.1.2 GigabitEthernet3/0/25 v2/SD 1 30 1 3.1.1.2
100.1.1.1 Vlan100 v2/P 0 30 1 100.1.1.1
10.1.1.1 GigabitEthernet3/0/20 v2/P 0 30 1 10.1.1.1
```

示例：手动给组播组分配一个 RP

这个示例显示了如何只针对组播组 225.2.2.2 将 RP 地址配置为 147.106.6.22：

```
Device(config)# access-list 1 permit 225.2.2.2 0.0.0.0
Device(config)# ip pim rp-address 147.106.6.22 1
```

示例：配置 Auto-RP

这个示例显示了如何将 RP 通告从所有启用了 PIM 的接口发送出去，同时将最大跳数设置为 31 跳。端口 1 的 IP 地址是 RP。编号为 5 的访问列表描述了这台设备充当的是哪个组的 RP：

```
Device(config)# ip pim send-rp-announce gigabitethernet1/0/1 scope 31 group-list 5
Device(config)# access-list 5 permit 224.0.0.0 15.255.255.255
```

示例：包含 Auto-RP 的稀疏模式

下面的示例配置了包含 Auto-RP 的稀疏模式：

```
ip multicast-routing
ip pim autorp listener
ip pim send-rp-announce Loopback0 scope 16 group-list 1
ip pim send-rp-discovery Loopback1 scope 16
no ip pim dm-fallback
access-list 1 permit 239.254.2.0 0.0.0.255
access-list 1 permit 239.254.3.0 0.0.0.255
.
.
.
access-list 10 permit 224.0.1.39
access-list 10 permit 224.0.1.40
access-list 10 permit 239.254.2.0 0.0.0.255
access-list 10 permit 239.254.3.0 0.0.0.255
```

示例：定义 IP 组播边界来拒绝 Auto-RP 信息

这个示例显示了拒绝 Auto-RP 信息的 IP 组播边界（部分）配置：

```
Device(config)# access-list 1 deny 224.0.1.39
Device(config)# access-list 1 deny 224.0.1.40
```

```
Device(config)# access-list 1 permit all
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip multicast boundary 1
```

示例：过滤入站 RP 通告消息

这个示例显示了 Auto-RP 映射代理的配置示例，用户使用这个 Auto-RP 映射代理来防止未经授权的候选 RP 发送的候选 RP 通告被设备接受：

```
Device(config)# ip pim rp-announce-filter rp-list 10 group-list 20
Device(config)# access-list 10 permit host 172.16.5.1
Device(config)# access-list 10 permit host 172.16.2.1
Device(config)# access-list 20 deny 239.0.0.0 0.0.255.255
Device(config)# access-list 20 permit 224.0.0.0 15.255.255.255
```

这个映射代理只接受两台设备（172.16.5.1 和 172.16.2.1）发来的候选 RP 通告。这个映射代理只接受这两台设备发送给组范围在 224.0.0.0 到 239.255.255.255 之间的组的候选 RP 通告。映射代理不支持网络中其他设备发送的候选 RP 通告。此外，如果通告发往 239.1.1.1 到 239.255.255.255 这个范围的组，那么即使是 172.16.5.1 和 172.16.2.1 发送的候选 RP 通告，映射代理也不支持。这个范围就是管理作用范围的地址范围。

示例：防止加入消息被发送给非法 RP

如果所有接口都处于稀疏模式，那么用户可以使用默认配置的 RP 来支持两个知名的（well-known）组 224.0.1.39 和 224.0.1.40。Auto-RP 会使用这两个指明的组来收集和分发 RP 信息。此时，如果用户配置了命令 **ip pim accept-rp auto-rp**，那么用户就必须按照下面所示的方法配置另一条命令 **ip pim accept-rp** 来接受 RP：

```
Device(config)# ip pim accept-rp 172.10.20.1 1
Device(config)# access-list 1 permit 224.0.1.39
Device(config)# access-list 1 permit 224.0.1.40
```

示例：配置候选 BSR

这个示例显示了如何配置候选 BSR，同时以 IP 地址 172.21.24.18 作为通告的 BSR 地址，用 30 位作为散列码长度，并且将优先级配置为 10：

```
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# ip address 172.21.24.18 255.255.255.0
Device(config-if)# ip pim sparse-dense-mode
Device(config-if)# ip pim bsr-candidate gigabitethernet1/0/2 30 10
```

示例：配置候选 RP

这个示例显示了如何配置设备，让它在向 BSR 发送通告时，将自己通告为 PIM 域中的候选 RP。本例用编号为 4 的标准访问列表设置了 RP 对应的组前缀（RP 的地址是通过端口的编号

来标识的）。这个 RP 负责前缀为 239 的组播组：

```
Device(config)# ip pim rp-candidate gigabitethernet1/0/2 group-list 4
Device(config)# access-list 4 permit 239.0.0.0 0.255.255.255
```

其他参考资料

相关文档

相关主题	文档名
如需了解本章所述命令的完整语法结构及使用信息	《IP 组播路由命令参考手册（Inspur 6650 交换机）》
IGMP Helper 的完整语法结构及使用信息	《IP 组播路由命令参考手册（Inspur 6650 交换机）》
组播源发现协议（MSDP）	《IP 路由：协议独立组播配置指南，Inspur INOS XE 3E 版（Inspur 6650 交换机）》
增强型内部网关路由协议（EIGRP）末节路由	《IP 路由：EIGRP 配置指南，Inspur INOS XE 3E 版（Inspur 6650 交换机）》
开放式最短路径优先（OSPF）末节路由	《IP 路由：OSPF 配置指南，Inspur INOS XE 3E 版（Inspur 6650 交换机）》
Inspur INOS 命令	《Inspur INOS 主命令列表，所有版本》
Inspur INOS IP SLA 命令	《Inspur INOS IP 组播命令参考手册》

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com

标准与 RFC

标准/RFC	标题
PIM 定义在 RFC 4601 和右侧的 IETF（互联网工程任务组）互联网草案中	• *协议独立组播（PIM）：动机与架构* • *协议独立组播（PIM）：密集模式协议标准* • *协议独立组播（PIM）：稀疏模式协议标准* • *draft-ietf-idmr-igmp-v2-06.txt，互联网组管理协议，第 2 版* • *draft-ietf-pim-v2-dm-03.txt ，PIM 第 2 版密集模式*

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产	http://www.icntnetworks.com

品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	

配置接口特征的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

配置 MSDP

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于使用 MSDP 来互连多个 PIM-SM 域的信息

使用 MSDP 来互连多个 PIM-SM 域的好处

- 可以让一个汇集点（RP）动态发现域外的活动源；
- 让建立多域之间的组播分发树的方式更容易管理；

MSDP 是一种用来连接多个 PIM-SM 域的机制。MDSP 的作用在于发现其他 PIM 域中的组播源。MDSP 的主要优势在于，它可以减少多个 PIM-SM 域忽略的复杂性，因为 MSDP 可以让

PIM-SM 域使用一种域间源树（而不是共享树）。当用户在网络中配置了 MSDP 时，RP 就会与其他域中的 RP 交互源信息。对于那些正在向树中拥有接收方的组发送数据源，RP 会加入域间的源树。RP 之所以可以做到这一点，因为 RP 是其域内共享树的根，而这棵共享树有很多叶网络通向有活动接收方的域内各点。当最后一跳设备（在源发送的组播数据包沿着共享树到达最后一跳设备时）学习到了 PIM-SM 域之外的源时，它就可以向源发送加入消息，并且加入域间的源树了。

注释： 如果 RP 在特定组没有共享树，或者其共享树的出站接口列表为空，那么它就不会向另一个域中的源发送加入消息。

在 MSDP 启用之后，PIM-SM 域中的一个 RP 就会与其他域中启用 MSDP 的设备维护 MDP 对等体关系。这种对等体关系是通过一条 TCP 连接建立起来的，双方主要交换的是向组播组发送消息的源列表。使用点到点的 TCP 对等体表示每个对等体都需要由用户进行配置，这一点 MSDP 和 BGP 是一样的。此外，RP 之间的 TCP 连接是通过底层的路由系统建立起来的。接收方 RP 会使用源列表来建立源路径。如果组播源对于一个有接收方的域感兴趣，那么组播数据就会通过一跳由 PIM-SM 提供的普通源树建立机制进行转发。MSDP 的作用是通告向组发送数据的源。而这些通告消息一定是以域的 RP 作为始发点的。

下图显示了 MSDP 对等体间的 MSDP 操作。PIM 会以 MSDP 作为标准机制，来向域的 RP 注册源。

注释： 虽然下图和下面的示例在配置中使用得到都是路由器，但用户可以使用任意设备（路由器和交换机）。

图 29：RP 对等体之间运行的 MSDP

PIM sparse mode domain B	PIM 稀疏模式 B 域
RP+MSDP peer	RP+MSDP 对等体
PIM sparse mode domain C	PIM 稀疏模式 C 域
RP+MSDP peer	RP+MSDP 对等体
RP+MSDP peer	RP+MSDP 对等体
Register	注册
Multicast	组播
Source	源
PIM sparse mode domain A	PIM 稀疏模式 A 域
TCP connection	TCP 连接
Peer RPF flooding	对等体 RPF 泛洪
RP+MSDP peer	RP+MSDP 对等体
Receiver	接收方
PIM sparse mode domain D	PIM 稀疏模式 D 域
(S,G) Join	(S,G)加入消息

在实施了 MSDP 之后，网络会按顺序发生下面的情况：

1 当一台 PIM 指定设备（DR）如图所示在 RP 上注册了一个源时，RP 向所有的 MSDP 对等体发送了一个源-活动（SA）消息；

注释： DR 只会（在源进入活动状态时）向每个 RP 发送一次封装过的数据包。如果源超时，那么当它再次进入活动状态时，这个进程就会重复。这种做法不同于周期性发送（包含

了所有注册到原始 RP 上的源的）SA 消息。这些 SA 消息都是 MSDP 控制数据包，因此并不会包含活动源发送的封装的数据。

1　SA 消息标识出了源地址、源发送消息的目的组，以及地址或 RP 的发起方 ID（originator ID）；

2　每个接收到 SA 消息 MSDP 对等体都会向下游的所有对等体泛洪 SA 消息。在有些情况下（如图中 B 域和 C 域中的 RP），RP 可能会从多台 MSDP 对等体那里接收到 SA 消息的副本。为了避免出现环路，RP 会查询 BGP 下一跳数据库，来判断去往该 SA 消息发起方的下一跳设备。如果用户配置了 MBGP 和单播 MGP，那么设备会首先校验 MBGP，然后再校验单播 BGP。这个下一跳邻居是发起方的 RPF 对等体。如果设备是从连接 RPF 对等体的接口之外的其他接口接收到了发起方发来的 SA 消息，那么设备就会丢弃这些消息。因此，SA 消息泛洪进程也称为对等体 RPF 泛洪。由于这种对等体 RPF 泛洪的机制，因此 BGP 或 MBGP 必须与 MSDP 结合起来使用；

1　当 RP 接收到一条 SA 消息时，它会查看在组的(*,G)出站接口列表中是否有自己的接口，以此判断自己的域中是否有该消息所通告组的成员设备。如果没有组成员，那么 RP 就不会执行任何操作。如果有组成员，那么 RP 就会向源发送一条(S,G)加入消息。于是，一棵跨越系统系统边界通向 RP 的域间源树也就构建了起来。当组播数据包到达 RP 时，它们就会在 RP 域中验证自己的共享树转发给组成员。成员的 DR 可以选择使用标准的 PIM-SM 进程来加入通往源的汇集点树（RPT）；

2　只要源还在向组发送数据包，那么发起的 RP 就会继续周期性地每 60 秒发送一次（S,G）状态的 SA 消息。当 RP 接收到 SA 消息时，它会将 SA 消息缓存起来。比如，我们假设 RP 从始发 RP 10.5.4.3 那里接收到了一条 SA 消息(172.16.5.4, 228.1.2.3)。那么这台 RP 就会查看自己的组播路由表，然后发现那里找不到组 228.1.2.3 的活动成员，于是它就向这条 SA 消息向着 10.5.4.3 的下游方向转发给自己的对等体。如果域中一台主机接下来向 RP 发送了一跳加入组 228.1.2.3 的消息，那么 RP 就会将面向这台主机的接口添加到自己(*,224.1.2.3)条目的出站接口列表当中。由于 RP 会缓存 SA 消息，因此设备会有一条(172.16.5.4, 228.1.2.3)的条目，同时也可以在主机请求加入的时候加入源树。

注释：　在所有当前和支持的软件版本中，对 MSDP SA 消息执行缓存是强制性的，无法手动启用或禁用。在默认情况下，当用户配置 MSDP 对等体的时候，命令 **ip multicast cache-sa-state** 就会自动添加到设备的运行配置当中。

MSDP 消息类型

有 4 种基本的 MSDP 消息类型，它们都各自使用了不同的 Type（类型）、Length（长度）和 Value（TLV）数据格式。

SA 消息

SA 消息的作用是通告域中的活动源。此外，这些 SA 消息中可能会包含源发送的初始组播数据包。

SA 消息中包含发起 RP 的 IP 地址，以及一个或多个通告的(S,G)对。此外，SA 消息中可能会包含封装的数据包。

注释：　要想了解更多关于 SA 消息，可以查看 SA 消息发起接收与处理

SA 请求消息

SA 请求消息的作用是请求特定组的活动源列表。这些消息会发送给 MSDP 的 SA 缓存。在 SA 缓存中，设备会维护一个活动(S,G)对的列表。用户可以使用 SA 请求消息来请求一个组的活动源列表，而不是被动地等待 60 秒直至发起 RP 重新通告了组中的所有活动源，这可以达

到减少加入延迟的效果。

SA 响应消息

SA 响应消息是 MSDP 对等体为了响应 SA 请求消息而发送的消息。SA 响应消息中包含发起 RP 的 IP 地址，以及（缓存中保存的）发起 RP 域中的一条或多条活动源的(S,G)对。

保活（keepalive）消息

保活消息每 60 秒发送以此，目的是保证 MSDP 会话处于活动状态。如果超过 75 秒没有接收到保活消息或 SA 消息，那么 MSDP 会话就会重置。

SA 消息发起接收与处理

在这一节中，我们会描述详细 SA 消息的发起、接收与处理。

SA 消息的发起

当本地 PIM-SM 域中有新的源进入活动状态时，（假设用户配置了 MSDP）RP 就会触发 SA 消息。本地源是与 RP 直连的源，或者是向 RP 注册的第一跳 DR。RP 只会为其 PIM-SM 域中的本地源发送 SA 消息。也就是说，RP 只会为其注册了的本地源发送 SA 消息。

注释： 本地源是用 RP 上的(S,G)组播路由条目所设置的 A 标记进行标识的（用户可以用命令 **show ip mroute** 来对此进行查看）。这个标记表示这个源是从 RP 向 MSDP 对等体发送通告的候选设备。

当本地 PIM-SM 域中有一个源时，RP 上就会创建出(S,G)状态。在 RP 接收到注册消息，或者从直连源那里接收到第一条(S,G)数据包时，它就会检测到新的源。源发送的初始组播数据包（无论是封装在注册消息中，还是从直连的源那里接收到）都会被封装在初始的 SA 消息当中。

SA 消息的接收

只有从通向发起方的最佳路径中的 MSDP RPF 对等体那里接收到的 SA 消息，设备才会接受。从其他 MSDP 对等体那里接收到的 MSDP 则一定会被忽略，否则就会形成 SA 环路。

要想给到达的 SA 消息选择合理的 MSDP RPF 对等体，需要掌握 MSDP 的拓扑。但 MSDP 并不会用路由更新的格式来分发拓扑信息。MSDP 会以（M）BGP 路由数据作为 SA PRF 校验机制中最理想的 MSDP 近似拓扑，来推断这些信息。因此，MSDP 拓扑必须与 BGP 对等体拓扑基本相同。除了个别例外之外（比如默认的 MSDP 对等体和 MSDP 全互联组[MSDP mesh group]中的 MSDP 对等体），MSDP 对等体同时也基本应该就是（M）BGP 对等体。

RPF 校验规则如何应用于 SA 消息

设备在对 SA 消息执行 RPF 校验时，使用的规则与 MSDP 对等体之间的 BGP 对等体关系有关：

- 规则 1：若发送方 MSDP 对等体同时也是内部（M）BGP 对等体；
- 规则 2：若发送方 MSDP 对等体同时也是外部（M）BGP 对等体；
- 规则 3：若发送方 MSDP 对等体不是（M）BGP 对等体。

在下列情况下，设备不会执行 RPF 校验：

- 发送方 MSDP 对等体是唯一的 MSDP 对等体，若用户只配置了一台 MSDP 对等体或者只有一台默认 MSDP 对等体，就属于这种情况；
- 发送方 MSDP 对等体是一个全互联组的成员；
- 发送方 MSDP 对等体地址是 SA 消息中包含的 RP 地址。

软件如何判断 RPF 校验的应用规则

软件会采用下面的逻辑来判断在执行 RPF 校验时应用何种 RPF 规则：

- 找到与发送方 MSDP 对等体 IP 地址相同的那个（M）BGP 邻居：

- 如果匹配的（M）BGP 邻居是一台内部 BGP（iBGP）对等体，则应用规则 1；
- 如果匹配的（M）BGP 邻居是一台外部 BGP（iBGP）对等体，则应用规则 2；
- 如果找不到匹配，则应用规则 3；

RPF 校验规则的选择方式旨在表示：在设备上配置 MSDP 对等体的 IP 地址，必须与在同一台设备上配置（M）BGP 对等体的那个 IP 地址相匹配。

在 MSDP 中，RPF 校验 SA 消息的规则 1

在 MSDP 中，如果发送方 MSDP 同时也是一台 i（M）BGP 对等体，那么设备就会执行 RPF 校验的规则 1。若执行规则 1，RPF 校验进程为：

1　对等体查询 BGP 组播路由信息库（MRIB），以寻找去往发送这条 SA 消息的那个 RP 的最佳路径。如果 MRIB 中没有找到路径，那么对等体就会查询单播路由信息库（URIB）。如果还是没有找到路径，则 RPF 校验失效；

2　如果之前的查询成功（也就是找到了最佳路径），那么对等体就会判断这条最佳路径的 BGP 邻居地址。这个地址应该是通过 BGP 更新消息向对等体发送这条路径的那个 BGP 邻居的地址。

注释：　BGP 邻居地址与路径中的下一跳地址并不相同。因为 i（M）BGP 对等体并不会更新路径的下一跳属性，所以下一跳地址往往并不是发送这条路径的那个 BGP 对等体的地址。

注释：　BGP 邻居地址未必是向对等体发送这条路径的那台设备的 BGP ID。

1　如果发送方 MSDP 对等体的 IP 地址是 BGP 邻居的地址（也就是发送这条路径的那个 BGP 对等体的地址），则 RPF 校验成功；否则 RPF 校验失败。

在 MSDP 中，RPF 校验规则 1 的含义

MSDP 拓扑必须能够反映（M）BGP 拓扑。总地来说，只要两台设备之间有一条 i（M）BGP 对等体连接，那就应该配置一条 MSDP 对等体连接。说得更具体一点，那就是远端 MSDP 对等体连接的 IP 地址必须与远端 i（M）BGP 对等体连接的 IP 地址相同。地址必须相同的原因在于，一个自治系统内的 i（M）BGP 对等体之间的 BGP 拓扑，并不是通过 AS path 描述的。假如 i（M）BGP 对等体永远会在向另一个 i（M）BGP 对等体发送更新时更新下一跳地址，那么对等体也许真的可以依靠下一跳地址来描述 i（M）BGP 拓扑（及 MSDP 拓扑）。不过，由于 i（M）BGP 对等体默认的操作是并不更新下一跳地址，因此对等体无法依靠下一跳地址来描述 i（M）BGP 拓扑（及 MSDP 拓扑）。所以，i（M）BGP 对等体会使用发送路径的那台 i（M）BGP 对等体的地址，来描述自制系统中的 i（M）BGP 拓扑（及 MSDP 拓扑）。

提示：　在配置 MSDP 对等体地址的时候，要确保 i（M）BGP 对等体和 MSDP 对等体的地址是相同的。

在 MSDP 中，RPF 校验 SA 消息的规则 2

在 MSDP 中，如果发送方 MSDP 同时也是一台 e（M）BGP 对等体，那么设备就会执行 RPF 校验的规则 1。若执行规则 1，RPF 校验进程为：

1　对等体查询 BGP MRIB 来寻找去往发送这条 SA 消息的那个 RP 的最佳路径。如果 MRIB 中没有找到路径，那么对等体就会查询 URIB。如果还是没有找到路径，则 RPF 校验失效；

2　如果之前的查询成功（也就是找到了最佳路径），那么对等体就会判断这条最佳路径。如果去往 RP 的最佳路径中，第一个自治系统与 e（M）BGP 对等体（同时也是发送方 MSDP 对等体）的自治系统相同，则 RPF 校验成功；否则 RPF 校验失败。

在 MSDP 中，RPF 校验规则 2 的含义

MSDP 拓扑必须能够反映（M）BGP 拓扑。总地来说，只要两台设备之间有一条 e（M）BGP 对等体连接，那就应该配置一条 MSDP 对等体连接。与规则 1 相对的是，远端 MSDP 对等体连接的 IP 地址不需要与远端 e（M）BGP 对等体连接的 IP 地址相同。这个地址不需要相同的理由在于，两台 e（M）BGP 对等体之间的 BGP 拓扑并不是通过 AS path 来描述的。

在 MSDP 中，RPF 校验 SA 消息的规则 3

在 MSDP 中，如果发送方 MSDP 根本就不是（M）BGP 对等体，那么设备就会执行 RPF 校验的规则 3。若执行规则 3，RPF 校验进程为：

1　对等体查询 BGP MRIB 来寻找去往发送这条 SA 消息的那个 RP 的最佳路径。如果 MRIB 中没有找到路径，那么对等体就会查询 URIB。如果还是没有找到路径，则 RPF 校验失效；

2　如果之前的查询成功（也就是找到了去往发送 SA 消息的那台 RP 的最佳路径），那么对等体就会查找 BGPMRIB 来寻找去往发送 SA 消息的那台 MSDP 对等体的最佳路径。如果 MRIB 中没有找到路径，那么对等体就会查找 URIB。如果仍然没有找到路径，则 RPF 校验失败。

注释：　发送 SA 的 MSDP 对等体所在的自治系统，是始发的自治系统，也就是去往 MSDP 对等体的那条 AS path 中的最后一个自治系统。

1　如果去往 RP 的最佳路径中，第一个自治系统与发送方 MSDP 对等体的自治系统相同，则 RPF 校验成功；否则 RPF 校验失败。

SA 消息的处理

在 MSDP 对等体处理 SA 消息时，会执行下面的步骤：

1　对等体会使用 SA 消息中，(S,G)对中的组地址 G，在组播路由表中寻找相关的(*,G)条目。如果找到了(*,G)条目，且其出站接口列表非空，那么在 PIM-SM 域中，SA 消息中通告的源就由活动的接收方；

2　于是，MSDP 就会针对通告的源创建一个(S,G)条目；

3　如果没有(S,G)条目，那么 MSDP 对等体会离开向源发送一条(S,G)加入消息，以加入这棵源树；

4　接下来，对等体会将 SA 消息泛洪给所有 MSDP 对等体，但不包括：

- 从其接收到 SA 消息的那个 MSDP 对等体；
- 与这台设备处于同一个 MSDP 全互联组中的所有 MSDP 对等体（如果这个对等体是一个全互联组的成员）。

注释：　SA 消息会保存在设备 SA 缓存本地。

MSDP 对等体

MSDP 也会像 BGP 那样，与其他对等体之间建立邻居关系。MSDP 对等体会使用 TCP 端口 639 来建立连接。IP 地址比较低的对等体会在打开 TCP 连接时扮演主动的角色。IP 地址较高的对等体会在 LISTEN 状态中等待对端建立连接。MSDP 对等体会每 60 秒发送一次保活消息，以防止会话老化。如果连续 75 秒没有接收到保活消息或数据，TCP 链路就会重置。

MSDP MD5 密码认证

MSDP MD5 密码认证特性支持通过摘要消息 5（MD5）特征来保护两个 MSDP 对等体之间的 TCP 连接。这项特性可以保护 MSDP，使其免遭恶意用户将欺骗的 TCP 数据段注入 TCP 连接流中并以此给网络带来的威胁。

MSDP MD5 密码认证是如何工作的

MSDP MD5 密码认证特性是参考 RFC 2385 标准开发的，它的作用是对 MSDP 对等体之间通过 TCP 链路发送的每个数据段进行验证。用户可以使用命令 **ip msdp password peer** 来针对两台 MSDP 对等体之间的 TCP 连接启用 MD5 认证。当两台 MSDP 对等体之间启用了 MD5 认证之后，对等体之间通过 TCP 链路发送的所有数据段都会进行验证。MSDP 对等体之间的

MD5 认证必须配置相同的密码；否则，它们之间的连接就无法建立起来。配置 MD5 认证会让 Inspur INOS 软件来对 TCP 连接上的每个数据段都创建出 MD5 摘要，并且进行验证。

MSDP MD5 密码认证的好处

- 保护两个 MSDP 对等体之间的 TCP 连接。这项特性可以保护 MSDP，使其免遭恶意用户将欺骗的 TCP 数据段注入 TCP 连接流中并以此给网络带来的威胁；
- 使用行业标准的 MD5 算法来提供通信的可靠性和安全性。

SA 消息的限制

用户可以使用命令 **ip msdp sa-limit** 来针对特定 MSDP 对等体限制设备可以接受的 SA 消息数量。在配置了命令 **ip msdp sa-limit** 之后，设备就会在 SA 缓存中针对每个对等体维护 SA 消息的计数值，如果一台对等体达到了用户配置的 SA 消息限制数量，那么设备就会忽略后续的消息。

命令 **ip msdp sa-limit** 是一种保护 MSDP 设备免遭拒绝服务攻击（DoS）的方式。我们推荐用户在设备上针对所有 MSDP 对等体都配置 SA 消息限制。在与末节 MSDP 区域的对等体之间，应该配置一个相对更低的 SA 限制（例如，有的对等体可能有一些下游对等体设备，但它们不会通过 Internet 的其他部分来转发 SA 消息）。对于那些会通过 Internet 来转发 SA 的 MSDP 对等体，用户则应该配置比较高的 SA 限制数量。

MSDP 保活和抑制时间时间间隔

用户可以使用命令 **ip msdp keepalive** 来调整 MSDP 对等体发送保活消息的时间间隔，以及 MSDP 对等体会在宣告对端宕机之前，会等待多长时间来接收对端对等体发来的保活消息。一旦 MSDP 对等体会话建立起来，连接的两端都会发送一个保活消息，并且设置一个保活计时器。如果保持计时器超时，那么本地 MSDP 对等体就会发送一条保活小，并且重置自己的保活计时器；这个时间间隔称为保活时间间隔。参数 *keepalive-interval* 的作用是调整保活消息的发送时间间隔。当对等体启动时，其保活计时器就会重置为用户通过 *keepalive-interval* 设置的参数。每当一台 MSDP 保活消息发送给对等体时，这台对等体都会将自己的保活计时器重置为用户通过 *keepalive-interval* 设置的参数，在计时器过期时，计时器也会重置。当 MSDP 对等体关系关闭时，保活计时器就会被删除。在默认情况下，保活计时器会被设置为 60 秒。

注释： 用户通过 *keepalive-interval* 参数设置的值必须小于用户通过 *holdtime-interval* 参数设置的值，且必须最少为 1 秒。

只要对等体之间的连接建立起来，抑制时间计时器就会初始化为用户通过 *holdtime-interval* 参数设置的值。而且每当设备接收到一跳 MSDP 保活消息时，抑制时间计时器都会重置为用户通过 *holdtime-interval* 参数设置的值。当 MSDP 对等体关系关闭时，抑制时间计时器就会被删除。在默认情况下，保活计时器会被设置为 75 秒。

用可以设置通过参数 *keepalive-interval* 来调整 MSDP 对等体会在宣告对端宕机之前，会等待多长时间来接收对端对等体发来的保活消息。

MSDP 连接重试时间间隔

用户可以调整所有 MSDP 对等体在对等体会话重置之后，尝试重新建立对等体会话之前，会等待多长时间。这个时间间隔称为连接重试时间间隔。在默认情况下，MSDP 对等体在对等体会话重置之后，尝试重新建立对等体会话之前，会等待 30 秒的时间。修改后的连接重试时间间隔会应用于这台设备的所有 MSDP 对等体。

默认 MSDP 对等体

出于对冗余的需求，末节自治系统可能也希望能够与多台 RP 之间建立 MSDP 对等体关系。例如，设备不能接受来自多台默认对等体的 SA 消息，因为没有 RPF 校验机制。所以，设备只能接受一台对等体发来的 SA 消息。如果那台对等体出现了故障，那么对等体就会接受另一台对等体发来的 SA 消息。当然，这里有一个前提，那就是两台默认对等体发送的是相同的 SA 消息。

这张图显示了一个使用了默认 MSDP 对等体的环境。在图中，拥有设备 B 的客户通过两个互联网服务提供商（ISP）连接到了 Internet，其中一个 ISP 拥有设备 A，另一个 ISP 拥有设备 C。它们之间没有运行 BGP 或 MBGP。为了让客户学习到 ISP 域内或其他域中的源，设备 B 将设备 A 标识为了自己的默认 MSDP 对等体。设备 B 同时向设备 A 和设备 C 通告了 SA 消息，但只接受设备 A 或只接受设备 C 发来的 SA 消息。如果设备 A 是配置中的第一台默认对等体，那么只要它运行正常，设备就会使用设备 A。只有在设备 A 无法正常工作时，设备 B 才会接受从设备 C 发来的 SA 消息。

ISP 也可能会使用前缀列表来定义它会接受客户设备发来的哪些前缀。客户端会定义多台默认对等体，每台对等体关联有一个或多个前缀。

客户端有两个 ISP 可以使用。客户端同时将这两个 ISP 都定义为了默认对等体。只要配置中设置的第一个默认对等体运行正常，它就会称为默认对等体，设备也只会接受它发来过来的所有 SA 消息。

注释： 虽然下图和下面的示例在配置中使用得到都是路由器，但用户可以使用任意设备（路由器和交换机）。

图 30：默认 MSDP 对等体的环境

Device A	设备 A
Default MSDP peer	默认 MSDP 对等体
ISP A PIM domain	ISP A 的 PIM 域
Device C	设备 C
Default MSDP peer	默认 MSDP 对等体
ISP C PIM domain	ISP C 的 PIM 域
Device B	设备 B
Default MSDP peer	默认 MSDP 对等体
Customer PIM domain	客户的 PIM 域

设备 B 同时向设备 A 和设备 C 通告了 SA 消息，但只接受设备 A 或只接受设备 C 发来的 SA 消息。如果设备 A 是配置中的第一台默认对等体，那么只要它运行正常，设备就会使用设备 A。只有在设备 A 无法正常工作时，设备 B 才会接受从设备 C 发来的 SA 消息。

如果用户设置了一个前缀列表，那么对等体只会成为列表中前缀的默认对等体。如果用户给

每台默认对等体关联一个前缀列表的话，那么一台设备可以有多个活动的默认对等体。如果没有配置前缀列表，同时又配置了多台默认对等体，那么只要其中的第一台对等体与设备之间保持连通，并且这台设备没有宕机，那么它就是唯一活动的默认对等体。如果用户配置的第一台设备宕机，或者对等体连接断开，那么用户配置的第二台设备就会成为活动的默认对等体，以此类推。

MSDP 全互联组

MSDP 全互联组是一组相互之间通过全互联方式建立连接的 MSDP 设备。换句话说，组中的每台 MSDP 对等体都必须与组中的其他每一台 MSDP 对等体之间拥有对等体关系（即建立连接）。当用户给一组 MSDP 对等体配置成了一个 MSDP 全互联组时，SA 消息的泛洪就会减少。因为当组中的一台 MSDP 对等体接收到了组中其他 MSDP 对等体发来的 SA 消息，它就会默认这个 SA 消息也发送给了组中的其他 MSDP 对等体。因此，接收到消息的 MSDP 对等体没有必要将 SA 消息泛洪给组中的其他 MSDP 对等体。

MSDP 全互联组的好处

- 优化 SA 泛洪：在组中有两个或多个对等体时，SA 全互联组对于优化 SA 泛洪格外适用；
- 减少 Internet 中的 SA 流量：在使用 MSDP 全互联组时，SA 消息不会被泛洪给其他的全互联组对等体；
- 不需要对到达的 SA 消息执行 RPF 校验：在配置了 MSDP 全互联组之后，全互联组的对等体总是会接受 SA 消息。

SA 发起过滤器

在默认情况下，运行 MSDP 的 RP 会在 SA 消息中通告向所有由它充当 RP 的本地源。于是，注册在 RP 上的本地源就可以在 SA 消息中进行通告，但这种通告方式有时并不必要。比如，如果 PIM-SM 域内的源使用的是私有地址（比如，网络 10.0.0.0/8），用户就应该配置一个 SA 发起过滤器，来限制这些地址通过 Internet 通告给其他 MSDP 对等体。

要控制 SA 消息中可以通告哪些源，用户可以在 RP 上配置 SA 发起过滤器。通过创建 SA 发起过滤器，用户可以按照如下方式控制 SA 消息中通告的源：

- 用户可以配置 RP 来防止设备在 SA 消息中通告本地源。设备还是会按照正常的方式转发其他 MSDP 对等体发来的 SA 消息，只是它通告的 SA 消息中不会再包含本地源了；
- 用户可以让设备发起的 SA 消息中只包含一部分本地源，即那些向与扩展访问列表中定义的(S,G)对相匹配的组发送消息的本地源。其他本地源都不会在 SA 消息中进行通告；
- 用户可以让设备发起的 SA 消息中只包含一部分本地源，即那些向与 AS-path 访问列表中定义的 AS-path 相匹配的组发送消息的本地源。其他本地源都不会在 SA 消息中进行通告；
- 用户可以让设备发起的 SA 消息中只包含与 route map 中定义的标准相匹配的本地源。其他本地源都不会在 SA 消息中进行通告；
- 用户可以配置一个 SA 发起过滤器，其中包含一个扩展访问列表、一个 AS-path 访问列表和一个 route map，或者这三者的任意组合。此时，设备通告的 SA 消息中，包含的本地源必须匹配所有定义的条件。

在 MSDP 中使用出站过滤列表

在默认情况下，所有启用了 MSDP 的设备都会将自己接收到的所有 SA 消息转发给自己所有的 MSDP 对等体。但用户可以通过创建出站过滤列表，来防止设备将 SA 消息转发给自己的 MSDP 对等体。出站过滤列表会应用于所有的 SA 消息，无论是本地发起的 SA 消息，还是从其他 MSDP 对等体那里接收到的 SA 消息，而 SA 发起过滤器则只会应用于那些本地发起的 SA 消息。如需了解针对本地设备发起的 MSDP SA 消息启用过滤功能，可以参考控制 RP 为本地源发起的 SA 消息部分。

用户可以通过创建出站过滤器列表，来控制设备发送给对等体的 SA 消息，具体方法如下：

- 用户可以让设备不向某台 MSDP 对等体转发自己的 SA 消息，以此来过滤所有转发给某个特定 MSDP 对等体的出站 SA 消息；
- 用户可以让设备只向某台 MSDP 对等体转发那些与扩展访问列表定义的(S,G)对相匹配的 SA 消息，以此来根据扩展访问列表中定义的(S,G)对，过滤一部分转发给某个特定 MSDP 对等体的出站 SA 消息。让不满足访问列表条件的 SA 消息不再转发给这个 MSDP 对等体；
- 用户可以让设备只转发那些与 route map 定义的标准相匹配的 SA 消息，以此来根据 route map 中定义的标准，过滤一部分转发给某个特定 MSDP 对等体的出站 SA 消息。让不满足 route map 条件的 SA 消息不再转发给这个 MSDP 对等体；
- 用户可以让设备只根据 SA 的来源来过滤 SA 消息（即使 SA 消息已经经由一台或多台 MSDP 对等体进行了转发），即根据 SA 消息中通告的 RP 地址来过滤来自某个对等体的 SA 消息。其他 SA 消息不再转发给这个 MSDP 对等体；
- 用户可以配置一个出站过滤列表，其中包含一个扩展访问列表、一个 route map，以及一个 RP 访问列表或一个 RP route map。此时，MSDP 对等体必须保证所有用户定义的条件都满足，才会转发出站 SA 消息。

注意： 随意过滤 SA 消息可能会让下游 MSDP 对等体无法接收到关于合法活动源的 SA 消息。因此，用户在配置过滤策略时应该谨慎。一般来说，出站过滤列表只应该用来拒绝那些不应该转发的源，譬如那些使用私有地址的源。

在 MSDP 中使用入站过滤列表

在默认情况下，所有启用了 MSDP 的设备都会将自己接收到的所有 SA 消息转发给自己所有的 MSDP 对等体。但用户可以创建入站过滤列表，来控制设备从 MSDP 对等体那里接收到的源信息。

通过创建入站过滤列表，用户可以控制设备从对等体那里接收到的入站 SA 消息，具体方法如下：

- 用户可以让设备忽略某台 MSDP 对等体发送过来的 SA 消息，以此来过滤来自某个特定 MSDP 对等体的入站 SA 消息；
- 用户可以让设备只接收某台 MSDP 对等体发来的那些与扩展访问列表定义的(S,G)对相匹配的 SA 消息，以此来根据扩展访问列表中定义的(S,G)对，过滤一部分某个特定 MSDP 对等体发送过来的入站 SA 消息。让设备忽略这个 MSDP 对等体发来的，不满足访问列表条件的 SA 消息；
- 用户可以让设备只接收那些与 route map 定义的标准相匹配的 SA 消息，以此来根据 route map 中定义的标准，过滤一部分由某个特定 MSDP 对等体发来的入站 SA 消息。让

设备忽略这个 MSDP 对等体发来的，不满足 route map 条件的 SA 消息；

- 用户可以让设备只有在 SA 消息同时匹配扩展访问列表中定义的(S,G)对，和 route map 中定义的标准时，才接收该 SA 消息。通过这种方式，用户可以同时根据扩展列表中定义的(S,G)对，和 route map 中定义的标准来过滤入站的 SA 消息。MSDP 对等体发来的其他入站数据包都会被忽略。
- 用户可以让设备只根据 SA 的来源来过滤 SA 消息（即使 SA 消息已经经由一台或多台 MSDP 对等体进行了转发），即根据 SA 消息中通告的 RP 地址来过滤来自某个对等体的入站 SA 消息；
- 用户可以配置一个入站过滤列表，其中包含一个扩展访问列表、一个 route map，以及一个 RP 访问列表或一个 RP route map。此时，MSDP 对等体必须保证所有用户定义的条件都满足，才会接收这个入站的 SA 消息。

注意： 随意过滤 SA 消息可能会让下游 MSDP 对等体无法接收到关于合法活动源的 SA 消息。因此，用户在配置过滤策略时应该谨慎。一般来说，入站过滤列表只应该用来拒绝那些不应该转发的源，譬如那些使用私有地址的源。

MSDP 中的 TTL 门限值

生存时间（TTL）让用户可以限制数据包在被丢弃之前，可以转发多少跳。用户可以使用命令 **ip multicast ttl-threshold** 来给封装的 SA 消息在发送给指定 MSDP 对等体的过程中，设置 TTL 值。在默认情况下，如果数据包的 TTL 值大于 0，那么 SA 消息中的组播数据包就可以转发给 MSDP 对等体，这就是标准的 TTL 操作方式。

总地来说，将初始组播数据包封装在一个 SA 消息中的这种方式，可能会带来 TTL 门限值的问题。由于组播数据包是封装在单播 SA 消息中的（单播 SA 消息的 TTL 值为 255），因此 SA 消息的 TTL 值在向 MSDP 对等体转发的过程中并不会减少。此外，SA 消息穿越的设备总跳数可能和常规的组播数据包明显不同，因为组播和单播流量可能会按照完全不同的路径到达 MSDP 对等体和远程 PIM-SM 域。因此，封装的数据包可能会违反 TTL 门限值的限制。解决问题的方法是使用命令 **ip multicast ttl-threshold** 配置一个 TTL 门限值，这是发送给某个 MSDP 对等体的 SA 消息中封装的组播数据包的门限值。在输入命令 **ip msdp ttl-threshol** 之后，如果组播数据包 IP 头部的 TTL 值小于参数 *ttl-value* 所设置的 TTL 值，那么这个组播数据包就不会封装到 SA 消息中，并且被发送给对等体。

SA 请求消息

用户可以配置一个非缓存设备，让它向一台或多台特定的 MSDP 对等体发送 SA 请求消息。如果一台非缓存 RP 有一台会缓存 SA 的 MSDP 对等体，用户可以让非缓存对等体发送 SA 请求消息，以此减少非缓存对等体的加入延迟。当一台主机请求加入一个特定组时，非缓存 RP 就会向其有缓存的对等体发送一条 SA 请求消息。如果对等体缓存了这个组的源信息，它就会将一条 SA 响应消息发送给请求的 RP。请求的 RP 会使用 SA 响应消息中的信息，但不会将消息转发给任何其他对等体。如果非缓存 RP 接收到了一条 SA 请求，它会向请求方发回一条错误消息。

注释： 在所有当前和支持的软件版本中，对 MSDP SA 消息执行缓存是强制性的，无法手动启用或禁用。在默认情况下，当用户配置 MSDP 对等体的时候，配置的命令就会自动添加到设备的运行配置当中。

SA 请求过滤器

在默认情况下，设备会处理所有 MSDP 对等体发来的 SA 请求消息；也就是说，设备会将缓存的源信息通过 SA 响应消息发送给请求的 MSDP 对等体。用户可以创建一个 SA 请求过滤器，来控制从特定对等体发来的出站 SA 请求消息。SA 请求过滤器会按照如下的方式，控制 MSDP 对等体发来的出站 SA 请求：

- 用户可以让设备忽略某台 MSDP 对等体发送过来的所有 SA 请求，以此来过滤来自某个特定对等体发来的所有 SA 请求消息；
- 用户可以让设备只响应某台 MSDP 对等体发来的那些与标准访问列表定义的组相匹配的 SA 请求消息，以此来根据标准访问列表中定义的组，过滤一部分某个特定对等体发送过来的 SA 请求消息。这个对等体发送的其他组的 SA 请求消息会被忽略。

如何使用 MSDP 来互连多个 PIM-SM 域

第一项配置任务是必配的，其它任务都是可选的。

配置 MSDP 对等体

注释： 通过启用 MSDP 对等体，可以隐式地启用 MSDP。

在开始前

- IP 组播路由必须启用，PIM-SM 必须配置；
- 除了一个 MSDP 对等体（即默认 MSDP 对等体），以及 MSDP 全互联组的环境中之外，用户必须在配置 MSDP 之前先让所有 MSDP 对等体运行 BGP。

总步骤

1 1. enable

2. configure terminal

3. ip msdp peer {*peer-name*| *peer-address*} [*connect-source type number*] [**remote-as** *as-number*]

4. ip msdp description {*peer-name*| *peer-address*} *text*

5. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip msdp peer** {*peer-name*| *peer-address*} [*connect-source type number*]	启用 MSDP，并且根据 DNS 名称或 IP 地址来配置 MSDP 对等体。 **注释：**选择要配置为 MSDP 对等体的设备通常也是

	[**remote-as** *as-number*] 示例： `Device(config)# ip msdp peer 192.168.1.2 connect-source loopback0`	BGP 邻居. 如果设置了关键字 **connect-source**，那么用户通过 *type* 和 *number* 值选择的本地接口的主用地址就会充当 TCP 连接的源 IP 地址。我们推荐用户配置关键字 **connect-source**，特别是在与远端域中的设备建立对等体的边界 MSDP 设备上
步骤 4	**ip msdp description** {*peer-name*| *peer-address*} *text* 示例： `Device(config)# ip msdp description 192.168.1.2 router at customer a`	（可选）给特定对等体配置一段描述文字，让用户更容易在 **show** 命令的输出信息中将其分辨出来
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

关闭 MSDP 对等体

用户可以根据需要来关闭 MSDP 对等体。

如果用户配置了多台 MSDP 对等体，而又不希望任何对等体在自己完全全部对等体的配置之前先行生效，那么用户就可以先关闭各个对等体、依次配置各个对等体，然后再打开这些对等体。用户有可能也希望关闭一条 MSDP 会话，同时又不丢失对这个 MSDP 对等体所作的配置命令。

注释： 在关闭 MSDP 对等体时，TCP 连接也会终结，直到用户（针对这个对等体）使用命令 **no ip msdp shutdown** 来重新打开这个对等体，TCP 连接才会重新建立起来。

在开始前

MSDP 已经开始运行，同时用户已经配置了 MSDP 对等体。

总步骤

1. enable

2. configure terminal

3. ip msdp shutdown {*peer-name* | *peer-address*}

4. 重复步骤 3 来根据需要关闭其他 MSDP 对等体

5. end

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device#` **configure terminal**	
步骤 3	**ip msdp shutdown {*peer-name* | *peer-address*}** 示例： `Device(config)# ip msdp shutdown 192.168.1.3`	管理关闭特定的 MSDP 对等体
步骤 4	重复步骤3来关闭其他MSDP对等体	——
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

在 MSDP 对等体之间配置 MSDP MD5 密码认证

用户可以根据需要在 MSDP 对等体之间配置 MSDP MD5 密码认证。

总步骤

1. enable

2. configure terminal

3. ip msdp password peer {*peer-name* | *peer-address*} [*encryption-type*] *string*

4. exit

5. show ip msdp peer [peer-address | peer-name]

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip msdp password peer** {*peer-name* | *peer-address*} [*encryption-type*] *string* 示例： `Device(config)# ip msdp password peer`	在两个 MSDP 对等体之间针对 TCP 连接启用 MD5 密码加密。 **注释**：在配置 MD5 认证时，两边的 MSDP 对等体上必须使用相同的密码。否则，它们之间的连接就无法建立起来。 • 如果配置或修改两个 MSDP 对等体之间使用的 MD5 认证的密码，本地设备不会在配置密码之后断开当前的连接。用户必须手动断开会

	10.32.43.144 0 test	话来激活新配置的密码
步骤 4	**exit** 示例： Device(config)# exit	离开全局配置模式并回到特权 EXEC 模式
步骤 5	**show ip msdp peer** [peer-address | peer-name] 示例： Device# show ip msdp peer	（可选）显示关于 MSDP 对等体的详细信息。 **注释：**使用这条命令来验证 MSDP 对等体上是否启用了 MD5 密码认证

排错技巧

如果一台 MSDP 对等体上配置了密码，另一台 MSDP 对等体上没有配置密码，那么当设备尝试相互建立 MSDP 会话时，控制台就会显示下面的信息：

```
%TCP-6-BADAUTH: No MD5 digest from [peer's IP address]:11003 to [local router's
IP address]:179
```

同样，如果两台设备上配置了不同的密码，那么控制台就会显示下面的信息：

```
%TCP-6-BADAUTH: Invalid MD5 digest from [peer's IP address]:11004 to [local router's
IP address]:179
```

命令 **debug ip tcp transactions** 的作用是显示重要的 TCP 交互信息，如状态交换、重传、重复包等等。在对 MSDP MD5 密码认证进行监控和排错时，使用命令 **debug ip tcp transactions** 可以查看到 MD5 密码是否已经启用，以及 MSDP 对等体是否接收到了保活消息。

通过针对特定 MSDP 对等体限制 SA 缓存中允许的 SA 消息数量来防止 DoS 攻击

用户可以根据需要(但强烈推荐)来限制设备从特定 MSDP 对等体那里接受的 SA 消息数量。这样做可以保护 MSDP 设备免遭分布式拒绝服务（DoS）攻击。

注释： 我们推荐用户在所有 MSDP 对等体设备上执行下面的配置。

总步骤

1. enable

2. configure terminal

3. ip msdp sa-limit {*peer-address* | *peer-name*} *sa-limit*

4. 重复步骤3来根据需要对其他MSDP对等体配置SA限制

5. exit

6. show ip msdp count [*as-number*]

7. show ip msdp peer [*peer-address* | *peer-name*]

8. show ip msdp summary

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密码

	示例： `Device> enable`	
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip msdp sa-limit** {*peer-address* | *peer-name*} *sa-limit* 示例： `Device(config)# ip msdp sa-limit 192.168.10.1 100`	限制 SA 缓存中可以针对某个特定 MSDP 对等体保存的 SA 消息数量
步骤 4	重复步骤3来关闭其他MSDP对等体	——
步骤 5	**exit** 示例： `Device(config)# exit`	离开全局配置模式并回到特权 EXEC 模式
步骤 6	**show ip msdp count** [*as-number*] 示例： `Device# show ip msdp count`	（可选）显示 MSDP SA 消息中发起的源和组的数量，以及 SA 缓存中来自一台 MSDP 对等体的 SA 消息数量
步骤 7	**show ip msdp peer** [peer-address | peer-name] 示例： `Device# show ip msdp peer`	（可选）显示关于 MSDP 对等体的详细信息。 **注释：**使用这条命令可以查看从某个 MSDP 对等体那里接收到的、保存在缓存中的 SA 消息数量
步骤 8	**show ip msdp summary** 示例： `Device# show ip msdp summary`	（可选）显示 MSDP 对等体的状态。 **注释：**使用这条命令可以查看每个对等体的“SA Count”值，这个参数会显示出缓存中保存的 SA 数量。

调整 MSDP 保活和抑制时间间隔

用户可以根据需要来调整 MSDP 对等体发送保活消息的时间间隔，，以及 MSDP 对等体会在宣告对端宕机之前，会等待多长时间来接收对端对等体发来的保活消息。在默认情况下，MSDP 会等待 75 秒啊词汇检测到对等体会话另一方的 MSDP 设备已经宕机。在部署了 MSDP 对等体的网络环境中，减少抑制时间间隔可以加速 MSDP 对等体出现故障时，MSDP 对等体的收敛时间。

注释： 我们不推荐用户修改命令 **ip msdp keepalive** 的默认设置，因为这条命令的默认设置是参照 RFC 3618，组播源发现协议的标准设计的。如果用户的网络需要修改这个默认值，那么在 MSDP 对等体两边的设备上，一定要配置相同的 *keepalive-interval* 和 *hold-time-interval* 参数。

总步骤

1. enable

2. configure terminal

3. ip msdp keepalive {*peer-address* | *peer-name*} *keepalive-interval hold-time-interval*

4. 重复步骤 3 来根据需要调整其他 MSDP 对等体的保活消息时间间隔

5. exit

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip msdp keepalive** {*peer-address* | *peer-name*} *keepalive-interval hold-time-interval* 示例： `Device(config)# ip msdp keepalive 10.1.1.3 40 55`	配置 MSDP 对等体发送保活消息的时间间隔，，以及 MSDP 对等体会在宣告对端宕机之前，会等待多长时间来接收对端对等体发来的保活消息。
步骤 4	重复步骤3来根据需要调整其他MSDP对等体的保活消息时间间隔	——
步骤 5	**exit** 示例： `Device(config)# exit`	离开全局配置模式并回到特权 EXEC 模式

调整 MSDP 连接重试时间间隔

用户可以根据需要来调整 MSDP 对等体在对等体会话重置之后，尝试重新建立对等体会话之前，会等待多长时间。在网络环境中，如果需要 SA 消息快速恢复（比如在交易大厅的网络环境中），那么用户可能就会希望将连接重试间隔时间设置为一个比默认的 30 秒更短的时间。

总步骤

1. enable

2. configure terminal

3. ip msdp timer *connection-retry-interval*

4. exit

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip msdp timer** *connection-retry-interval* 示例： `Device# ip msdp timer 45`	配置 MSDP 对等体在对等体会话重置之后，尝试重新建立对等体会话之前，会等待多长时间。
步骤 4	**exit** 示例： `Device(config)# exit`	离开全局配置模式并回到特权 EXEC 模式

配置默认 MSDP 对等体

用户可以根据需要来配置默认 MSDP 对等体。

在开始前

MSDP 默认对等体必须是之前配置的 MSDP 对等体。在配置默认 MSDP 对等体之前，必须首先配置 MSDP 对等体。

总步骤

1. enable

2. configure terminal

3. ip msdp default-peer {*peer-address* | *peer-name*} [**prefix-list** *list*]

4. exit

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device# configure terminal`	
步骤 3	**ip msdp default-peer** {*peer-address* | *peer-name*} [**prefix-list** *list*] 示例： `Device(config)# ip msdp default-peer 192.168.1.3`	配置设备从其接受所有 MSDP SA 消息的默认对等体
步骤 4	**exit** 示例： `Device(config)# exit`	离开全局配置模式并回到特权 EXEC 模式

配置 MSDP 全互联组

用户可以根据需要来配置 MSDP 全互联组。

注释： 用户可以针对各台设备分别配置多个全互联组。

总步骤

1. enable

2. configure terminal

3. ip msdp mesh-group *mesh-name* {*peer-address* | *peer-name*}

4. 重复步骤 3 来根据需要将 MSDP 对等体添加为全互联组的成员

5. exit

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip msdp mesh-group** *mesh-name* {*peer-address* | *peer-name*} 示例： `Device(config)# ip msdp mesh-group peermesh`	配置一个 MSDP 全互联组，并且将一个 MSDP 对等体配置为该互联组的成员。 **注释：**对于一台参与全互联组的设备，其所有 MSDP 对等体都必须与组中的所有其他 MSDP 对等体进行全互联。每台设备上都需要使用命令 **ip msdp peer** 将各个 MSDP 对等体配置为对等体，同时还要使用命令 **ip msdp mesh-group** 将 MSDP 对等体配置为全互联组的成员

步骤 4	重复步骤3来根据需要将MSDP对等体添加为全互联组的成员	——
步骤 5	**exit** 示例： `Device(config)# exit`	离开全局配置模式并回到特权 EXEC 模式

控制 RP 发起的 SA 消息中的本地源

用户可以根据需要启用一个过滤器，限制在 SA 消息中通告的注册源，以此达到控制 SA 消息中本地源的目的。

注释： 要想了解更多与配置 MSDP SA 消息过滤机制的最佳做法有关的信息，可以阅读组播源发现协议 SA 过滤推荐方案。

总步骤

1. enable

2. configure terminal

3. ip msdp redistribute [**list** *access-list*] [**asn** *as-access-list*] [**route-map** *map-name*]

4. exit

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip msdp redistribute** [**list** *access-list*] [**asn** *as-access-list*] [**route-map** *map-name*] 示例： `Device(config)# ip msdp redistribute route-map customer-sources`	针对本地设备发起的 MSDP SA 消息启用过滤机制。 **注释：**用户也可以配置命令 **ip msdp redistribute** 来通告 RP 所知道，但没有注册的源。但我们强烈推荐用户不要针对没有在 RP 上注册的源发起通告
步骤 4	**exit** 示例： `Device(config)# exit`	离开全局配置模式并回到特权 EXEC 模式

使用出站过滤列表来控制转发给 MSDP 对等体的 SA 消息

用户可以根据需要来配置出站过滤列表，来控制向 MSDP 对等体转发的 SA 消息。

注释： 要想了解更多与配置 MSDP SA 消息过滤机制的最佳做法有关的信息，可以阅读组播源发现协议 SA 过滤推荐方案。

总步骤

1. enable

2. configure terminal

3. ip msdp sa-filter out {*peer-address* | *peer-name*} [**list** *access-list*] [**route-map** *map-name*] [**rp-list** *access-list* | **rp-route-map** *map-name*]

4. 重复步骤 3 来根据需要针对其他 MSDP 对等体配置出站过滤列表

5. exit

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip msdp sa-filter out** {*peer-address* | *peer-name*} [**list** *access-list*] [**route-map** *map-name*] [**rp-list** *access-list* | **rp-route-map** *map-name*] 示例： `Device(config)# ip msdp sa-filter out 192.168.1.5 peerone`	给出站 MSDP 消息配置过滤策略
步骤 4	重复步骤3来根据需要针对其他MSDP对等体配置出站过滤列表	——
步骤 5	**exit** 示例： `Device(config)# exit`	离开全局配置模式并回到特权 EXEC 模式

使用出站过滤列表来控制从 MSDP 对等体接收的 SA 消息

用户可以根据需要来控制从 MSDP 对等体接收入站 SA 消息的操作。

注释： 要想了解更多与配置 MSDP SA 消息过滤机制的最佳做法有关的信息，可以阅读组播源发现协议 SA 过滤推荐方案。

总步骤

1. enable

2. configure terminal

3. ip msdp sa-filter in {*peer-address* | *peer-name*} [**list** *access-list*] [**route-map** *map-name*] [**rp-list** *access-list* | **rp-route-map** *map-name*]

4. 重复步骤 3 来根据需要针对其他 MSDP 对等体配置入站过滤列表

5. exit

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip msdp sa-filter in** {*peer-address* | *peer-name*} [**list** *access-list*] [**route-map** *map-name*] [**rp-list** *access-list* | **rp-route-map** *map-name*] 示例： `Device(config)# ip msdp sa-filter in 192.168.1.3`	给入站 MSDP 消息配置过滤策略
步骤 4	重复步骤3来根据需要针对其他MSDP对等体配置入站过滤列表	——
步骤 5	**exit** 示例： `Device(config)# exit`	离开全局配置模式并回到特权 EXEC 模式

使用 TTL 门限值来限制 SA 消息中发送的组播数据

用户可以根据需要来建立一个生存时间（TTL）门限值来限制 SA 消息中发送的组播数据。

总步骤

1. enable

2. configure terminal

3. ip msdp ttl-threshold {*peer-address* | *peer-name*} *ttl-value*

4. exit

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip msdp ttl-threshold** {*peer-address* | *peer-name*} *ttl-value* 示例： `Device(config)# ip msdp ttl-threshold 192.168.1.58`	给本地设备发起的 MSDP 消息设置一个 TTL 值 • 在默认情况下，如果数据包的 TTL 值大于 0，那么 SA 消息中的组播数据包就可以转发给 MSDP 对等体，这就是标准的 TTL 操作方式
步骤 4	**exit** 示例： `Device(config)# exit`	离开全局配置模式并回到特权 EXEC 模式

从 MSDP 对等体那里请求源信息

用户可以根据需要让设备从 MSDP 对等体那里请求源信息。

注释： 在之前的 Inspur 软件版本中，对 MSDP SA 消息执行缓存是强制性的，无法手动启用或禁用，因此用户很少需要执行这项操作。

总步骤

1. enable

2. configure terminal

3. ip msdp sa-request {*peer-address* | *peer-name*}

4. 重复步骤 3 来让设备向其他 MSDP 缓存对等体发送 SA 请求消息

5. exit

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device#` **configure terminal**	
步骤 3	**ip msdp sa-request** {*peer-address* | *peer-name*} 示例： `Device(config)# ip msdp sa-request 192.168.10.1`	让设备向指定的 MSDP 对等体发送 SA 请求消息
步骤 4	重复步骤3来让设备向其他MSDP缓存对等体发送SA请求消息	——
步骤 5	**exit** 示例： `Device(config)# exit`	离开全局配置模式并回到特权 EXEC 模式

使用 SA 请求过滤器来控制对 MSDP 对等体的出站 SA 请求消息作出的响应

用户可以根据需要来控制设备会处理哪些从 MSDP 对等体那里发来的出站 SA 请求消息。

总步骤

1. enable

2. configure terminal

3. ip msdp filter-sa-request {*peer-address* | *peer-name*} [**list** *access-list*]

4. 重复步骤 3 来针对其他设备配置 SA 请求过滤器

5. exit

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip msdp filter-sa-request** {*peer-address* | *peer-name*} [**list** *access-list*] 示例：	针对出站 SA 请求消息启用过滤器 **注释**：针对每个 MSDP 对等体只能配置一个 SA 请求过滤器

	Device(config)# ip msdp filter sa-request 172.31.2.2 list 1	
步骤 4	重复步骤3来让设备向其他MSDP缓存对等体发送SA请求消息	——
步骤 5	**exit** 示例： Device(config)# exit	离开全局配置模式并回到特权 EXEC 模式

在 MSDP 中包含边界 PIM 密集模式

用户可以根据需要来控制边界设备，让它们对 PIM 密集模式区域的活动源发送 SA 消息。用户可以配置一台设备，让它同时充当 PIM-SM 域和 PIM-DM 域的边界。在默认情况下，PIM-DM 域中的源都不包含在 MSDP 中。用户可以对这台边界设备进行配置，让它针对 PIM-SM 域中的活动源发送 SA 消息。如果进行了上述配置，那么用户一定也要配置命令 **ip msdp redistribute** 来控制对 PIM-SM 域中的哪些本地源进行通告。如果不进行这样配置，那么在 PIM-DM 域中的一个源不再发送消息之后，(S,G)状态仍然会被长时间保留下来。要想了解更多配置信息，请参阅控制 RP 发起的 SA 消息中的本地源部分。

总步骤

1. enable

2. configure terminal

3. ip msdp border sa-address *type number*

4. exit

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： Device> **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： Device# **configure terminal**	进入全局配置模式
步骤 3	**ip msdp border sa-address** *type number* 示例： Device(config)# ip msdp border sa-address gigabitethernet0/0/0	配置 PIM-SM 和 PIM-DM 域边界的设备，让它针对 PIM-DM 域中的活动源发起 SA 消息。 • 接口的 IP 地址会充当发起方 ID，也即 SA 消息中的 RP 字段
步骤 4	**exit**	离开全局配置模式并回到特权 EXEC 模式

	示例： `Device(config)# exit`	

配置 RP 地址之外的发起地址

用户可以根据需要，让发起 SA 消息的 MSDP 设备使用其接口的 IP 地址作为 SA 消息中的 RP 地址。

用户可以因下列原因来修改发起方 ID：

- 如果用户在 MSDP 互联组中针对任意播 RP 配置了多台设备；
- 如果用户有一台设备是 PIM-SM 域和 PIM-DM 域的边界设备。如果一台设备是 PIM-SM 域和 PIM-DM 域的边界设备，而用户又希望在 PIM-SM 域中通告活动源，那就要将 SA 消息中的 RP 地址配置为发起设备接口的地址。

在开始前

要首先启用 MSDP，并配置好 MSDP 对等体。要想了解更多关于配置 MSDP 对等体信息，请参阅配置 MSDP 对等体部分。

总步骤

1. enable

2. configure terminal

3. ip msdp originator-id *type number*

4. exit

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip msdp originator-id** *type number* 示例： `Device(config)# ip msdp originator-id ethernet 1`	将 SA 消息中的 RP 地址配置为发起设备接口的地址
步骤 4	**exit** 示例： `Device(config)# exit`	离开全局配置模式并回到特权 EXEC 模式

监控 MSDP

用户可以根据需要对 MSDP SA 消息、对等体、状态和对等体状态进行监控。

总步骤

1. enable

2. debug ip msdp [*peer-address* | *peer-name*] [**detail**] [**routes**]

3. debug ip msdp resets

4. show ip msdp count [*as-number*]

5. show ip msdp peer [*peer-address* | *peer-name*]

6. show ip msdp sa-cache [*group-address* | *source-address* | *group-name* | *source-name*] [*as-number*]

7. show ip msdp summary

具体步骤

步骤 1　enable

示例：

```
Device> enable
```

进入特权 EXEC 模式。

- 在提示时输入密码。

步骤 2　debug ip msdp [*peer-address* | *peer-name*] [**detail**] [**routes**]

用户可以使用这条命令来调试 MSDP 的活动。

用户可以使用可选参数 *peer-address* 或 *peer-name* 来设置对哪些对等体调试事件进行记录。

下面是命令 **debug ip msdp** 的输出示例：

示例：

```
Device# debug ip msdp
MSDP debugging is on
Device#
MSDP: 224.150.44.254: Received 1388-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 1388, ec: 115, RP: 172.31.3.92
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.92, used EMBGP peer
MSDP: 224.150.44.250: Forward 1388-byte SA to peer
MSDP: 224.150.44.254: Received 1028-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 1028, ec: 85, RP: 172.31.3.92
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.92, used EMBGP peer
MSDP: 224.150.44.250: Forward 1028-byte SA to peer
MSDP: 224.150.44.254: Received 1388-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 1388, ec: 115, RP: 172.31.3.111
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.111, used EMBGP peer
MSDP: 224.150.44.250: Forward 1388-byte SA to peer
MSDP: 224.150.44.250: Received 56-byte message from peer
MSDP: 224.150.44.250: SA TLV, len: 56, ec: 4, RP: 192.168.76.241
MSDP: 224.150.44.250: Peer RPF check passed for 192.168.76.241, used EMBGP peer
MSDP: 224.150.44.254: Forward 56-byte SA to peer
MSDP: 224.150.44.254: Received 116-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 116, ec: 9, RP: 172.31.3.111
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.111, used EMBGP peer
```

```
MSDP: 224.150.44.250: Forward 116-byte SA to peer
MSDP: 224.150.44.254: Received 32-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 32, ec: 2, RP: 172.31.3.78
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.78, used EMBGP peer
MSDP: 224.150.44.250: Forward 32-byte SA to peer
```

步骤 3　debug ip msdp resets

用户可以使用这条命令来调试 MSDP 对等体的重置原因。

示例：

```
Device# debug ip msdp resets
```

步骤 4　show ip msdp count [*as-number*]

用户可以使用这条命令来查看 MSDP SA 消息中发起的源和组的数量，以及 SA 缓存中，MSDP 对等体发来的 SA 消息数量。用户要想让这条命令产生输出信息，必须配置命令 **ip msdp cache-sa-state**。

下面是命令 **show ip msdp count** 的输出信息示例：

示例：

```
Device# show ip msdp count
SA State per Peer Counters, <Peer>: <# SA learned>
192.168.4.4: 8
SA State per ASN Counters, <asn>: <# sources>/<# groups>
Total entries: 8
?: 8/8
```

步骤 5　show ip msdp peer [*peer-address* | *peer-name*]

用户可以使用这条命令来显示关于 MSDP 对等体的具体信息。

用户可以使用可选参数 *peer-address* 或 *peer-name* 来查看关于某个特定对等体的信息。

下面是命令 **show ip msdp peer** 的输出信息示例：

示例：

```
Device# show ip msdp peer 192.168.4.4
MSDP Peer 192.168.4.4 (?), AS 64512 (configured AS)
Connection status:
State: Up, Resets: 0, Connection source: Loopback0 (2.2.2.2)
Uptime(Downtime): 00:07:55, Messages sent/received: 8/18
Output messages discarded: 0
Connection and counters cleared 00:08:55 ago
SA Filtering:
Input (S,G) filter: none, route-map: none
Input RP filter: none, route-map: none
Output (S,G) filter: none, route-map: none
Output RP filter: none, route-map: none
SA-Requests:
Input filter: none
Peer ttl threshold: 0
SAs learned from this peer: 8
Input queue size: 0, Output queue size: 0
MD5 signature protection on MSDP TCP connection: not enabled
```

步骤 6 **show ip msdp sa-cache** [*group-address* | *source-address* | *group-name* | *source-name*] [*as-number*]

用户可以使用这条命令来查看从 MSDP 对等体那里学习到的(S,G)状态。

下面是命令 **show ip msdp sa-cache** 的输出信息示例：

示例：

```
Device# show ip msdp sa-cache
MSDP Source-Active Cache - 8 entries
(10.44.44.5, 239.232.1.0), RP 192.168.4.4, BGP/AS 64512, 00:01:20/00:05:32, Peer
192.168.4.4
(10.44.44.5, 239.232.1.1), RP 192.168.4.4, BGP/AS 64512, 00:01:20/00:05:32, Peer
192.168.4.4
(10.44.44.5, 239.232.1.2), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer
192.168.4.4
(10.44.44.5, 239.232.1.3), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer
192.168.4.4
(10.44.44.5, 239.232.1.4), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer
192.168.4.4
(10.44.44.5, 239.232.1.5), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer
192.168.4.4
(10.44.44.5, 239.232.1.6), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer
192.168.4.4
(10.44.44.5, 239.232.1.7), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer
192.168.4.4
```

步骤 7 **show ip msdp summary**

用户可以使用这条命令来查看 MSDP 对等体的状态。

下面是命令 **show ip msdp summary** 的输出信息示例：

示例：

```
Device# show ip msdp summary
MSDP Peer Status Summary
Peer Address AS State Uptime/ Reset SA Peer Name
Downtime Count Count
192.168.4.4 4 Up 00:08:05 0 8 ?
```

清除 MSDP 连接统计数据及 SA 缓存条目

用户可以根据需要清除 MSDP 的连接、统计数据和 SA 缓存条目。

总步骤

1. enable

2. clear ip msdp peer [*peer-address* | *peer-name*]

3. clear ip msdp statistics [*peer-address* | peer-name]

4. clear ip msdp sa-cache [*group-address*]

具体步骤

	命令或操作	目的

步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**clear ip msdp peer** [*peer-address* | *peer-name*] 示例： `Device# configure terminal`	清除与指定 MSDP 对等体之间的 TCP 连接，并重置所有 MSDP 消息的计数器
步骤 3	**clear ip msdp statistics** [*peer-address* | peer-name] 示例： `Device# clear ip msdp statistics`	清除与指定 MSDP 对等体之间的统计数据计数器，并重置所有 MSDP 消息的计数器
步骤 4	**clear ip msdp sa-cache** [*group-address*] 示例： `Device# clear ip msdp sa-cache`	清除 SA 缓存条目。 • 如果用户在输入命令时，设置了可选的参数 *group-address* 或参数 *source-address*，那么所有 SA 缓存条目就都会被清除； • 使用可选的参数 *group-address* 来清楚与一个特定组相关的所有 SA 缓存条目

针对 MSDP 启用 SNMP 监控

用户可以根据需要启用 MSDP 的简单网络管理协议（SNMP）监控。

在开始前

- 要在设备上配置好 SNMP 和 MSDP；
- 在每个 PIM-SM 域中，都应该有一台设备被配置为了 MSDP 设备。这台设备上必须同时启用了 SNMP 和 MSDP MIB。

注释： 所有 MSDP-MIB 对象都被配置为只读；

在 Inspur 网络中实施 MSDP MIB 时，不支持请求表；

在 Inspur 网络中实施 MSDP MIB 时，不支持 msdpEstablished 通告消息

总步骤

1. enable

2. snmp-server enable traps msdp

3. snmp-server host *host* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth**| **priv** | **noauth**]}] *community-string* [**udp-port** *port-number*] **msdp**

4. exit

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例：	进入特权 EXEC 模式。在提示时输入密码

	Device> **enable**	
步骤 2	**snmp-server enable traps msdp** 示例： Device# snmp-server enable traps msdp	在设备上启用 MSDP 通告发送，以支持 SNMP。 **注释：**命令 **snmp-server enable traps msdp** 会同时启用 trap 和通告
步骤 3	**snmp-server host** *host* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth**| **priv** | **noauth**]}] *community-string* [**udp-port** *port-number*] **msdp** 示例： Device# snmp-server host examplehost msdp	设置 MSDP trap 和通告的接收方（主机）
步骤 4	**exit** 示例： Device(config)# exit	离开全局配置模式并回到特权 EXEC 模式

排错技巧

用户可以将 MSDP MIB 通告的结果，与在设备上输入命令 **show ip msdp summary** 和命令 **show ip msdp peer** 后的输出信息进行比较。用户也可以将这些命令的输出信息，与 SNMP Get 操作的结果进行比较。用户可以使用命令 **show ip msdp sa-cache** 来查看 SA 缓存表的条目。其他排错信息，如连接的本地地址、本地端口、远程端口，可以从命令 **debug ip msdp** 的输出信息中获得。

使用 MSDP 来互连多个 PIM-SM 域的配置示例

示例：配置 MSDP 对等体

下面的示例显示了如何在 MSDP 对等体之间建立 MSDP 对等体连接：

设备 A

```
!
interface Loopback 0
ip address 10.220.8.1 255.255.255.255
!
ip msdp peer 10.220.16.1 connect-source Loopback0
ip msdp peer 10.220.32.1 connect-source Loopback0
!
```

设备 B

```
!
interface Loopback 0
```

```
ip address 10.220.16.1 255.255.255.255
!
ip msdp peer 10.220.8.1 connect connect-source Loopback0
ip msdp peer 10.220.32.1 connect connect-source Loopback0
!
```

设备 C

```
!
interface Loopback 0
ip address 10.220.32.1 255.255.255.255
!
ip msdp peer 10.220.8.1 connect 10.220.8.1 connect-source Loopback0
ip msdp peer 10.220.16.1 connect 10.220.16.1 connect-source Loopback0
!
```

示例：配置 MSDP MD5 密码认证

下面的示例显示了如何在两台 MSDP 对等体之间针对 TCP 连接启用 MD5 密码认证：

设备 A

```
!
ip msdp peer 10.3.32.154
ip msdp password peer 10.3.32.154 0 test
!
```

设备 B

```
!
ip msdp peer 10.3.32.153
ip msdp password peer 10.3.32.153 0 test
!
```

示例：配置默认 MSDP 对等体

这张图显示了一个使用了默认 MSDP 对等体的环境。在图中，拥有设备 B 的客户通过两个互联网服务提供商（ISP）连接到了 Internet，其中一个 ISP 拥有设备 A，另一个 ISP 拥有设备 C。它们之间没有运行 MBGP。为了让客户学习到 ISP 域内或其他域中的源，设备 B 将设备 A 标识为了自己的默认 MSDP 对等体。设备 B 同时向设备 A 和设备 C 通告了 SA 消息，但只接受设备 A 或只接受设备 C 发来的 SA 消息。如果设备 A 是配置中的第一台默认对等体，那么只要它运行正常，设备就会使用设备 A。只有在设备 A 无法正常工作时，设备 B 才会接受从设备 C 发来的 SA 消息。

ISP 也可能会使用前缀列表来定义它会接受客户设备发来的哪些前缀。客户端会定义多台默认对等体，每台对等体关联有一个或多个前缀。

客户端有两个 ISP 可以使用。客户端同时将这两个 ISP 都定义为了默认对等体。只要配置中设置的第一个默认对等体运行正常，它就会称为默认对等体，设备也只会接受它发来过来的所有 SA 消息。

注释： 虽然下图和下面的示例在配置中使用得到都是路由器，但用户可以使用任意设备

（路由器和交换机）。

图 31：默认 MSDP 对等体的环境

Device A	设备 A
Default MSDP peer	默认 MSDP 对等体
ISP A PIM domain	ISP A 的 PIM 域
Device C	设备 C
Default MSDP peer	默认 MSDP 对等体
ISP C PIM domain	ISP C 的 PIM 域
Device B	设备 B
Default MSDP peer	默认 MSDP 对等体
Customer PIM domain	客户的 PIM 域

设备 B 同时向设备 A 和设备 C 通告了 SA 消息，但只接受设备 A 或只接受设备 C 发来的 SA 消息。如果设备 A 是配置中的第一台默认对等体，那么只要它运行正常，设备就会使用设备 A。只有在设备 A 无法正常工作时，设备 B 才会接受从设备 C 发来的 SA 消息。

如果用户设置了一个前缀列表，那么对等体只会成为列表中前缀的默认对等体。如果用户给每台默认对等体关联一个前缀列表的话，那么一台设备可以有多个活动的默认对等体。如果没有配置前缀列表，同时又配置了多台默认对等体，那么只要其中的第一台对等体与设备之间保持连通，并且这台设备没有宕机，那么它就是唯一活动的默认对等体。如果用户配置的第一台设备宕机，或者对等体连接断开，那么用户配置的第二台设备就会成为活动的默认对等体，以此类推。

下面的示例显示了图中设备 A 和设备 C 上的部分配置。这两个 ISP 都可能有多个使用默认对等体的客户，就像图中的客户那样。此时，它们的配置都是相似的。也就是说，它们只会接受默认对等体发来的 SA，只要这些 SA 获得了对应前缀列表的放行。

设备 A 的配置

```
ip msdp default-peer 10.1.1.1
ip msdp default-peer 10.1.1.1 prefix-list site-b ge 32
ip prefix-list site-b permit 10.0.0.0/8
```

设备 C 的配置

```
ip msdp default-peer 10.1.1.1 prefix-list site-b ge 32
ip prefix-list site-b permit 10.0.0.0/8
```

示例：配置 MSDP 全互联组

下面的案例显示了如何将 3 台设备配置为 MSDP 全互联组的全互联成员。

设备 A 的配置

```
ip msdp peer 10.2.2.2
ip msdp peer 10.3.3.3
ip msdp mesh-group test-mesh-group 10.2.2.2
ip msdp mesh-group test-mesh-group 10.3.3.3
```

设备 B 的配置

```
ip msdp peer 10.1.1.1
ip msdp peer 10.3.3.3
ip msdp mesh-group test-mesh-group 10.1.1.1
```

```
ip msdp mesh-group test-mesh-group 10.3.3.3
```

设备 C 的配置

```
ip msdp peer 10.1.1.1
ip msdp peer 10.2.2.2
ip msdp mesh-group test-mesh-group 10.1.1.1
ip msdp mesh-group test-mesh-group 10.2.2.2
```

其他参考资料

相关文档

相关主题	文档名
IPv6 编址与连接	《IPv6 配置指南》
如需了解本章所述命令的完整语法结构及使用信息	《IP 组播路由命令参考手册（Inspur 6650 交换机）》
Inspur INOS 命令	《Inspur INOS 主命令列表，所有版本》
IP 组播命令	《Inspur INOS IP 组播命令参考手册》
IPv6 命令	《Inspur INOS IPv6 命令参考手册》
Inspur INOS IPv6 特性	《Inspur INOS IPv6 特性映射》

标准与 RFC

标准/RFC	标题
IPv6 的 RFC 标准	IPv6 RFC

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

组播源发现协议的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

配置 SSM

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置 SSM 的前提条件

下面是配置特定源组播（SSM）和 SSM 映射的前提条件：

- 在配置 SSM 映射之前，用户必须执行下面的配置任务：
  - 启用 IP 组播路由；
  - 启用 PIM 稀疏模式；
  - 配置 SSM；
- 在配置静态 SSM 映射之前，必须配置好访问控制列表（ACL），来定义要被映射为源地址的组范围；
- 用户需要先向运行的 DNS 服务器中添加记录，然后才能通过 DNS 查找来配置和使用 SSM 映射。如果没有 DNS 服务器，那就需要先安装一台 DNS 服务器。

注释： 用户可以使用诸如 Inspur 网络注册器（Inspur Network Registrar）这样的产品来向运行的 DNS 服务器中添加记录。

配置 SSM 的限制条件

下面是配置 SSM 的限制条件：

- 要在 IGMPv3 环境中运行 SSM，那么 Inspur INOS 路由器、应用所运行的主机和应用本身必须都能够支持 SSM；
- 在 SSM 之前，网络中的应用无法工作在 SSM 范围之内，除非用户对它们进行了修改，让它们支持订阅(S,G)信道。因此，如果当前的应用使用了指定的 SSM 范围，那么在网络中启用 SSM 可能会对这些应用造成影响；
- IGMP Snooping——IGMPv3 使用了新的成员关系报告消息，这些报告老版的 IGMP snooping 设备有可能无法正常识别；
- 在将 SSM 与二层交换机制一起使用时，人们在某种程度上还是需要进行地址管理。Inspur 组管理协议（CGMP），IGMP snooping 或路由器端口组管理协议（RGMP）只支持

特定组过滤，不支持(S,G)特定信道过滤。如果一个交换型网络的不同接收方请求共享同一个组的不同(S,G)信道，那么这些接收方就无法从当前的机制当中获益。相反，这些接收方都会接收到所有(S,G)信道的流量，并且在入站时过滤掉不需要的流量。由于 SSM 可以对许多独立应用复用 SSM 范围的组地址，因此交换型网络中过滤的流量就会减少。有鉴于此，用户要对应用使用 SSM 范围中的随机 IP 地址，这样可以减少在不同应用中复用 SSM 范围内某一个地址的几率，这一点十分重要。例如，一个提供一系列电视频道的应用应该给每个电视(S,G)频道使用不同的组，即使使用 SSM 也是如此。这种设置方法可以保证对于同一个应用服务来说，其不同信道中的多个接收方不会在包含二层设备的网络中遭遇到流量混淆的问题；

- 在 PIM-SSM 环境中，只要接口有对应的(S,G)订阅，最后一跳路由器就会继续周期性地发送(S,G)加入消息。因此，只要接收方发送(S,G)订阅，从接收方到源的最短路径树（SPT）状态就会不断得到维护，即使这个源已经很长时间没有发送流量（甚至永远也不会发送流量）了也是如此；

  PIM-SM 的情况与此相反，只有源仍然在发送流量，而且接收方加入了这个组时，设备才会维护(S,G)状态。在 PIM-SM 中，如果源超过 3 分钟没有发送流量了，那么设备就会删除(S,G)状态，而且只有在源发送的数据包再次通过 RPT（汇集点树）到达之后，这个状态才会重新建立起来。由于在 PIM-SSM 环境中，没有机制可以通告接收方源处于活动状态，因此只要接收方还在请求接收这条信道的流量，网络就必须维护(S,G)状态。

下面是配置 SSM 映射的限制条件：

- SSM 映射特性并不拥有 SSM 的所有优势。SSM 映射会从主机那里获取到组 G 的加入消息，并且用一个（关联了一个或多个源的）应用来标识这个组。因此，它只能针对每个组 G 支持一个这样的应用。不过，完整的 SSM 应用可能会共享 SSM 应用中使用的组；
- 如果用户完全依靠 SSM 映射作为完整 SSM 的传输解决方案，那么在最后一跳路由器上启用 IGMPv3 时就一定要格外小心。在同时启用 SSM 映射和 IGMPv3，且主机已经支持 IGMPv3（但不支持 IGMP）时，主机就会发送 IGMPv3 组报告。SSM 映射不支持这些 IGMPv3 组报告，而路由器不能正确地将这些报告与源进行关联。

关于 SSM 的信息

特定源组播（SSM）特性是 IP 组播的一种扩展，即网络只将有接收方明确加入的组播源发送的数据流量转发给接收方。对于配置了 SSM 的组播组，网络只会创建 SSM 分发树（而不是共享树）。

在这一节介绍了如何配置特定源组播（SSM）。用户要想进一步了解本节中描述的 SSM 命令，可以参阅 IP 组播命令参考手册。要查询本章中出现的其他命令，可以使用命令参考主索引，或者在线进行搜索。

SSM 组件概述

SSM 是一种数据报文传输模型，这种模型特别适合用于那些执行一对多传输的应用，也就是通称的广播应用。SSM 是 Inspur 在音频和视频广播应用环境中实施 IP 组播的核心网络技术。

设备支持下列支持 SSM 实施的组件：

- 协议独立组播特定源模式（PIM-SSM）。

  PIM-SSM 是一种支持实施 SSM 的路由协议，这种协议来源于 PIM 稀疏模式（PIM-SM）；

- 互联网组管理协议第 3 版（IGMPv3）

SSM 与互联网标准组播（ISM）

在互联网和很多企业的内联网中，当前的 IP 组播架构都是基于 PIM-SM 协议和组播源发现协议（MSDP）的。这些协议对于互联网标准组播（ISM）服务模型存在限制。例如，在 ISM 环境中，网络必须了解网络中有哪些主机正在活动地发送组播流量。

ISM 的服务包含从任意源，向一组接收方（称为组播主机组）传输 IP 数据报。去往组播主机组的数据流量，是由源地址为任意 IP 单播源地址（S），目的地址为组播组地址（G）的数据流所组成的。当系统成为主机组的成员时，它就会接收到这些流量。主机组中的成员只需要通过 IGMP 第 1 版、第 2 版或第 3 版来将流量通告给主机组。

在 SSM 中，数据包基于(S,G)信道进行传输的。在 SSM 和 ISM 环境中，成为源不需要发送信令。但在 SSM 中，接收方必须通过订阅和取消订阅(S,G)信道的方式，来接收或停止接收某个源的流量。换言之，接收方只能从自己订阅的(S,G)信道中接收到的流量。而在 ISM 中，接收方不需要了解源的 IP 地址。针对订阅信道的信令，标准方法是使用 IGMP，并且包含模式成员关系报告，这种保持只有 IGMP 第 3 版可以支持。

SSM 的 IP 地址范围

用户可以将 SSM 的传输模型应用于一个配置好的 IP 地址组地址范围的子集，实现 SSM 与 ISM 服务的共存。Inspur INOS 软件可以对范围在 224.0.0.0 到 239.255.255.255 之间的 IP 组播地址配置 SSM。在定义好了 SSM 范围之后，当网络中当前的 IP 组播接收方应用尝试使用 SSM 范围中的一个地址时，它们不会接收到任何流量（除非用户使用显式的(S,G)信道订阅修改了应用）。

SSM 的操作

如果一个网络基于 PIM-SM 来提供 IP 组播服务，那么这个网络就可以支持 SSM 服务。如果网络中只需要 SSM 服务，那么 SSM 也可以在网络中单独进行部署，而不需要使用部署域间 PIM-SM 所需的那一套完整的协议（例如，MSDP、Auto-RP、自举路由器[BSR]）。

如果在一个已经配置了 PIM-SM 的网络中部署 SSM，那么只有最后一跳路由器会支持 SSM。不与接收方直连的路由器不需要支持 SSM。总地来说，在 SSM 范围中，这些非最后一跳路由器必须只运行 PIM-SM，而且有可能需要执行一些访问控制配置来抑制 SSM 范围中发生的 MSDP 信令、注册，或者 PIM-SM 共享树操作。

用户可以使用全局配置命令 **ip pim ssm** 来配置 SSM 范围并启用 SSM。这条配置命令会产生下列效果：

- 对于 SSM 范围内的组，可以通过 IGMPv3 include（包含）模式的成员关系报告来接受对(S,G)信道的订阅；
- 在 SSM 范围内的地址中，PIM 的操作会更改为 PIM-SSM，这是一种来源于 PIM-SM 的模式。在这种模式下，路由器只会创建 PIM(S,G)加入和修剪消息，但不会创建(S,G)汇集点树（RPT）或者(*,G)RPT 消息。与 RPT 相关的入站消息会被忽略或者拒绝，而入站方向的 PIM 注册消息会立刻通过注册停止消息得到响应。PIM-SSM 可以向后兼容 PIM-SM，

除非这台路由器是最后一跳路由器。因此，不是最后一跳路由器的路由器都可以针对 SSM 组运行 PIM-SM（如果路由器还不支持 SSM 时，就可以这样操作）；

- 网络不会接受、生成或者转发 SSM 范围内的 MSDP 源活动（SA）消息。

SSM 映射

在一个典型的机顶盒部署环境中，每个 TV 频道都会使用一个独立的 IP 组播组，并拥有一个发送 TV 频道的活动服务器主机。一台服务器可以发送多个组播 TV 频道，但每个频道对应一个不同的组。在这种网络环境中，如果一台路由器接收到了某个组的 IGMPv1 或 IGMPv2 成员关系报告，路由器就会将报告发送给组播组对应 TV 频道的知名 TV 服务器。

如果用户配置了 SSM 映射，那么若路由器接收到了某个组的 IGMPv1 或 IGMPv2 成员关系报告，路由器就会将报告转换为发往该组对应的知名源的（一个或几个）信道成员关系。

若路由器接收到了某个组的 IGMPv1 或 IGMPv2 成员关系报告，路由器会使用 SSM 映射来判断出这个组的一个或几个源 IP 地址。接下来，SSM 映射就会将这个成员关系报告转换为一个 IGMPv3 报告，并且按照它接收到的就是 IGMPv3 报告那样进行后续的处理。接下来，路由器会发送 PIM 加入消息，并且只要这台路由器还能继续接收到 IGMPv1 或 IGMPv2 成员关系报告，它就会继续保持在这个组中，而这个组的 SSM 映射关系也会保持不变。

SSM 映射会让最后一跳路由器能够通过用户在路由器上静态配置的表，或者通过 DNS 服务器判断出源地址。当静态配置的表或者 DNS 映射出现变化时，路由器会离开加入的组当前关联的源。

静态 SSM 映射

使用配置静态 SSM 映射的方式，用户可以对最后一跳路由器进行配置，让它使用静态映射来判断正在向组发送数据的源。静态 SSM 映射要求用户配置 ACL 来定义组范围。在配置了 ACL 来定义组范围之后，用户可以使用全局配置命令 **ip igmp ssm-map static** 将与 ACL 匹配的组映射到源。

在不需要使用 DNS 的小型网络中，或者当用户希望用静态配置的 SSM 映射覆盖 DNS 映射时，就可以配置静态 SSM 映射。如果配置了静态 SSM 映射，那么它的优先级会高于 DNS 映射的优先级。

基于 DNS 的 SSM 映射

用户可以使用基于 DNS 的映射来配置最后一跳路由器，让它执行逆向 DNS 查询，以判断向组发送数据的源。如果用户配置了基于 DNS 的 SSM 映射，那么路由器就会创建一个包含组地址的域名，并对 DNS 执行逆向查找。路由器会查找 IP 地址资源记录，并且用这些记录作为这个组关联的源地址。对于每个组，SSM 映射支持最多 20 个源。路由器会加入所有给组配置的源。

下图显示了基于 DNS 的 SSM 映射。

图 33：基于 DNS 的 SSM 映射

Source	源
(S,G) Join	(S,G)加入消息
(S,G) Join	(S,G)加入消息
DNS response	DNS 响应消息
Reverse DNS lookup	逆向 DNS 查找
DNS server	DNS 服务器
IGMPv2 membership report	IGMPv2 成员关系报告

STP host 1	STP 主机 1
STP host 2	STP 主机 2
STP host 3	STP 主机 3

SSM 映射机制可以让最后一跳路由器加入组的多个源，这种机制可以给 TV 广播提供源的冗余性。在这个环境中，最后一跳路由器会使用 SSM 映射来同时加入同一个 TV 频道的两个视频源，以此实现冗余。不过，为了防止最后一跳路由器复制视频流量，视频源必须使用服务器端的切换机制。一旦视频源是主动的，另一个备用的视频源是被动的。被动的视频源会在检测到主动源出现故障时，才会向 TV 频道中发送视频流量。因此，服务器端的切换机制可以确保只有一台服务器会主动向 TV 频道发送视频流量。

要查找某个包含 G1、G2、G3 和 G4 的组中的一个或多个源地址，用户必须在 DNS 服务器上配置这些 DNS 记录。

```
G4.G3.G2.G1 [multicast-domain] [timeout] IN A source-address-1
IN A source-address-2
IN A source-address-n
```

用户可以查看 DNS 服务器文档，来了解更多关于配置 DNS 资源记录的信息。

如何配置 SSM

如需获取本节中关于特定源组播（SSM）命令的完整描述，可以参阅 IP 组播命令参考手册，Inspur INOS XE 3SE 版（Inspur 3850 交换机）。要查询本章中出现的其他命令，可以使用命令参考主索引，或者在线进行搜索。

配置 SSM（CLI）

用户可以按照下面的步骤来配置 SSM：

这个流程是可选的。

在开始前

如果用户希望使用访问列表来定义特定源组播（SSM）范围，那就要在通过命令 ip pim ssm 调用访问列表之前，先配置好访问列表。

总步骤

1. enable

2. configure terminal

3. ip pim ssm [**default** | **range** *access-list*]

4. interface *type number*

5. ip pim {**sparse-mode** | **sparse-dense-mode**}

6. ip igmp version 3

7. end

8. show running-config

9. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密码

	示例： `Device>` **enable**	
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip pim ssm** [**default** | **range** *access-list*] 示例： `Device(config)#` **ip pim ssm range 20**	定义 IP 组播地址的 SSM 范围
步骤 4	**interface** *type number* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	指定与可以启用 IGMPv3 的主机相连的接口，并进入接口配置模式。 设置的接口必须是下面两类之一： • 路由端口：通过在接口配置模式下输入命令 **no switchport**，设置为三层端口的物理端口。用户还需要在这个接口上启用 IP PIM 稀疏-密集模式，并将这个接口作为静态连接的组成员加入 IGMP 静态组当中； • SVI：使用全局配置命令 **interface vlan** *vlan-id* 创建的 VLAN 接口。用户还需要在这个 VLAN 上启用 IP PIM 稀疏-密集模式，并将这个 VLAN 作为静态连接的组成员加入 IGMP 静态组当中，然后在这个 VLAN、IGMP 静态组和物理接口上启用 IGMP snooping。 用户必须给这些接口分配 IP 地址
步骤 5	**ip pim** {**sparse-mode** | **sparse-dense-mode**} 示例： `Device(config-if)#` **ip pim sparse-dense-mode**	在接口上启用 PIM。用户必须使用稀疏模式或者稀疏-密集模式
步骤 6	**ip igmp version 3** 示例： `Device(config-if)# ip igmp version 3`	在这个接口上启用 IGMPv3。IGMP 默认的版本是 IGMP 第 2 版。
步骤 7	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 8	**show running-config**	查看配置的条目

	示例： `Device#` **show running-config**	
步骤 9	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置特定源组播映射

在端系统无法支持 SSM 或由于管理和技术原因不应该支持 SSM 时，特定源组播（SSM）映射特性可以实现 SSM 转换。用户可以使用 SSM 映射来利用 SSM 向不支持 IGMPv3 的传统 STB 或者没有使用 IGMPv3 主机栈的应用实现视频传输。

配置静态 SSM 映射（CLI）

用户可以按照下面的步骤来配置静态 SSM 映射：

总步骤

1. enable

2. configure terminal

3. ip igmp ssm-map enable

4. no ip igmp ssm-map query dns

5. ip igmp ssm-map static *access-list source-address*

6. end

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip igmp ssm-map enable** 示例： `Device(config)# ip igmp ssm-map enable`	对配置的 SSM 范围内的组启用 SSM 映射。 **注释：**在默认情况下，这条命令会启用基于 DNS 的 SSM 映射
步骤 4	**no ip igmp ssm-map query dns**	（可选）禁用基于 DNS 的 SSM 映射。 **注释：**如果用户希望完全依靠静态 SSM 映射，那就可以禁用基于 DNS 的 SSM 映射。在默认情况

	示例： `Device(config)# no ip igmp ssm-map query dns`	下，输入命令 **ip igmp ssm-map** 会启用基于 DNS 的 SSM 映射
步骤 5	**ip igmp ssm-map static** *access-list source-address* 示例： `Device(config)# ip igmp ssm-map static 11 172.16.8.11`	配置静态 SSM 映射。 • 参数 *access-list* 所指定的 ACL，用来定义要与参数 *source-address* 指定的源 IP 地址进行映射的组。 **注释**：用户可以配置多个静态 SSM 映射。如果用户配置了多个 SSM 映射，而路由器又接收到了发送给 SSM 范围中一个组的 IGMPv1 或 IGMPv2 成员关系报告，那么设备就会查看每条用户配置的 ip igmp ssm-map static 命令，来判断这个组关联的源。设备每个组可以关联最多 20 个源。 用户可以根据需要，重复这一步来配置多个静态 SSM 映射
步骤 6	**ip igmp version 3** 示例： `Device(config-if)# ip igmp version 3`	在这个接口上启用 IGMPv3。IGMP 默认的版本是 IGMP 第 2 版。
步骤 7	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 8	**show running-config** 示例： `Device# show running-config`	查看配置的条目
步骤 9	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中

配置基于 DNS 的 SSM 映射（CLI）

要配置基于 DNS 的 SSM 映射，用户需要创建一个 DNS 服务器区域或者向当前区域中添加记录。如果路由器在使用基于 DNS 的 SSM 映射的同时，也用 DNS 来满足其他需求，那么用户就应该使用一台正常配置的 DNS 服务器。如果基于 DNS 的 SSM 映射是唯一需要在路由器上实施的 DNS 配置，那么用户可以执行伪 DNS 设置，即根区域为空，或者将根区域指向自己。

总步骤

1. enable

2. configure terminal

3. ip igmp ssm-map enable

4. ip igmp ssm-map query dns

5. ip domain multicast *domain-prefix*

6. ip name-server *server-address1* [*server-address2...server-address6*]

7. 根据需要，重复步骤6来配置其他DNS服务器以提供冗余

8. end

9. show running-config

10. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip igmp ssm-map enable** 示例： `Device(config)# ip igmp ssm-map enable`	对配置的 SSM 范围内的组启用 SSM 映射。
步骤 4	**ip igmp ssm-map query dns** 示例： `Device(config)# ip igmp ssm-map query dns`	（可选）启用基于 DNS 的 SSM 映射。 • 在默认情况下，输入命令 **ip igmp ssm-map** 会启用基于 DNS 的 SSM 映射。只有这条命令的 **no** 形式，才会保存到运行配置当中。 注释：如果用户禁用了基于 DNS 的 SSM 映射，用户可以使用这条命令来重新启用基于 DNS 的 SSM 映射
步骤 5	**ip domain multicast** *domain-prefix* 示例： `Device(config)# ip domain multicast ssm-map.icntnetworks.com`	（可选）修改用于基于 DNS 的 SSM 映射的域前缀。 • 在默认情况下，软件会使用 ip-addr.arpa 这个域前缀。
步骤 6	**ip name-server** *server-address1* [*server-address2...server-address6*] 示例： `Device(config)# ip name-server 10.48.81.21`	设置用于执行域名解析和地址解析的一台或多台域名服务器的地址
步骤 7	根据需要，重复步骤6来配置其他DNS服务器以提供冗	——

	余	
步骤 8	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 9	**show running-config** 示例： `Device# show running-config`	查看配置的条目
步骤 10	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中

通过 SSM 映射来配置静态流量转发（CLI）

用户可以按照下面的步骤在最后一跳路由器上配置 SSM 映射以实现静态流量转发：

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. ip igmp static-group *group-address* **source ssm-map**

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *type number* 示例： `Device(config)# interface gigabitethernet 1/0/1`	选择使用 SSM 映射静态向组播组转发流量的接口，并进入接口配置模式。 设置的接口必须是下面两类之一： • 路由端口：通过在接口配置模式下输入命令 **no switchport**，设置为三层端口的物理端口。用户还需要在这个接口上启用 IP PIM 稀疏-密集模式，并将这个接口作为静态连接的组成员加入 IGMP 静态组当中； • SVI：使用全局配置命令 **interface vlan** *vlan-id*

		创建的 VLAN 接口。用户还需要在这个 VLAN 上启用 IP PIM 稀疏-密集模式，并将这个 VLAN 作为静态连接的组成员加入 IGMP 静态组当中，然后在这个 VLAN、IGMP 静态组和物理接口上启用 IGMP snooping。 用户必须给这些接口分配 IP 地址。 **注释：**使用 SSM 映射来静态转发流量可以通过基于 DNS 的 SSM 映射，或者通过静态配置的 SS 映射来实现
步骤 4	**ip igmp static-group** *group-address* **source ssm-map** 示例： `Device(config-if)#` **ip igmp static-group 239.1.2.1 source ssm-map**	配置 SSM 映射，从这个接口静态转发(S,G)信道。如果用户希望静态向某个组转发 SSM 流量，就应当配置这条命令。要判断信道的源地址，需要使用基于 DNS 的 SSM 映射
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** 示例： `Device#` **show running-config**	查看配置的条目
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

监控 SSM

用户可以使用下表中的 EXEC 命令来监控 SSM。

表 46：监控 SSM 的命令

命令	目的
show ip igmp groups detail	显示通过 IGMPv3 注册的(S,G)信道
show ip mroute	显示组播组是否支持 SSM 服务，或者是否接收到了特定源的主机报告

监控 SSM 映射

用户可以使用下表中的 EXEC 命令来监控 SSM 映射。

表 47：监控 SSM 映射的命令

命令	目的
Device# **show ip igmp ssm-mapping**	显示关于 SSM 映射的信息
Device#**show ip igmp ssm-mapping** *group-address*	显示 SSM 映射某个特定组的源
Device#**show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**]	显示那些拥有与路由器直连、并且是通过 IGMP 学习过来的源的接收方
Device#**show host**	显示默认域名、域名查询服务的类型、域名服务器主机列表，以及缓存的主机名与地址的列表
Device#**debug ip igmp** *group-address*	显示接收和发送的 IGMP 数据包，以及与 IGMP 主机相关的事件

配置完 SSM 接下来做什么

用户还可以配置：

- IGMP
- PIM
- IP 组播路由
- 服务发现网关

其他参考资料

相关文档

相关主题	文档名
SSM 与其他可用命令	《IP 组播命令参考手册，Inspur INOS XE 3SE 版（Inspur 3850 交换机）》
平台独立配置信息	• 《IP 组播：PIM 配置指南，Inspur XE 3SE 版（Inspur 3850 交换机）》 • 《IP 组播：IGMP 配置指南，Inspur XE 3SE 版（Inspur 3850 交换机）》 • 《IP 组播：组播优化配置指南，Inspur INOS 3SE 版（Inspur 3850 交换机）》

标准与 RFC

标准/RFC	标题
RFC 4601	协议独立组播稀疏模式（PIM-SM）：协议标准

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档	http://www.icntnetworks.com

和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	

组播源发现协议的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

配置基本 IP 组播路由

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

基本 IP 组播路由的前提条件

下面是配置基本 IP 组播路由的前提条件：

- 用户要想使用这项特性，设备或主用设备必须运行 IP Services 特性集。IP Services 镜像会包含完整的组播路由；

- 用户必须配置 PIM 版本和 PIM 模式以执行 IP 组播路由。交换机会创建自己的组播路由表，并且会根据模式的设定来转发它从直连局域网那里接收到的组播数据包。用户可以将接口配置为 PIM 密集模式、稀疏模式或者稀疏-密集模式；
- 在接口上启用 PIM 也会在这个接口上启用 IGMP 操作；（要参与 IP 组播，组播主机、路由器和多层设备必须能够执行 IGMP 操作）
  如果在多个接口上启用 PIM，那么当大多数接口都不在出站接口列表当中，并且用户禁用了 IGMP snooping 时，出站接口有可能无法维持以线速发送组播流量，因为这会导致组播流量被超量复制。

基本 IP 组播路由的限制条件

下面是 IP 组播路由的限制条件：
- 运行 LAN Base 特性集的交换机不支持 IP 组播路由；
- 用户不能将 Inspur 3850 和 Inspur 6650 设备混合部署在一个设备堆栈中

关于基本 IP 组播路由的信息

IP 组播是一种有效利用网络资源的方式，尤其适用于带宽密集型服务，如音频和视频类服务。IP 组播路由可以让一台主机（源）使用一种特殊形式的 IP 地址（称为 IP 组播组地址），向 IP 网络中其他地方的一组主机（接收方）发送数据包。
发送方主机会将组播组地址插入数据包的 IP 目的地址字段，而 IP 组播路由器和多层设备会将入站 IP 组播数据包从所有连接了组播组成员的接口发送出去。任何主机，无论是不是组成员，都可以向一个组发送数据包。但只有组成员可以接收到消息。

组播转发信息库概述

设备会使用组播转发信息库（MFIB）架构和组播路由信息库（MRIB）来转发 IP 组播。
MFIB 架构会在组播控制平面（协议独立组播[PIM]和互联网组管理协议[IGMP]）和组播转发平面（MFIB）提供模块化与隔离。这个架构会用于 Inspur INOS 的 IPv6 组播实施。
MFIB 自身是一种独立于组播路由协议的转发引擎。也就是说，这种转发引擎不依赖 PIM 或其他任何组播路由协议。它会负责：
- 转发组播数据包；
- 向 MRIB 注册，以学习条目和控制平面设置的接口标记；
- 处理那些必须发送给控制平面的、因数据导致的事件；
- 记录接收、丢弃和转发的组播数据包的数量、速率与字节数；

MRIB 是 MRIB 客户端之间的通信信道。MRIB 客户端的示例包括 PIM、IGMP 和组播路由表，以及 MFIB。

组播路由与设备堆栈

对于所有组播路由协议，整个堆栈会充当网络中的一台路由器，并且像一台组播路由器那样进行工作。

在设备堆栈中，主用设备会执行下列功能：

- 负责执行堆栈 IP 组播路由的功能。它会完整地执行初始化并且运行 IP 组播路由协议；
- 它会给整个堆栈建立并维护组播路由表；
- 它会负责将组播路由表分发给所有堆栈成员。

堆栈成员会执行下面的功能：

- 它们会充当组播路由备用设备，并且准备好在主用设备出现故障时，接替主用设备的功能。如果主用设备出现了故障，所有堆栈成员都会删除自己的组播路由表。新选举出来的主用设备会开始建立路由表，并且将它们分发给堆栈成员；
- 它们不会建立组播路由表，而只会使用主用设备分发过来的组播路由表。

默认的 IP 组播路由配置

这张表显示了默认的 IP 组播路由配置。

表 48：默认的 IP 组播路由配置

特性	**默认设置**
组播路由	所有接口上都禁用
PIM 版本	第 2 版
PIM 模式	未定义模式
PIM 末节路由	未配置
PIM RP 地址	未配置
PIM 域边界	禁用
PIM 组播边界	无
候选 BSR	禁用
候选 RP	禁用
最短路径树门限值	0kb/s
PIM 路由器查询消息时间间隔	30 秒

如何配置基本 IP 组播路由

配置基本 IP 组播路由

在默认情况下，组播路由是禁用的，设备上没有设置默认的模式。

这个流程是必需的。

在开始前

用户必须配置 PIM 版本和 PIM 模式。交换机会创建自己的组播路由表，并且会根据模式的设定来转发它从直连局域网那里接收到的组播数据包。

在创建组播路由表时，设备会将密集模式的接口添加到表中。只有当周期性的加入消息是从下游设备那里接收到的，或者当接口上有直连成员时，设备才会将稀疏模式的接口添加到表中。在从 LAN 执行转发时，如果设备知道这个组的 RP，那么它就会执行稀疏模式的操作。若如此，设备就会封装数据包，并且将其发送给 RP。如果设备不知道 RP，那么数据包就会按照密集模式的方式进行泛洪。如果特定源发送的组播流量充足，那么接收方的第一跳路由器可能会向源发送加入消息，来建立基于源的分发树。

总步骤

1. enable

2. configure terminal

3. ip multicast-routing

4. interface *interface-id*

5. ip pim {**dense-mode** | **sparse-mode** | **sparse-dense-mode**}

6. end

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip multicast-routing** 示例： `Device(config)#` **ip multicast-routing**	启用 IP 组播路由。 设备会通过组播转发信息库（MFIB）和组播路由信息库（MRIB）来支持 IP 组播路由
步骤 4	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	选择使用 SSM 映射静态向组播组转发流量的接口，并进入接口配置模式。 设置的接口必须是下面两类之一： • 路由端口：通过在接口配置模式下输入命令 **no switchport**，设置为三层端口的物理端口。用户还需要在这个接口上启用 IP PIM 稀疏-密集模式，并将这个接口作为静态连接的组成员加入 IGMP 静态组当中； • SVI：使用全局配置命令 **interface vlan** *vlan-id* 创建的 VLAN 接口。用户还需要在这个 VLAN 上启用 IP PIM 稀疏-密集模式，并将这个 VLAN 作为静态连接的组成员加入 IGMP 静态组当中，然后在这个 VLAN、IGMP 静态组和物理接口上启用 IGMP snooping。 用户必须给这些接口分配 IP 地址。
步骤 5	**ip pim** {**dense-mode**	**sparse-mode**	**sparse-dense-mode**} 示例： `Device(config-if)#` **ip pim**	在接口上启用 PIM 模式。 在默认情况下，接口上没有配置模式。 文中关键字的含义如下： • **dense-mode**：启用密集模式的操作； • **sparse-mode**：启用稀疏模式的操作。如果配

	sparse-dense-mode	置稀疏模式，就必须配置 RP； • **sparse-dense-mode**：让接口按照所在组的模式进行操作。稀疏-密集模式是推荐的设置。 **注释**：要在接口上禁用 PIM，需要使用接口配置命令 **no ip pim**
步骤 6	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 7	**show running-config** 示例： `Device# show running-config`	查看配置的条目
步骤 8	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中

配置 IP 组播转发（CLI）

用户可以使用下面的流程在设备上配置 IPv4 组播转发信息库（MFIB）中断级入站数据包或出站数据包的 IP 组播转发。

注释： 在通过命令 **ip multicast-routing** 启用了 IP 组播路由之后，IPv4 组播转发也会被启用。由于 IPv4 组播转发默认就是启用的，因此用户可以在命令 **ip mfib** 前面添加 no 来禁用 IPv4 组播转发。

总步骤

1. enable

2. configure terminal

3. ip mfib

4. exit

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例：	进入全局配置模式

	`Device#` **configure terminal**	
步骤 3	**ip mfib** 示例： `Device(config)#` **ip mfib**	启用 IP 组播转发
步骤 4	**exit** 示例： `Device(config)#` **exit**	返回特权 EXEC 模式
步骤 5	**show running-config** 示例： `Device#` **show running-config**	查看配置的条目
步骤 6	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置静态组播路由（mroute）（CLI）

用户可以使用下面的流程配置静态路由。静态组播路由与单播静态路由类似，但两者存在下列区别：

- 静态组播路由是用来计算 RPF 信息，而不是转发流量的；
- 静态组播路由不能重分发。

静态组播路由是在所定义的设备本地的。由于协议独立组播（PIM）没有自己的路由协议，所以没有机制可以在网络中分发静态组播路由。结果是，静态组播路由的管理往往比管理单播静态路由更加复杂。

在配置好静态组播路由之后，这些路由会被存在设备的一个独立的表中，这个表称为静态组播路由表。在配置时，输入命令 **ip mroute** 之后，设备就会按照用户通过源地址和掩码参数设置的源地址或源地址范围，将静态组播路由添加到静态组播路由表中。当源匹配源地址或落在用户通过源地址参数设置的源地址范围内，那么这个源就会 RPF 到用户通过参数 *rpf-address* 设置的接口 IP 地址，或者通过参数 *interface-type* 和 *interface-number* 指定的设备本地接口。如果用户通过参数 *rpf-address* 设置了 IP 地址，那么设备就会在单播路由表中对这个地址执行递归查询，以找到直连的邻居。

如果用户配置了多条静态组播路由，那么设备就会对组播路由表执行最长匹配查询。当设备在组播路由表中找到了最长匹配（的源地址），那么设备就会停止搜索，并且使用匹配的静态组播路由。配置静态组播路由的顺序并不重要。

用户可以使用可选的距离参数，来设置组播路由的管理距离。如果没有通过距离参数设置数值，那么组播路由器的距离就会被设置为 0。如果静态组播路由的距离与另一个 RPF 源相同，那么静态组播路由更优。这条规则仅有两个例外：直连路由和默认单播路由。

总步骤

1. enable

2. configure terminal

3. ip mroute [**vrf** *vrf-name*] *source-address mask* { **fallback-lookup** {**global** | **vrf** *vrf-name* }[*protocol*] {*rpf-address* | *interface-type interface-number*}} [**distance**]

4. exit

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip mroute** [**vrf** *vrf-name*] *source-address mask* { **fallback-lookup** {**global** | **vrf** *vrf-name* }[*protocol*] {*rpf-address* | *interface-type interface-number*}} [**distance**] 示例： `Device(configure)#` **ip mroute 10.1.1.1 255.255.255.255 10.2.2.2**	将源 IP 地址配置为通过 IP 地址 10.2.2.2 对应的接口可达
步骤 4	**exit** 示例： `Device(config)#` **exit**	返回特权 EXEC 模式
步骤 5	**show running-config** 示例： `Device#` **show running-config**	查看配置的条目
步骤 6	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 IP 组播路由的可选特性

定义 IP 组播边界（CLI）

用户可以定义组播边界来防止 Auto-RP 消息进入 PIM 域当中。用户可以创建一个访问列表来拒绝去往 224.0.1.39 和 224.0.1.40 的数据包，这些数据包会携带 Auto-RP 的信息。
这个流程是可选的。

总步骤

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **deny** *source* [*source-wildcard*]
4. **interface** *interface-id*
5. **ip multicast boundary** *access-list-number*
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**access-list** *access-list-number* **deny** *source* [*source-wildcard*] 示例： `Device(config)#` **access-list 12 deny 224.0.1.39** **access-list 12 deny 224.0.1.40**	创建一个标准访问列表，用户应根据需要重复配置这条命令。 • 在 *access-list-number* 部分，取值范围是 1 到 99； • 关键字 **deny** 是指在条件匹配时，即拒绝访问； • 在 *source* 部分，输入组地址 224.0.1.39 和 224.0.1.40，这些组携带 Auto-RP 信息； • （可选）在 *source-wildcard* 部分，用点分十进制格式输入要应用于源的反掩码，配置反掩码应在要忽略的位取 0。 访问列表的最后永远包含一条隐式的全部拒绝语句
步骤 4	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	指定要配置的接口，并进入接口配置模式。 指定的接口必须为下列接口之一： • 路由端口：通过接口配置命令 **no switchport** 被配置为了三层端口的物理端口。用户还应该在这个接口上启用 IP PIM 稀疏-密集模式，

		并让这个接口作为静态连接成员加入一个 IGMP 静态组中； • SVI：通过全局配置命令 **interface vlan** *vlan-id* 创建的 VLAN 接口。用户还应该在这个 VLAN 上启用 IP PIM 稀疏-密集模式，让这个接口作为静态连接成员加入一个 IGMP 静态组中，并且在 VLAN、IGMP 静态组和物理接口上启用 IGMP snooping； 这些接口上必须配置 IP 地址。
步骤 5	**ip multicast boundary** *access-list-number* 示例： `Device(config-if)#` **ip multicast boundary 12**	配置边界，指定在步骤 2 创建的访问列表
步骤 6	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 7	**show running-config** 示例： `Device#` **show running-config**	查看配置的条目
步骤 8	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置对 sdr 侦听器的支持

启用对 sdr 侦听器的支持（CLI）

在默认条件下，设备不会侦听会话目录通告。

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. ip sap listen

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密码

	示例： `Device>` **enable**	
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	指定要配置的接口，并进入接口配置模式。 指定的接口必须为下列接口之一： • 路由端口：通过接口配置命令 **no switchport** 被配置为了三层端口的物理端口。用户还应该在这个接口上启用 IP PIM 稀疏-密集模式，并让这个接口作为静态连接成员加入一个 IGMP 静态组中。如需查看配置示例，可以参见示例：将接口配置为路由端口； • SVI：通过全局配置命令 **interface vlan** *vlan-id* 创建的 VLAN 接口。用户还应该在这个 VLAN 上启用 IP PIM 稀疏-密集模式，让这个接口作为静态连接成员加入一个 IGMP 静态组中，并且在 VLAN、IGMP 静态组和物理接口上启用 IGMP snooping。如需查看配置示例，可以参见示例：将接口配置为 SVI，这些接口上必须配置 IP 地址。
步骤 4	**ip sap listen** 示例： `Device(config-if)#` **ip sap listen**	启用设备软件来侦听会话目录通告
步骤 5	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式
步骤 6	**show running-config** 示例： `Device#` **show running-config**	查看配置的条目
步骤 7	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

限制 sdr 缓存条目存在的时间（CLI）

在默认情况下，条目永远不会从 sdr 缓存中被删除。用户可以限制条目保持活动的时间，这

样如果源停止通告 SAP 信息，老的通告就不会保持。
这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. ip sap cache-timeout *minutes*

4. end

5. show running-config

6. show ip sap

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip sap cache-timeout** *minutes* 示例： `Device(config)# ip sap cache-timeout 30`	限制会话通告协议（SAP）缓存在缓存中保持活动状态的时间。 在默认情况下，条目永远不会从缓存中被删除。 在 *minutes* 部分，取值范围是从 1 到 1440 分钟（24 小时）
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** 示例： `Device# show running-config`	查看配置的条目
步骤 6	**show ip sap** 示例： `Device# show ip sap`	显示 SAP 缓存
步骤 7	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）将输入的条目保存到配置文件中

基本 IP 组播路由的监控与维护

清除缓存、表与数据库

用户可以清楚一个缓存、表或数据库中的所有内容。当结构中的内容已经失效或者疑似失效时，用户可能就会需要清除缓存、表或数据库中的内容。
用户可以使用下表中的任意特权 EXEC 命令来清除 IP 组播缓存、表和数据库。

表 49：清除缓存、表和数据库的命令

命令	目的
clear ip igmp group {**group** [*hostname* | *IP address*] | **vrf** *name* **group** [*hostname* | *IP address*] }	从 IGMP 缓存中删除条目
clear ip mfib { **counters** [*group* | *source*] | **global counters** [*group* | *source*] | **vrf *** }	清除所有活动 IPv4 组播转发信息库（MFIB）流量计数器
clear ip mrm {**status-report** [*source*] }	IP 组播路由清除命令
clear ip mroute { ***** | [*hostname* | *IP address*] | **vrf** *name* **group** [*hostname* | *IP address*] }	从 IP 组播路由表中删除条目
clear ip msdp { **peer** | **sa-cache** | **statistics** | **vrf** }	清除组播源发现协议（MSDP）缓存
clear ip multicast { **limit** | **redundancy statistics** }	清楚 IP 组播信息
clear ip pim { **df** [**int** | **rp** *rp address*] | **interface** | **rp-mapping** [*rp address*] | **vrf** *vpn name* { **df** | **interface** | **rp-mapping** }	清楚 PIM 缓存
clear ip sap [*group-address* | *"session-name"*]	删除会话目录协议第 2 版缓存或一个 sdr 缓存条目

显示系统和网络统计数据

用户可以查看特定的统计数据，譬如 IP 路由表、缓存和数据库中的内容。
注释： 这个版本不支持针对每条路由显示统计数据。
用户可以查看信息，来了解资源的使用情况，并且解决网络的问题。用户也可以显示关于节点可达性的信息，并且观察设备中的数据包选择了什么样的路由路径。
用户可以使用下表中的特权 EXEC 命令来查看各类路由统计数据。

表 50：查看系统与网络统计数据的命令

命令	目的
ping [*group-name* | *group-address*]	向组播组地址发送 ICMP Echo 请求消息
show ip igmp filter	显示 IGMP 过滤器的信息
show ip igmp groups [*group-name*|*group-address*|*type-number*]	显示与设备直连并且通过 IGMP 学习到的组播组
show ip igmp interface [*type number*]	显示一个接口与组播相关的信息
show ip igmp profile [profile_number]	显示 IGMP 配置文件信息

show ip igmp ssm-mapping [*hostname/IP address*]	显示 IGMP SSM 映射信息
show ip igmp static-group {**class-map** [**interface** [type]]	显示静态组信息
show ip igmp membership [*name/group address* | **all** | **tracked**]	显示用来执行转发的 IGMP 成员关系信息
show ip igmp vrf	显示通过域名选择的 VPN 路由/转发实例
show ip mfib [*type number*]	显示 IP 组播转发信息库
show ip mrib { **client** | **route** | **vrf** }	显示组播路由转发信息库
show ip mrm { **interface** | **manager** | **status-report** }	显示 IP 组播路由监控信息
show ip mroute [*group-name* | *group-address*] [*source*] [**count** | **interface** | **proxy** | **pruned** | **summary** | **verbose**]	显示 IP 组播路由表中的内容
show ip msdp { **count** | **peer** | **rpf-peer** | **sa-cache** | **summary** | **vrf** }	显示组播源发现协议（MSDP）信息
show ip multicast [**interface** | **limit** | **mpls** | **redundancy** | **vrf**]	显示全局组播信息
show ip pim all-vrfs { **tunnel** }	显示所有 VRF
show ip pim autorp	显示全局 Auto-RP 信息
show ip pim boundary [*type number*]	显示边界信息
show ip pim bsr-router	显示自举路由器信息（第 2 版）
show ip pim interface [*type number*] [**count** | **detail** | **df** | **stats**]	显示关于配置了 PIM 的接口信息。这条命令所有软件版本都可用
show ip pim neighbor [*type number*]	列出设备发现的 PIM 邻居。这条命令所有软件版本都可用
show ip pim mdt [**bgp**]	显示组播隧道信息
show ip pim rp [*group-name* | *group-address*]	显示与稀疏模式组播组相关联的 RP 路由器。这条命令所有软件版本都可用
show ip pim rp-hash [*group-name* | *group-address*]	显示指定组选出的 RP
show ip pim tunnel [*tunnel* | *verbose*]	显示注册的隧道
show ip pim vrf *name*	显示 VPN 路由与转发实例
show ip rpf {*source-address* | *name*}	显示设备执行逆向路径转发的方式（也就是从单播路由表、DMVPN 路由表或静态组播路由） 命令参数包括： • *Host name* 或 *IP address*：IP 主机名或组地址； • *Select*：基于组的 VRF 选择信息 • **vrf**：选择 VPN 路由/转发实例
show ip sap [*group* | *"session-name"* | **detail**]	显示会话通告协议（SAP）第 2 版的缓存。 命令参数包括：

	• *A.B.C.D*：IP 组地址 • *WORD*：会话名称（要输入双引号） • **detail**：会话具体信息

查看组播对等体、数据包速率与丢包信息，以及路径追踪

用户可以使用下表中的特权 EXEC 命令来监控 IP 组播路由器、数据包与路径。

表 51：查看组播对等体、数据包速率与丢包信息以及路径追踪的命令

命令	目的
mrinfo { [*hostname* | *address*] | **vrf** }	向组播路由器或多层设备查询哪些邻居组播设备与自己建立了对等体关系
mstat { [*hostname* | *address*] | **vrf** }	显示 IP 组播数据包速率和丢包信息
mtrace { [*hostname* | *address*] | **vrf** }	追踪给定组的组播分发树从源到目的叶网络的路径

IP 组播路由的配置示例

示例：配置 IP 组播边界

这个示例显示了如何给所有管理作用范围的地址设置边界：

```
Device(config)# access-list 1 deny 239.0.0.0 0.255.255.255
Device(config)# access-list 1 permit 224.0.0.0 15.255.255.255
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip multicast boundary 1
```

示例：响应 mrinfo 请求

软件会响应路由系统、Inspur 路由器和多层设备发送的 mrinfo 请求。软件会返回关于通过 DMVRP 隧道和所有路由接口连接的邻居的信息。这些信息包括度量值（始终为 1）、配置的 TTL 门限值、接口的状态和各类标记。用户也可以使用特权 EXEC 命令 **mrinfo** 来查询路由器或设备自身，如下例所示：

```
Device# mrinfo
171.69.214.27 (mm1-7kd.icntnetworks.com) [version inspur 11.1]
[flags: PMS]: 171.69.214.27 -> 171.69.214.26 (mm1-
r7kb.icntnetworks.com) [1/0/pim/querier] 171.69.214.27 ->
171.69.214.25 (mm1-45a.icntnetworks.com) [1/0/pim/querier]
171.69.214.33 -> 171.69.214.34 (mm1-45c.icntnetworks.com) [1/0/pim]
171.69.214.137 -> 0.0.0.0 [1/0/pim/querier/down/leaf]
171.69.214.203 -> 0.0.0.0 [1/0/pim/querier/down/leaf]
171.69.214.18 -> 171.69.214.20 (mm1-45e.icntnetworks.com) [1/0/pim]
171.69.214.18 -> 171.69.214.19 (mm1-45c.icntnetworks.com) [1/0/pim]
171.69.214.18 -> 171.69.214.17 (mm1-45a.icntnetworks.com) [1/0/pim]
```

其他参考资料

相关文档

相关主题	文档名
如需了解本章所述命令的完整语法结构及使用信息	《IP 组播路由命令参考手册（Inspur 6650 交换机）》
如需了解配置组播源发现协议（MSDP）的信息	《路由命令参考手册（Inspur 6650 交换机）》
平台独立配置信息	• IP 组播：PIM 配置指南，Inspur INOS XE 3SE 版（Inspur 6650 交换机） • IP 组播：IGMP 配置指南，Inspur INOS XE 3SE 版（Inspur 6650 交换机） • IP 组播：组播优化配置指南，Inspur INOS XE 3SE 版（Inspur 6650 交换机）
Inspur INOS 命令	《Inspur INOS 主命令列表，所有版本》
Inspur INOS IP SLA 命令	《Inspur INOS IP 组播命令参考手册》

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com

标准与 RFC

标准/RFC	标题
RFC 1112	IP 组播转发的主机扩展
RFC 2236	互联网组管理协议，第 2 版
RFC 4601	协议独立组播稀疏模式（PIM-SM）：协议标准

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

组播源发现协议的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

配置基于 GRE 隧道的组播路由

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

基于 GRE 隧道的组播路由的前提条件

在配置基于 GRE 的组播路由之前，用户应该熟悉 IP 组播路由技术和 GRE 隧道技术的相关概念。

基于 GRE 隧道的组播路由的限制条件

下面是配置基于 GRE 隧道的组播路由时的限制条件：

- 基于 GRE 隧道的 IPv6 组播是不支持的；
- 全部隧道支持的组播路由总数为 2000；
- 不支持双向 PIM；
- 要在第一跳路由器（FHR）、汇集点（RP）和最后一跳路由器（LHR）上配置组播路由，以支持基于 GRE 隧道的组播；
- 在 Inspur 3850 和 Inspur 6650 系列交换机上，隧道源可以是环回接口、物理接口或三层 EtherChannel 接口；

- GRE 隧道上不支持与诸如 IPSec、ACL、隧道计数器、加密、分片、Inspur 发现协议（CDP）、QoS、GRE 保活、多点 GRE 等特性实现互操作。

关于基于 GRE 隧道的组播路由的信息

本章描述了如何配置通用路由封装（GRE）隧道来封装非 IP 组播区域之间的 IP 组播数据包。IP 组播流量的好处在于，流量可以穿越一段不支持 IP 组播的区域来从源向组播组发送流量。通过 GRE 隧道发送组播路由的技术，支持 ip pim dense-mode、sparse-dense mode、sparse mode 和 pim-ssm mode，且支持静态 RP 和 Auto-RP。用户可以参考关于汇集点和 Auto-RP 的介绍，来了解关于配置静态 RP 和 Auto-RP 的信息。

注释： 从 Inspur INOS Denali 16.3.1 开始，GRE 隧道可以支持组播路由和 NHRP。用户可以根据需要在配置隧道接口组播时配置 NHRP，以便实现隧道端点的动态实现。用户可以参考关于 NHRP 的介绍，来了解关于在隧道接口上配置 NHRP 的信息。

用隧道来连接非 IP 组播区域的好处

如果用户配置了一条隧道，在不支持组播路由的媒介上，为源和目的传输 IP 组播数据包。

总步骤

1. enable

2. configure terminal

3. ip multicast-routing

4. interface tunnel *number*

5. ip address ip_address *subnet_mask*

6. ip pim { **sparse-dense-mode** | **sparse-mode** | **dense-mode** }

7. tunnel source { *ip-address* | *interface-name* }

8. tunnel destination { *hostname* | *ip-address* }

9. end

10. show interface *type number*

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip multicast-routing** 示例： `Device(config)# ip multicast-routing`	启用 IP 组播路由
步骤 4	**interface tunnel** *number*	进入隧道接口配置模式

	示例： `Device(config)# interface tunnel 0`	
步骤 5	**ip address** *ip-address network-mask* 示例： `Device(config-if)# ip address 192.168.24.1 255.255.255.252`	配置 IP 地址和 IP 子网
步骤 6	**ip pim { sparse-dense-mode | sparse-mode | dense-mode }** 示例： `Device(config-if)# ip pim sparse-dense-mode`	在隧道接口上启用协议独立组播（PIM），并且执行下列操作模式之一： • **sparse-dense-mode**：按照稀疏模式或密集模式的方式执行处理，具体模式取决于组播组的操作模式； • **sparse-mode**：启用稀疏模式的操作 • **dense-mode**：启用密集模式的操作
步骤 7	**tunnel source** { *ip-address* | *interface-name* } 示例： `Device(config-if)# tunnel source 100.1.1.1`	配置隧道源
步骤 8	**tunnel destination** { *hostname* | *ip-address* } 示例： `Device(config-if)# tunnel destination 100.1.5.3`	配置隧道目的
步骤 9	**end** 示例： `Device(config-if)# end`	离开接口配置模式并返回特权 EXEC 模式
步骤 10	**show interface** *type number* 示例： `Device# show interface tunnel 0`	显示隧道接口信息

使用隧道连接非 IP 组播区域的示例

下图所示为 Inspur 6650/3850 交换机通过 GRE 隧道执行组播路由转发的示例。

Mcast Src Grp: 239.1.1.20	组播源 组 239.1.1.20
Mcast Receiver Grp: 239.1.1.20	组播接收方 组 239.1.1.20
IP cloud	IP 云
Catalyst 3850/3650 Switch-1	Inspur 6650/3850 交换机 1
Catalyst 3850/3650 Switch-2	Inspur 6650/3850 交换机 2

在上图中，组播源（10.1.1.1）与 Inspur 3850/6650 交换机 1 相连，该源需要向组播组 239.1.1.20 发送流量。组播接收方（10.2.2.3）与 Inspur 3850/6650 交换机 2 相连，该源会接收发送给组播组 239.1.1.20 的流量。IP 云隔开了交换机 1 和交换机 2，而 IP 云没有配置组播路由。

通过配置，交换机 1 和交换机 2 会以自己的环回接口作为消息源。交换机 1 和交换机 2 上启用了组播路由。用户在接口上配置了命令 **ip pim sparse-dense-mode** 来支持稀疏模式或密集模式。在隧道接口上配置稀疏-密集模式，可以依靠组汇集点（RP）的配置，通过隧道来转发稀疏模式或密集模式数据包。

交换机 1 的配置：

```
Device(config)# ip multicast-routing
Device(config)# interface Loopback0 //Tunnel source interface
Device(config-if)# ip address 2.2.2.2 255.255.255.255
Device(config)# interface Tunnel 10 //Tunnel interface configured for PIM traffic
Device(config-if)# ip address 192.168.24.1 255.255.255.252
Device(config-if)# ip pim sparse-dense-mode
Device(config-if)# ip nhrp map 192.168.24.3 4.4.4.4 //NHRP may optionally be configured to dynamically discover tunnel end points.
Device(config-if)# ip nhrp map multicast 4.4.4.4
Device(config-if)# ip nhrp network-id 1
Device(config-if)# ip nhrp nhs 192.168.24.3
Device(config-if)# tunnel source Loopback0
Device(config-if)# tunnel destination 4.4.4.4
Device(config)# interface GigabitEthernet 0/0/0 //Source interface
Device(config-if)# ip address 10.1.1.2 255.255.255.0
Device(config-if)# ip pim sparse-dense-mode
```

交换机 2 的配置：

```
Device(config)# ip multicast-routing
Device(config)# interface Loopback0 //Tunnel source interface
Device(config-if)# ip address 4.4.4.4 255.255.255.255
Device(config)# interface Tunnel 10 //Tunnel interface configured for PIM traffic
Device(config-if)# ip address 192.168.24.2 255.255.255.252
Device(config-if)# ip nhrp map 192.168.24.4 2.2.2.2 //NHRP may optionally be configured to dynamically discover tunnel end points.
Device(config-if)# ip nhrp map multicast 2.2.2.2
```

```
Device(config-if)# ip nhrp network-id 1
Device(config-if)# ip nhrp nhs 192.168.24.4
Device(config-if)# ip pim sparse-dense mode
Device(config-if)# tunnel source Loopback0
Device(config-if)# tunnel destination 2.2.2.2
Device(config)# interface GigabitEthernet 0/0/0 //Receiver interface
Device(config-if)# ip address 10.2.2.2 255.255.255.0
Device(config-if)# ip pim sparse-dense-mode
```

配置服务发现网关

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置服务发现网关的前提条件

下面是配置服务发现网关的限制条件：

- 服务发现网关不支持多跳的拓扑。所有网段都要通过配置与之直连。服务发现网关可以从所有直连的网段那里学习服务，来建立自己的缓存并且充当代理来对请求消息作出响应；
- 该特性不支持使用第三方的 mDNS 服务器或应用；
- 在运行 mDNS 的 Cat4500sup8e MC（3.7）上，运行 INOS7.0 的 iphone 和 ipad 可能会在通过 mDNS 访问打印服务时出现问题。

关于服务发现网关与 mDNS 的信息

mDNS

mDNS 旨在实现零配置。根据零配置的定义，mDNS 应该提供下列特性：

- **编址**：为主机分配 IP 地址；
- **命名**：使用名称（而不是 IP 地址）来表示主机；
- **服务发现**：在网络中自动寻找服务

通过 mDNS，网络用户就不需要再分配 IP 地址、主机名或名称类型，才能访问网络中的服务了。用户只需要询问有哪些网络可以使用，然后从列表中选择即可。

通过 mDNS，编址可以通过使用 DHCP/DHCPv6 或 IPv4 和 IPv6 链路本地作用范围地址来实现。当网络中没有诸如 DHCP 或 DNS 这样的基础设施服务，也没有自分配的链路本地编制可以使用时，零配置的好处就显露出来了。客户端可以在链路本地范围（169.254.0.0/24）中选择一个随机的 IPv4 地址，或者使用 IPv6 链路本地地址（FE80::/10）进行通信。

通过 mDNS，命名（在本地网络中，使用 mDNS 实现域名到地址的转换）查询消息会使用链路本地作用范围的 IP 组播在本地网络中传输。由于这些 DNS 查询消息会发送给一个组播地址（IPv4 地址 224.0.0.251 或 IPv6 地址 FF02::FB），没有一台了解全局信息的 DNS 服务器需要响应查询消息。当服务或设备查看到了任何它了解的服务，它就会用 DNS 响应来发送自己缓存中的信息。

通过 mDNS，服务发现可以通过浏览的方式来实现。mDNS 查询消息会查询某个服务类型和域，了解相匹配服务的设备会用该服务的信息作出响应。用户可以从查询出来的可用服务列表中进行选择。

mDNS 协议（mDNS-RFC）可以与 DNS 服务发现（DNS-SD-RFC）一同使用，实现零配置的编址、命名与服务发现。

mDNS-SD

多 DNS 服务发现（mDNS-SD）会使用 DNS 协议的语义和组播，而不是知名组播地址，来实现零配置服务发现。DNS 数据包会使用组播地址 224.0.0.251 和对应的 IPv6 地址 FF02:FB，通过端口 5353 进行收发。

由于 mDNS 会使用链路本地组播地址，因此它的范围会限制在一个物理或逻辑局域网之内。如果网络需要扩展到一个分布式园区，或者扩展到一个包含多种不同网络技术的广域网环境中，用户就应该实施 mDNS 网络。mDNS 网关可以通过过滤、缓存和重分发服务来跨越三层边界，将 mDNS 数据包从一个三层域转发到另一个三层域中。

服务发现网关

服务发现网关特性会让组播域名系统（mDNS）跨越三层边界（即不同子网）进行操作。mDNS 网关可以通过过滤、缓存和重分发服务来跨越三层边界，将 mDNS 数据包从一个三层域（子网）转发到另一个三层域中。在实施这项特性之前，因为用户使用的是链路本地范围的组播地址，因此 mDNS 会被限制在一个子网的作用范围之内。这项特性可以带来 BYOD（自带设备）的提升。

mDNS 网关与子网

用户需要为实现服务发现来启用一个 mDNS 网关，以实现跨子网的操作。用户可以给设备或

者接口启用 mDNS 网关。

注释： 用户需要在接口级别进行配置之前，首先在全局配置服务路由。

在启用设备或接口，用户可以跨子网重分布服务发现信息。用户可以创建服务策略，并且针对入站服务发现信息（称为入站过滤）或入站服务发现信息（称为出站过滤）应用过滤策略。

注释： 如果在全局启用重分布，应该给全局配置设置比接口配置更高的优先级。

例如，在图中，如果用户在路由器上启用了 mDNS 网关功能，那么服务信息就会从一个子网发送到另一个子网。例如，以 IP 地址 192.0.2.6 为源地址在网络中进行通告的打印机和传真服务信息，会以 IP 地址 198.51.100.4 被重分发到网络中。网络中 IP 地址为 192.0.2.6 的打印机和传真服务信息，都是通过网络中那些启用了 mDNS 主机与设备学习过来的。

图 35：示例网络环境

Router	路由器
Network 192.0.2.6	网络 192.0.2.6
Network 198.51.100.4	网络 198.51.100.4

过滤

在配置了 mDNS 网关和子网之后，用户可以对自己希望重分布的服务执行过滤。在创建服务列表时，可以使用 **permit** 或 **deny** 可选项：

- 输入命令 **permit** 可以让用户允许或传输特定服务列表信息；
- 可选项 **deny** 可以让用户拒绝网络将一些可用的服务列表信息传输到其他子网当中。

在使用命令可选项 **permit** 或 **deny** 时，需要包含的序列号。相同的服务列表也可以关联多个序列号，而每个序列号都会映射一条规则。

注释： 如果用户没有配置过滤器，那么网络默认的操作是拒绝服务列表信息通过设备或者设备进行传输。

在创建服务烈表示，查询是一个可选项。用户可以使用一个服务列表来创建查询。如果用户希望浏览一项服务，那就可以使用主动查询。这项功能可以保证缓存中的记录更新。

注释： 主动查询只能全局使用，不能在接口级别实现。

当服务启动时，服务端点（如打印机或传真）会未经请求发送通告。在此之后，每当网络发生变更事件（譬如接口启动或者关闭），端点都会主动发送通告。设备永远会对查询作出响应。

在使用命令可选项 **permit** 或 **deny** 创建了服务列表之后，用户可以基于 *service-instance*（服务实例）、*service-type*（服务类型）或 *message-type*（消息类型）来使用匹配语句（命令）执行过滤。

如何配置服务发现网关

配置服务列表（CLI）

下面的流程描述了创建服务列表、针对该服务列表应用过滤器，以及给该服务列表名配置参数的方式。

总步骤

1. enable

2. configure terminal

3. service-list mdns-sd *service-list-name* {**deny** *sequence-number* | **permit** *sequence-number* | **query**}

4. match message-type {**announcement** | **any** | **query**}

5. match service-instance { *LINE* }

6. match service-type {*LINE* }

7. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**service-list mdns-sd** *service-list-name* {**deny** *sequence-number* | **permit** *sequence-number* | **query**} 示例： `Device(config)#` **service-list mdns-sd sl1 permit 3** `Device(config)#` **service-list mdns-sd sl4 query**	进入 mDNS 服务发现服务列表模式。在这种模式下，用户可以 • 创建一个服务列表，并且使用应用于序列号的 **permit** 或 **deny** 选项，来针对这个服务列表实施过滤器； • 创建一个服务列表，并且使用 query 可选项，来针对服务列表名称关联一个查询 **注释**：序列号会决定规则的优先级。优先级较低的规则会首先得到选择，服务通告或查询会根据用户的定义得到放行或拒绝。用户应根据网络需求来定义序列号
步骤 4	**match message-type** {**announcement** | **any** | **query**} 示例： `Device(config-mdns-sd-sl)#` **match message-type announcement**	（可选）设置要匹配的消息类型。用户可以匹配下列类型的消息： • announcement（通告消息） • any（两者皆匹配） • query（查询消息） 这些命令的作用是给步骤 2 中配置的服务列表名配置参数。 如果 **match message-type** 匹配的是通告，那么服务列表规则会允许服务通告发往设备。如果 **match message-type** 匹配的是查询，那么只有网络中某类服务的客户端所发送的查询消息才会被允许。 用户可以给不同序列号的多种服务创建同一个名称，过滤器的优先级是通过序列号指定的。服务列表就是一组有序的配置语句，每条语句都对应一个 permit 或 deny 操作。匹配服务列表的过程包含按照预先定义的顺序来扫描列表，还包括对每条语句的标准一一执行匹配。一旦设备找到第一条匹配的语句，扫描列表的操作就会停止，设备会根据匹配的语句执行对应的 permit/deny 操作。对列表执行扫描时，设备默认的操作是拒绝（deny）。

		注释：如果用户在前面的步骤中使用了 **query** 选项，那就不能使用 **match** 命令。这条命令只能至于 **permit** 或 **deny** 这两个可选项
步骤 5	**match service-instance** { *LINE* } 示例： `Device(config-mdns-sd-sl)##` **match service-instance servInst 1**	（可选）设置要匹配的服务实例 这条命令的作用是给步骤 2 中配置的服务列表名，设置参数。 **注释**：如果用户在前面的步骤中使用了 **query** 选项，那就不能使用 **match** 命令。这条命令只能至于 **permit** 或 **deny** 这两个可选项
步骤 6	**match service-type** {*LINE* } 示例： `Device(config-mdns-sd-sl)#` **match service-type _ipp._tcp**	（可选）设置要匹配的 mDNS 服务类型字符串值 这条命令的作用是给步骤 2 中配置的服务列表名配置参数。 **注释**：如果用户在前面的步骤中使用了 **query** 选项，那就不能使用 **match** 命令。这条命令只能至于 **permit** 或 **deny** 这两个可选项
步骤 7	**end** 示例： `Device(config-if)# end`	离开接口配置模式并返回特权 EXEC 模式

接下来做什么

继续启用 mDNS 网关并重分布服务。

启用 mDNS 网关并重分布服务（CLI）

在给设备启用了 mDNS 网关之后，用户可以使用 **service-policy** 和 **service-policy-query** 命令，来分别应用过滤器（应用入站过滤或出站过滤）和主动查询。用户可以使用命令 **redistribute mdns-sd** 来重分布服务和服务通告，并使用命令 **cache-memory-max** 将一部分系统内存设置为缓存。

注释： 在默认情况下，mDNS 网关在所有接口上都是禁用的。

总步骤

1. enable

2. configure terminal

3. service-routing mdns-sd

4. service-policy *service-policy-name* {**IN** | **OUT**}

5. redistribute mdns-sd

6. cache-memory-max *cache-config-percentage*

7. service-policy-query *service-list-query-name service-list-query-periodicity*

8. exit

~~**9. wireless multicast**~~

~~**10. no wireless mdns-bridging**~~

11. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**service-routing mdns-sd** 示例： `Device (config)#` **service-routing mdns-sd**	给设备启用 mDNS 网关功能，并进入组播 DNS 配置（config-mdns）模式。 **注释**：这条命令会在全局启用 mDNS 功能。 **注释**：在全局配置或接口配置模式下输入命令 **service-routing mdns-sd source-interface** *if-name*，来给出站 mDNS 数据包设置替代源接口，这样一来，如果没有在出站接口上配置，那么设备就会使用替代源接口的 IP 地址
步骤 4	**service-policy** *service-policy-name* {**IN** | **OUT**} 示例： `Device (config-mdns)#` **service-policy serv-pol1 IN**	（可选）针对服务列表，对入站服务发现信息（入站过滤）或出站服务发现信息（出站过滤）执行过滤
步骤 5	**redistribute mdns-sd** 示例： `Device (config-mdns)#` **redistribute mdns-sd**	（可选）跨子网重分发服务或服务通告 **注释**：若在全局启用重分发，那么全局配置的优先级应高于接口配置的优先级
步骤 6	**cache-memory-max** *cache-config-percentage* 示例： `Device (config-mdns)#` **cache-memory-max 20**	（可选）将一部分系统内存（以百分比的形式）设置为缓存。 **注释**：在默认情况下，百分之 10 的系统内存会保留给缓存使用。用户可以使用这条命令来覆盖默认的参数
步骤 7	**service-policy-query** *service-list-query-name service-list-query-periodicity* 示例： `Device (config-mdns)#` **service-policy-query sl-query1 100**	（可选）配置周期性的服务列表查询
步骤 8	**exit**	（可选）返回全局配置模式

	示例： `Device (config-mdns)#`**exit**	
步骤 9	~~**wireless multicast**~~ ~~示例：~~ ~~`Device (config)#` **wireless multicast**~~	~~（可选）启用无线以太网组播支持~~
步骤 10	~~**no wireless mdns-bridging**~~ ~~示例：~~ ~~`Device (config)#` **no wireless mdns-bridging**~~	~~（可选）禁用将 mDNS 数据包桥接到无线客户端~~
步骤 11	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

监控服务发现网关

表 52：监控服务发现网关

命令	目的
show mdns requests [**detail** | **name** *record-name*| **type** *record-type* [**name** *record-name*]]	这条命令会显示大量关于 mDNS 请求消息的信息，包括记录名和记录类型信息
show mdns cache [**interface** *type number* | **name** *record-name* [**type** *record-type*]| **type** *record-type*]	这条命令会显示 mDNS 缓存信息
show mdns statistics {**all** | **service-list** *list-name* | **service-policy** {**all** | **interface** *type number* }}	这条命令会显示 mDNS 统计数据

配置示例

示例：设置出站 mDNS 数据包的替代源接口

下面的示例显示了如何给出站 mDNS 数据包设置替代的源接口，以便在出站接口上没有配置时，可以使用替代源接口的 IP 地址。

```
Device(config)# service-routing mdns-sd
Device(config-mdns)# source-interface if-name
```

示例：启用 mDNS 网关与重分发服务

下面的示例显示了如何给一台设备启用 mDNS 网关，并且启用跨子网的重分发服务。用户对服务列表 serv-pol1 应用了入站过滤。用户将 20%的系统内存用于缓存，将服务列表查询周期配置为了 100 秒。

```
Device# configure terminal
Device# service-routing mdns-sd
Device(config-mdns)# service-policy serv-pol1 IN
Device(config-mdns)# redistribute mdns-sd
Device(config-mdns)# cache-memory-max 20
Device(config-mdns)# service-policy-query sl-query1 100
Device(config-mdns)# exit
```

示例：全局 mDNS 配置

下面的示例显示了如何在全局配置 mDNS。

```
Device# configure terminal
Device(config)# service-list mdns-sd mypermit-all permit 10
Device(config-mdns-sd-s1)# exit
Device(config)# service-list mdns-sd querier query
Device(config-mdns-sd-s1)# service-type _dns._udp
Device(config-mdns-sd-s1)# end
Device# configure terminal
Device(config)# service-routing mdns-sd
Device(config-mdns)# service-policy mypermit-all IN
Device(config-mdns)# service-policy mypermit-all OUT
```

示例：接口 mDNS 配置

下面的示例显示了如何给一个接口配置 mDNS。

```
Device(config)#interface Vlan136
Device(config-if)# description *** Mgmt VLAN ***
Device(config-if)# ip address 9.7.136.10 255.255.255.0
Device(config-if)# ip helper-address 9.1.0.100
Device(config-if)# service-routing mdns-sd
Device(config-if-mdns-sd)# service-policy mypermit-all IN
Device(config-if-mdns-sd)# service-policy mypermit-all OUT
Device(config-if-mdns-sd)# service-policy-query querier 60
```

配置完服务发现网关接下来做什么

用户还可以配置：

- IGMP
- 无线组播
- PIM
- SSM
- IP 组播路由

其他参考资料

相关文档

相关主题	文档名
配置 DNS	《IP 编址：DNS 配置指南，Inspur INOS XE 3SE 版》
DNS 概念信息	《IP 编址：DNS 配置指南，Inspur INOS XE 3SE 版》中的“关于 DNS 的信息（Information About DNS）”一节
平台独立配置信息	《IP 编址：DNS 配置指南，Inspur INOS XE 3SE 版》

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com

标准与 RFC

标准/RFC	标题
RFC 6763	基于 DNS 的服务发现
组播 DNS 互联网草案	组播

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

服务发现网关的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

IP 组播优化：在大型 IP 组播环境中优化 PIM 稀疏模式

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

在大型 IP 组播环境中优化 PIM 稀疏模式的前提条件

- 用户必须在网络中运行 PIM 稀疏模式；
- 如果用户准备使用组列表来控制对哪些组应用最短路径树（SPT）门限值，那就必须在执行这项配置之前配置访问列表；

在大型 IP 组播环境中优化 PIM 稀疏模式的前提条件

PIM 注册进程

IP 组播源不会使用信令机制来通告自己。源只会将它们的数据发送给相连的网络，而接收方则会使用互联网组管理协议（IGMP）来通告自己。如果源向配置在 PIM 稀疏模式（PIM-SM）

的组播组发送了流量，通往源的指定路由器（DR）一定回向汇集点（RP）通告源的存在。如果 RP 有下游接收方希望接收到从这个源发送过来的组播流量，但又还没有加入通往这个源的最短路径树，那么 DR 一定会从源向 RP 流量。PIM 注册进程会针对每个(S,G)条目进行运行，这个进程会完成 DR 和 RP 之间这些工作。

当 DR 创建了一个新的(S,G)状态，注册进程即告开始。此时 DR 会将所有与(S,G)状态相匹配的数据包封装到 PIM 注册消息，并且用单播的形式将这些注册消息发送给 RP。

如果 RP 有下游接收方希望接收新的源发来的注册消息，那么 RP 可以继续通过 DR 接收注册消息，也可以加入通往源的最短路径树。在默认情况下，RP 会加入最短路径树，这是因为组播流量的发送会提供最高吞吐量。在接收到第一个通过最短路径达到的数据包时，RP 会向 DR 反向发送一跳注册停止消息。当 DR 接收到这条注册停止消息时，它就会停止向 RP 发送注册消息。

如果 RP 没有下游接收方希望接收到新的源发送的注册消息，那么 RP 就不会加入最短路径。此时，RP 会立刻向 DR 发回一跳注册停止消息。当 DR 接收到这条注册停止消息时，它就会停止向 RP 发送注册消息。

一旦针对一个源建立了路由条目，DR 和 RP 之间就会周期性地重新进行注册。在组播路由表状态超时前 1 分钟，只要源还处于互动状态，DR 就会每秒向 RP 发送一条无数据注册消息，直到 DR 从 RP 那里接收到注册停止消息为止。这项操作会重启组播路由表条目的超时时间，往往会每 2 分钟进行一次重新注册消息交换。重新注册对于维护状态、恢复丢失的状态，以及追踪 RP 的源等操作而言是必不可少的。它会独立替代 RP 加入最短路径。

PIM 第 1 版的兼容性

如果 RP 运行的是 PIMv1，它就无法理解无数据注册消息。此时，DR 不会向 RP 发送无数据注册消息。DR 会在从 RP 那里接收到注册停止消息之后大约 3 分钟，DR 会将来自源的入站数据包封装到注册消息中，并将其发送给 RP。DR 会继续发送注册消息，直到 DR 再次从 RP 那里接收到一跳注册停止消息为止。如果 DR 运行的是 PIMv1，也会出现相同的情况。

当 DR 运行的 PIMv1 时，它会针对特定(S,G)条目将数据包封装到注册消息中，此时 DR 会对这个条目执行进程交换，而不是快速交换或者硬件交换。在一个支持这些较快路径的平台上，运行 PIMv1 的 RP 或 DR 执行的 PIM 注册进程可能会周期性导致数据包传输失序。有鉴于此，我们推荐用户将网络从 PIMv1 升级为 PIMv2。

PIM 指定路由器

配置了 IP 组播的设备会通过发送 PIM hello 消息来判断那些设备会成为各个局域网段（子网）中的指定路由器（DR）。这些 hello 消息中会包含设备的 IP 地址，而拥有最高 IP 地址的设备就会成为 DR。

DR 会向直连局域网中的所有主机发送互联网组管理协议（IGMP）主机查询消息。如果 DR 工作在稀疏模式下，那么它就会向汇集点（RP）发送源注册消息。

在默认情况下，组播设备会每 30 秒发送一次 PIM 路由器查询消息。用户如果让设备更加频繁地发送 PIM hello 消息，那么设备就可以更快发送邻居没有响应的情况。因此，设备也就可以更加有效地实施故障切换或恢复流程。这种操作很适合部署在那些网络边缘的冗余设备上。

PIM 稀疏模式注册消息

无数据注册消息是以每秒一条消息的速率进行发送的。如果 DR 正在注册突发源（即高数据速率的源）或者 RP 运行的不是 PIMv2，那么 DR 有可能会持续以高速率发送注册消息。
在默认情况下，设备在发送 PIM 稀疏模式注册消息时是没有速率限制的。限制注册消息的速率，也可以限制 DR 和 RP 的负载。但这样做的代价是，那些超出设置限制的注册消息会被设备丢弃。在突发源开始发送数据包的头一秒时间之内，接收方可能会经历丢包。

防止使用最短路径树来减少对内存的需求

理解 PIM 共享树和源树可以帮助用户理解为何防止使用最短路径树可以减少网络对设备内存的需求。

PIM 共享树和源树-最短路径树

在默认情况下，组播组成员会通过一棵根在 RP 的数据分发树，接收到由发送方发送给组的数据。下图显示了共享分发树的这种类型。发送方发送的数据被发送给了 RP，以便转发给加入了这棵共享树的组成员。

图 36：共享树与源树（最短路径树）

Source	源
Source tree (Shortest path tree)	源树 （最短路径树）
Shared tree from RP	包含 RP 的共享树
Receiver	接收方
Router A	路由器 A
Router B	路由器 B
Router C	路由器 C

如果数据速率可以保证，共享树中的叶路由器（即没有任何下游连接的路由器）可以使用根在源的分发树。这种分发树称为最短路径树或源树。在默认情况下，软件会在从源那里接收到第一个数据包时，就构建出一棵源树。
这个进程描述了从共享树一到到源树的迁移过程：
1 接收方加入了一个组；叶路由器 C 向 RP 发送了一条加入消息；
2 RP 将与路由器 C 的链路添加到自己的出站接口列表中；
3 源发送数据；路由器 A 将数据封装到注册消息中，并将其发送给 RP；
4 RP 沿着共享树向路由器 C 发送数据，同时向源发送了一条加入消息。此时，数据可能已经两次到达了路由器 C，一次是封装的，一次是未封装的；
5 当未封装数据到达 RP 时，它会向路由器 A 发送一条注册停止消息；
6 在默认情况下，接受第一个数据时，路由器 C 会向源发送一条计入消息；
7 当路由器 C 在(S,G)接收到了数据，它会沿着共享树发送一条修剪消息；
8 RP 从(S,G)的出站接口中删除了与路由器 C 之间的链路。RP 向源发送一条修剪消息。
加入与修剪消息是发送给源和 RP 的。这些消息都是逐跳发送的。在去往源后 RP 的路径中，每台 PIM 设备都会对这类消息进行处理。注册与注册停止消息不是逐跳发送的。这些消息是由与源直连的指定路由器发送的，最终会被这个组的 RP 接收到。
向组发送数据的组播源会使用共享树。

防止或延迟使用最短路径树的好处

当第一个数据包到达最后一跳路由器时，就会发生从共享树到源树的变更（防止或延迟使用最短路径树的好处，C）。之所以会发生这种变更，是因为用户使用全局配置命令 **ip pim spt-threshold** 来控制时间，这个参数的默认设置为 0kbps。

最短路径树需要的内存比源树多，但是可以减少延迟。用户可能希望推迟使用最短路径树，或者不使用最短路径树，以达到减少内存需求的目的。如果不想让叶路由器立刻切换到最短路径树，用户可以设置一个流量必须达到的门限值。

用户可以配置 PIM 叶路由器何时可以加入指定组的最短路径树。如果源发送的流量大于等于用户设置的流量速率，那么设备就会向源发送一条 PIM 加入消息，来构建源树（即最短路径树）。如果用户设置了关键字 **infinity**，那么这个特定组的所有源都会使用共享树，而永远不会切换到源树。

如何在大型 IP 组播环境中优化 PIM 稀疏模式

在大型网络中优化 PIM 稀疏模式

如果 IP 组播网络比较大，那么用户可以考虑执行下面的配置任务。

这项配置任务中的步骤 3、5、6 是相互独立的，因此这三步也都是可选的。这些步骤都有助于优化 PIM 稀疏模式。如果用户执行步骤 5 或步骤 6，那么用户必须执行步骤 4。步骤 6 只能应用于指定路由器；修改 PIM 查询时间间隔只适用于那些部署在 PIM 域边缘的冗余路由器。

总步骤

1. enable

2. configure terminal

3. ip pim register-rate-limit *rate*

4. ip pim spt-threshold {*kbps*| **infinity**}[**group-list** *access-list*]

5. interface *type number*

6. ip pim query-interval *period* [**msec**]

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip pim register-rate-limit** *rate* 示例： `Router(config)# ip pim register-rate-limit 10`	（可选）设置每秒针对每个(S,G)路由条目发送的 PIM 稀疏模式注册消息的最大数量限制。 • 使用这条命令可以限制路由器（DR）允许针对每个(S,G)条目发送的注册消息数量； • 在默认情况下，是没有最大速率设置的； • 限制注册消息的速率可以限制 DR 和 RP 的负

		载，但会导致超出设置限制的注册消息被丢弃； • 在突发源开始发送数据包的头一秒时间之内，接收方可能会经历丢包
步骤 4	**ip pim spt-threshold** {*kbps*| **infinity**}[**group-list** *access-list*] 示例： `Router(config)# ip pim spt-threshold infinity group-list 5`	（可选）设置在切换到最短路径树之前必须达到的门限值。 • 默认值为 0，默认操作是路由器一旦接收到第一个数据包就立刻加入 SPT； • 设置关键字 **infinity** 会让路由器永不切换到最短路径树；这台路由器会永远使用共享树。这个关键字应该应用于多对多的组播通信环境当中； • 组列表部分是控制要将 SPT 门限值应用于哪些组的标准访问列表。如果设置的值为 0，那么就不会使用组列表，门限值应用于所有组； • 在例中，用户配置了 group-list 5 来放行组播组 239.254.2.0 和 239.254.3.0： access-list 5 permit 239.254.2.0 0.0.0.255 access-list 5 permit 239.254.3.0 0.0.0.255
步骤 5	**interface** *interface-id* 示例： `Device(config)# interface gigabitethernet 1/0/1`	指定要配置的接口，并进入接口配置模式： • 如果不想修改 PIM STP 门限值或 PIM 查询时间间隔的默认值，那就不要这一步；这项配置任务已经结束
步骤 6	**ip pim query-interval** *period* [**msec**] 示例： `Router(config-if)# ip pim query-interval 1`	（可选）配置组播路由器发送 PIM 路由器查询消息的频率： • 只在 PIM 域边缘的冗余路由器执行这一步配置； • 默认查询时间间隔为 30 秒； • 参数 *period* 的设置单位为秒，除非用户设置了 **msec** 这个关键字

在大型 IP 组播环境中优化 PIM 稀疏模式的配置示例

在大型 IP 组播环境中优化 PIM 稀疏模式的示例

下面的示例显示了如何：

- 将快速收敛的查询时间间隔设置为 1；
- 将路由器配置为永不切换到 SPT 而继续使用共享树；
- 将设备每秒针对每个(S,G)路由条目发送的 PIM 稀疏模式注册消息数量限制为 10。

```
interface ethernet 0
ip pim query-interval 1
.
```

```
.
.
!
ip pim spt-threshold infinity
ip pim register-rate-limit 10
```

其他参考资料

相关文档

相关主题	文档名
Inspur INOS 命令	《Inspur INOS 主命令列表，所有版本》
Inspur INOS IP SLA 命令	《Inspur INOS IP 组播命令参考手册》
PIM 稀疏模式的概念与配置	“配置基本 IP 组播”部分或 “在 IPv6 网络中配置基本 IP 组播”部分

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

在大型 IP 组播环境中优化 PIM 稀疏模式的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

IP 组播优化：组播亚秒级收敛

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

组播亚秒级收敛的前提条件

服务提供商必须有一台启用了组播的核心，以便使用 Inspur 组播亚秒级收敛特性。

组播亚秒级收敛的限制条件

使用亚秒级指定路由器（DR）故障切换增强特性的设备，必须能够以毫秒为单位来处理到达的 hello 时间间隔信息。遭遇拥塞的设备或没有足够的 CPU 资源来处理 hello 时间间隔的设备，都可以默认协议独立组播（PIM）邻居已经断开连接，尽管事实有可能并非如此。

关于组播亚秒级收敛的信息

组播亚秒级收敛的好处

- 扩展性组件可以提升服务用户（接收方）和服务负载（源或内容）增加（或减少）时，处理的效率；
- 新的算法和处理（如聚合加入消息，可以用一个数据包发送最多 1000 条信息）可以将实现收敛的时间降低至原本时间的 1/10；
- 组播亚秒级收敛可以提升大型组播网络中的服务可用性；
- 组播用户（如金融服务企业和经济公司）可以获得更优的服务质量（QoS），因为网络的组播功能可以用远高于之前的效率得到恢复。

组播亚秒级收敛实现的扩展性增强

组播亚秒级收敛特性可以提升网络的扩展性，增强网络在服务用户（接收方）和服务负载（源或内容）增加（或减少）时，执行处理的效率。在这个版本中，扩展性增强包括：

- 通过新的计时器管理方法，提升了互联网组管理协议（IGMP）和 PIM 状态维护功能；
- 提升了组播源发现协议（MSDP）活动源（SA）缓存的扩展性。

扩展性增强可以提供下列好处：

- 提升潜在的 PIM 组播路由（mroute）、IGMP 和 MSDP SA 缓存状态的容量；
- 降低 CPU 使用率。

PIM 路由器查询消息

组播亚秒级收收敛可以让设备每几毫秒发送一次 PIM 路由器查询消息（PIM hello 消息）。设备会用 PIM hello 消息来定位邻居 PIM 设备。在介绍这项特性之前，设备可以每几秒发送一次 PIM hello 消息。用户可以让设备更加频繁地发送 hello 消息，这可以让设备更快发送邻居没有响应的情况。因此，设备也就可以更加有效地实施故障切换或恢复流程。

逆向路径转发

单播逆向路径转发（RPF）可以丢弃那些 IP 源地址不可查证的 IP 数据包，以此减轻因被人改造了 IP 源地址的数据包进入网络中而给网络造成的问题。这些被人改造了源地址的数据包出现在网络中，表示网络中出现了基于源 IP 地址欺骗的拒绝服务（DoS）攻击。
RPF 可以使用访问控制列表（ACL）来的判断如何处理伪造 IP 源地址的数据包。ACL 命令中有一条可选项可以让系统管理员将丢弃或转发的数据包记录下来。这些与伪造数据包相关的日志记录信息可以帮助用户揭露出一些与网络攻击有关的信息。
基于接口的统计数据可以帮助系统管理员迅速发现网络中作为攻击入口点的那个接口。

RPF 校验

PIM 的目标是使用标准单播路由表来转发 IP 组播流量。PIM 会使用单播路由表来判断 IP 组播数据包的源是否是通过这个源的最优路径到达设备的。RPF 校验这项进程是协议独立的，因为它是基于单播路由表来实现的，并不是依赖任何特定的路由协议。

触发的 RPF 校验

组播亚秒级收敛提供了一种针对组播状态触发 RPF 变更校验的机制。这种校验是由单播路由变化所触发的。通过触发的 RPF 校验，用户可以将周期性的 RPF 校验设置为一个相对比较高的值（比如 10 秒），同时仍然能够做到快速切换。
触发的 RPF 校验可以减少网络从中断中恢复过来所需的时间，无论是单一服务事件（例如与某一个源和一个接收方有关的情况），还是与大量参数有关的服务（例如，设计大量源、大量接收方和大量接口）。这种增强特性可以减少 PIM（组播路由）、IGMP 和 MSDP（SA 缓存）状态的收敛时间。

RPF 故障切换

在一个使用触发的 RPF 校验，且又不稳定的组播路由环境中，环境可能会频繁触发 RPF 校验，这会给设备资源带来负担。为了避免这种问题，用户可以使用命令 **ip multicast rpf backoff** 防止设备在一段指定的时间长度内二次触发的 RPF 校验。也就是说，PIM 会从另一次触发的

RPF 校验中“回退”一段用户配置的最小毫秒数。
如果回退周期超时，而路由表又没有进一步发生变化，PIM 就会扫描路由变化，并且根据变化建立组播 RPF 变更。但是，如果在回退期间出现了更多路由变更，那么 PIM 就会将回退周期增倍，以避免路由表仍然收敛期间，因 PIM RPF 变更致使设备过载。

拓扑变更与组播路由恢复

组播亚秒级收敛特性集可以让网络在完成了单播路由恢复后，组播路径能够几乎同时恢复，这样会增强企业和服务运营商网络骨干。
由于当网络拓扑变化时，PIM 会依靠单播路由表来计算自己的 RPF，因此单播协议首先需要计算出流量最佳路径的可选项，然后组播就可以判断出最佳路径了。
组播亚秒级收敛可以让组播协议在单播计算完成之后，几乎立刻完成计算。因此，在拓扑出现变化之后，组播流量转发可以更快恢复。

如何配置组播亚秒级收敛

修改周期 RPF 校验时间间隔

用户可以根据需要，修改周期性 RPF 校验发生的时间间隔。
注释： Inspur 建议用户*不*要修改命令 **ip rpf interval** 的默认值。默认值可以实现亚秒级的 RPF 故障切换。周期 RPF 校验发生的默认时间间隔为 10 秒。
总步骤
1. enable
2. configure terminal
3. ip multicast rpf interval *seconds* [**list** *access-list* | **route-map** *route-map*]
具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip multicast rpf interval** *seconds* [**list** *access-list*	**route-map** *route-map*] 示例： `Device(config)# ip multicast rpf interval 10`	配置周期性 RPF 校验发生的时间间隔，单位为秒

配置 PIM RPF 故障切换的时间间隔

用户可以根据需要，配置因路由表中的变化而触发 PIM RPF 故障切换的时间间隔。

注释： Inspur 建议用户*不*要修改命令 **ip multicast rpf backoff** 的默认值。默认值可以实现亚秒级的 RPF 故障切换。

总步骤

1. enable

2. configure terminal

3. ip multicast rpf backoff *minimum maximum* [**disable**]

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip multicast rpf backoff** *minimum maximum* [**disable**] 示例： `Device(config)# ip multicast rpf backoff 100 2500`	配置最小和最大的回退时间间隔

修改 PIM 路由器查询消息时间间隔

用户可以根据需要修改 PIM 路由器查询消息时间间隔。

总步骤

1. enable

2. configure terminal

3. interface *type slot* / *subslot* / *port*

4. ip pim query-interval *period* [**msec**]

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device#` **configure terminal**	
步骤 3	**interface** *type slot / subslot / port* 示例： `Device(config)# interface gigabitethernet 1/0/0`	选择接口并进入接口配置模式
步骤 4	**ip pim query-interval** *period* [**msec**] 示例： `Device(config-if)# ip pim query-interval 45`	配置组播路由器发送 PIM 路由器查询消息的频率

查看组播亚秒级收敛的配置

用户可以通过下面的步骤来查看和验证关于组播亚秒级收敛特性的具体信息。

总步骤

1. enable

2. show ip pim interface *type number*

3. show ip pim neighbor

具体步骤

步骤1 enable

示例：

```
Device> enable
```

进入特权 EXEC 模式。在提示时输入密码。

步骤 2 show ip pim interface *type number*

用户可以使用这条命令来查看关于配置了 PIM 的接口的信息。

下面是命令 **show ip pim interface** 的输出信息示例。

示例：

```
Device# show ip pim interface GigabitEthernet 1/0/0
Address Interface Ver/ Nbr Query DR DR
Mode Count Intvl Prior
172.16.1.4 GigabitEthernet1/0/0 v2/S 1 100 ms 1 172.16.1.4
```

步骤 3 show ip pim neighbor

用户可以使用这条命令来查看 Inspur INOS XE 软件发现的 PIM 邻居。下面是命令 **show ip pim neighbor** 的输出信息示例。

示例：

```
Device# show ip pim neighbor
PIM Neighbor Table
Neighbor Interface Uptime/Expires Ver DR
Address Prio/Mode
172.16.1.3 GigabitEthernet1/0/0 00:03:41/250 msec v2 1 / S
```

组播亚秒级收敛的配置示例

修改周期性 RPF 校验时间间隔的示例

在下例中，用户将命令 **ip multicast rpf interval** 的参数设置为了 10 秒。这条命令并不会显示在命令 **show running-config** 的输出信息中，除非用户将时间间隔值配置为了一个非默认的值。

```
!
ip multicast-routing
ip multicast rpf interval 10
.
.
.
interface Ethernet0/0
ip address 172.16.2.1 255.255.255.0
.
.
.
ip pim sparse-mode
!
```

PIM RPF 故障切换的时间间隔

在下例中，用户使用命令 **ip multicast rpf backoff** 将最小回退时间间隔值设置为了 100，将最大回退时间间隔值设置为了 2500。这条命令并不会显示在命令 **show running-config** 的输出信息中，除非用户将时间间隔值配置为了一个非默认的值。

```
!
ip multicast-routing
.
.
.
ip multicast rpf backoff 100 2500
!
!
interface Ethernet0/0
```

```
ip address 172.16.2.1 255.255.255.0
.
.
.
ip pim sparse-mode
!
```

修改 PIM 路由器查询消息时间间隔的示例

在下例中，用户将命令 **ip pim query-interval** 的参数设置为了 100 毫秒。这条命令并不会显示在命令 **show running-config** 的输出信息中，除非用户将时间间隔值配置为了一个非默认的值。

```
!
interface gigabitethernet0/0/1
ip address 172.16.2.1 255.255.255.0
ip pim query-interval 100 msec
ip pim sparse-mode
```

其他参考资料

相关文档

相关主题	文档名
Inspur INOS 命令	《Inspur INOS 主命令列表，所有版本》
Inspur INOS IP SLA 命令	《Inspur INOS IP 组播命令参考手册》
PIM 稀疏模式的概念与配置	“配置基本 IP 组播”部分或 “在 IPv6 网络中配置基本 IP 组播”部分

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

组播亚秒级收敛的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

IP 组播优化：跨越等价路径实现 IP 组播负载分割

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

跨越等价路径实现 IP 组播负载分割的前提条件

用户要按照《IP 组播：PIM 配置指南》文档中，“配置基本组播”这部分内容的描述，在设备上启用 IP 组播。

关于跨越等价路径实现 IP 组播负载分割的信息

负载分割与负载分担

负载分割和负载分担并不相同。负载分割提供了一种跨越多条等价逆向路径转发（RPF）路径，来随机分布(*,G)和(S,G)数据的方式，这未必会让 IP 组播流量负载通过这些等代价路径平衡地进行发送。通过随机分发(*,G)和(S,G)流量，这种分割 IP 组播流量负载的方式会尝试

在不对数据流进行计数的情况下，通过作出为随机转发决策的方式，在每条可用的 RPF 路径上分发等价的流量。这些方式称为等价多路径（ECMP）组播负载分割，如果网络中有许多占用带宽大致相等的流量，那么这种方式可以更好地实现负载共享。
如果只有几种(S,G)或(*,G)状态需要通过一系列等价链路进行转发，那么它们能够很好地实现负载分割的几率就不会很高。要克服这种限制，可以预先给(S,G)状态计算源地址或者给(*,G)状态计算汇集点（RP），以此更加合理地实现负载分享。这种限制会在 Inspur 快速转发（CEF）中，或者通过 EtherChannels 平等地应用于以数据流为单位的负载分割：在只有几条数据流的情况下，这些负载分割的方式无法在没有工程人员手动干预的情况下达到良好的流量分发效果。

在存在多等价路径时 IP 组播的默认操作

在默认情况下，如果网络中有多条等价路径，那么对于 PIM-SM（协议独立组播稀疏模式）、PIM-SSM（特定源组播）、双向 PIM 和 PIM 密集模式（PIM-DM）组来说，针对 IPv4 组播流量执行的逆向路径转发（RPF）就会基于拥有最高 IP 地址的 PIM 邻居来执行操作。这种方式称为最高 PIM 邻居操作。这种操作方式符合 PIM-SM 的 RFC 2362 标准，但也同样适用于 PIM-SSM、PIM-DM 和双向 PIM。
下图显示了本节的示例拓扑，这张拓扑的目的是解释当网络中存在多等价路径时，IP 组播的默认操作。
注释： 虽然下图和下面的示例在配置中使用得到都是路由器，但用户可以使用任意设备（路由器和交换机）。
图 37：在存在多等价路径时 IP 组播的默认操作

Source 1	源 1
Source 2	源 2
Device 1	设备 1
Device 2	设备 2
Receiver	接收方

在途中，两个源 S1 和 S2 正在向 IPv4 组播组 G1 和 G2 发送流量。在这个拓扑中，既可以使用 PIM-SM，也可以使用 PIM-SSM 或 PIM-DM。如果使用的是 PIM-SM，那么我们假设用户在设备 2 上针对命令 **ip pim spt-threshold** 保留了默认参数 0，该设备运行了内部网关协议（IGP），并且命令 **show ip route**（如果在设备 2 上输入）对于 S1 和 S2 的输出信息会将设备 1 上的串行接口 0 和串行接口 1 显示为设备 2 的等价下一跳 PIM 邻居。
如果不作进一步的配置，那么拓扑中的 IPv4 组播流量就会穿越其中的一个串行接口（串行接口 0 或串行接口 1），具体使用哪个接口取决于哪个接口的 IP 地址更高。例如，若在设备 1 上，串行接口 0 和串行接口 1 上配置的 IP 地址分别为 10.1.1.1 和 10.1.2.1。在这种环境中，如果使用 PIM-SM 和 PIM-SSM，那么设备 2 永远会向 10.1.2.1 发送 PIM 加入消息，同时也永远会通过串行接口 1 接收图中所有源和组的 IPv4 组播流量。如果使用 PIM-DM，那么设备 2 会永远通过串行接口 1 来接收 IP 组播流量，唯有在这种情况下，PIM-DM 环境中才不会使用 PIM 加入消息。此时，设备 2 会修剪掉通过串行接口 0 接收到 IP 组播流量，并且通过串行接口 1 接收到这些流量。这是因为在设备 1 上，串行接口 1 的 IP 地址更高。
中间组播设备会执行 IPv4 RPF 查找来判断 IPv4 (*,G)和(S,G)组播路由（树）中的 RPF 接口和 RPF 邻居。RPF 查询由 RPF 路由选择和路由路径选择组成。RPF 路由选择只针对 IP 单播地址，以判断组播树的根。对于(*,G)路由（PIM-SM 和双向 PIM），组播树的根就是去往组 G 的 RP

地址；对于(S,G)树（PIM-SM、PIM-SSM 和 PIM-DM），组播树的根是源 S。RPF 路由选择会在路由信息库（RIB）中发现去往 RP 或源的最佳路由，和（如配置）距离矢量组播路由协议（DVMRP）路由表、多协议边界网关协议（MBGP）路由表或者用户配置的静态组播路由。如果得到的路由只有一跳可用路径，那么 RPF 查找即告完成，下一跳设备和路由的接口就会成为这棵组播树的 RPF 邻居和 RPF 接口。如果路由有多条可用路径，那么设备就会使用路由路径选择来判断选择哪条路径。

对于 IP 组播，设备会使用下面的路由路径选择方法：

注释： 在 IP 组播中，除了路由路径选择的默认方法之外，所有方法都可以启用某种形式的 ECMP 组播负载分割。

- 最高 PIM 邻居：这是默认的方法；因此，不需要执行任何配置。如果存在多等价路径，那么针对 IPv4 组播流量执行的逆向路径转发（RPF）就会基于拥有最高 IP 地址的 PIM 邻居来执行操作。因此，如果不进行配置，ECMP 组播负载分割默认是禁用的；
- 基于源地址的 ECMP 组播负载分割：用户可以使用命令 **ip multicast multipath** 来配置 ECMP 组播负载分割。输入这条 **ip multicast multipath** 命令可以使用 S 散列算法，启用基于源地址的 ECMP 组播负载分割。要想了解更多信息，可以参考使用 S 散列算法实现基于源地址的 ECMP 组播负载分割；
- 基于源和组地址的 ECMP 组播负载分割：用户可以在配置命令 **ip multicast multipath** 时，包含关键字 **s-g-hash** 和 **basic**，来配置 ECMP 组播负载分割。输入这条 **ip multicast multipath** 命令可以使用基本的 S-G 散列算法，启用基于源和组地址的 ECMP 组播负载分割。要想了解更多信息，可以参考使用基本 S-G 散列算法实现基于源和组地址的 ECMP 组播负载分割；
- 基于源、组和下一跳地址的 ECMP 组播负载分割：用户可以在配置命令 **ip multicast multipath** 时，包含关键字 **s-g-hash** 和 **next-hop-based**，来配置 ECMP 组播负载分割。输入这条 **ip multicast multipath** 命令可以使用基于下一跳的 S-G 散列算法，启用基于源、组和下一跳地址的 ECMP 组播负载分割。要想了解更多信息，可以参考基于源、组和下一跳地址的 ECMP 组播负载分割；

默认操作（即最高 PIM 邻居操作）不会让 IP 组播享受任何形式的 ECMP 负载分割，只会从可用路径的各个下一跳 PIM 邻居中选出拥有最高 IP 地址的 PIM 邻居。在使用命令 **show ip pim neighbor** 进行查看时，下一跳就会被设备认为是 PIM 邻居，如果设备从下一跳那里接收到 PIM hello 消息并且该消息并为过时，那么这个下一跳也确实就是 PIM 邻居。如果没有可用的下一跳是 PIM 邻居，那么设备就会选择拥有最高 IP 地址的下一跳。

负载分割 IP 组播流量的方法

总地来说，分割 IP 组播流量可以采用下面的方法：

- 用户可以基于源地址、基于源和组地址，或者基于源、组和下一跳地址来启用 ECMP 组播负载分割。在设备识别出等价路径之后，ECMP 组播负载分割会以(S,G)为单位执行分配，而不是像单播流量那样以数据包为单位执行负载分割；
- 对 IP 组播执行负载分割的另一种方式是将两条或多条等价路径统一为一条通用路由封装（GRE）隧道，并且允许单播路由协议执行负载分割，或者通过接口束（如快速或吉比特 EtherChannel 接口、多链路 PPP[MLPPP]链路束或多链路帧中继[FR.16]链路束）来分割负载；

ECMP 组播负载分割概述

在默认情况下，对 IPv4 组播流量执行 ECMP 组播负载分割是禁用的。ECMP 组播负载分割可以使用命令 **ip multicast multipath** 来进行启用。

使用 S 散列算法实现基于源地址 ECMP 组播负载分割

基于源地址的 ECMP 组播流量负载分割会使用 S 散列算法，这可以让设备为每个(*,G)或(S,G)状态从可用等价路径中选择 RPF 接口，具体选择的接口取决于状态解析出来的 RPF 地址。对于一条(S,G)条目，RPF 地址就是状态的源地址。而对于一条(*,G)条目，RPF 地址就是与状态组地址相关的 RP 的地址。

如果用户配置了基于源地址的 ECMP 组播负载分割，那么设备就可以通过多个等价接口接收到不同状态的组播流量。这种 IPv4 组播流量分割的方式与 IPv4 CEF 中默认的，基于数据流的负载分割，和通过快速和吉比特 EtherChannel 实现负载分割的概念十分相似。不过，这种 ECMP 组播负载分割会受到极化的影响。

使用基本 S-G 散列算法实现基于源和组地址的 ECMP 组播负载分割

基于源和组地址 ECMP 组播负载分割会使用简单散列算法，也称为基本 S-G 散列算法，这种算法是基于源和组地址进行运算的。基本 S-G 散列算法的计算结果是可以预测的，因为在散列值中不会使用随机值。但 S-G 散列机制会受到极化的影响，因为对于一个给定的源和组，无论由哪台设备执行运算，计算出来的散列值都是相同的。

注释： 基本的 S-G 散列算法会忽略双向 PIM 组。

使用 S 散列算法与基本 S-G 散列算法的可预测性

当拓扑中多处都拥有相同数量的等价路径时，IPv4 组播中的 ECMP 组播负载分割所使用的方法就可以给网络持续提供负载分割。如果通过计算一次 RP 地址或源地址来让流量通过 N 条路径进行分流，那么这些地址在拓扑各处也都会以相同的方式分割到多条路径中。持续负载分割是可预测的，而可预测这一特点又让管理员能够手动干预 IPv4 组播流量的负载分割。

使用 S 散列算法与基本 S-G 散列算法的极化

在 IPv4 组播中，用来基于源地址，或基于源和组地址对组播流量执行负载分割的散列机制容易收到一个问题的影响，这个问题叫作极化。极化是基于源或基于源和组地址执行 ECMP 组播负载分割的一个副产品，它可以让一些拓扑中的路由器无法有效利用所有可用路径来实现负载分割。

下图显示了本节中要使用的示例拓扑，这张图旨在解释基于源或基于源和组地址执行 ECMP 组播负载分割的极化问题。

注释： 虽然下图和下面的示例在配置中使用得到都是路由器，但用户可以使用任意设备（路由器和交换机）。

图 38：极化拓扑

Source 1	源 1
Source 2	源 2
Device 1	设备 1
Device 2	设备 2
Device 3	设备 3
Device 4	设备 4
Device 5	设备 5
Device 6	设备 6
Device 7	设备 7

Receiver	接收方

在上图所示的拓扑中，我们可以看到路由器 7 有两条通往源 S1 到 S10 的等价路径，一条通过路由器 5，一条通过路由器 6。在这个拓扑中，我们假设用户在拓扑中的所有路由器上使用命令 **ip multicast multipath** 启用了 ECMP 组播负载分割。在这个环境中，路由器 7 原本应该对 10 条(S,G)条目应用等价负载分割。但这个环境中的极化问题会对路由器 7 造成影响，因为这台路由器最终可能会选择路由器 5 上的串行接口 0 作为与源 S1 到 S5 通信的路径，而选择路由器 6 上串行接口 S1 作为与源 S6 到 S10 通信的路径。不仅如此，这种极化的问题也有可能影响拓扑中的路由器 5 和路由器 6。路由器 5 有两条去往 S1 到 S5 的等价路径，一条是通过路由器 1 的串行接口 0，一条是通过路由器 2 的串行接口 1。由于路由器 5 会应用相同的散列算法来选择使用这两条路径中的哪条路径，因此去往源 S1 到 S5 最终都会选择这两条路径中的某一条。这也就是说，要么所有流量都通过路由器 1 和路由器 5，要么所有流量都通过路由器 2 和路由器 5。在这个环境中，使用路由器 1 和路由器 5，同时使用路由器 2 和路由器 5 来实现负载分割是不可能的。同理，极化问题也适用于路由器 3 和路由器 6，以及路由器 4 和路由器 6。换言之，在这个环境中，使用路由器 3 和路由器 6，同时使用路由器 4 和路由器 6 来实现负载分割也是不可能的。

基于源组和下一跳地址的 ECMP 组播负载分割

基于源、组和下一跳地址来配置 ECMP 组播负载分割可以实现一种更加复杂的散列算法，也就是基于下一跳的 S-G 散列算法，这种算法是基于源、组和下一跳地址的算法。基于下一跳的 S-G 散列算法是可以预测的，因为在计算散列值的过程中，没有任何随机数的参与。基于下一跳的 S-G 散列算法所采用的散列机制与 S-散列算法和基本的 S-G 散列算法不同，这种机制不会受到极化问题的影响。

注释： 在 IPv4 组播中，基于下一跳的 S-G 散列算法与 IPv6 ECMP 组播负载分割中使用的算法相同，后者使用了与 PIM-SM 自举设备（BSR）相同的散列功能。

基于下一跳的散列机制并不会造成极化问题，同时也可以在路径出现故障时维系 RPF 的稳定性。不过这些优势也是有代价的。代价在于工程师无法使用源或 RP 的 IP 地址进行可靠预测，并且根据需要规划出使用基于下一跳的 S-G 散列算法的那种负载分割效果。鉴于许多客户网络都实施了等价多路径拓扑，因此手动对负载分割进行配置有时并不必需。网络更需要的，是 IP 组播的默认行为必须与 IP 单播相似；也就是说，IP 组播会通过尽力而为的方式使用多等价开销路径。因此，由于担心极化问题，对 IPv4 组播执行负载分割是不能默认启用的。

注释： 对 CEF 单播执行的负载分割采用的也是一种不会出现极化问题的方式，因此也同样不能用来预测负载分割的结果，也不能干涉负载分割的做法。

基于下一跳的散列功能可以避免极化问题，因为它将 PIM 邻居的实际下一跳地址引入到了计算当中，因此散列的结果因设备而异，于是也就不会出现极化的问题了。除了避免极化问题之外，这种散列机制也增加了路径故障时，设备所选 RPF 路径的稳定性。如果一台设备有 4 条等价路径，同时有大量状态通过这些路径进行负载分割。当其中一条路径出现故障时，网络中就还剩下了 3 条可用路径。此时如果使用那些极化散列算法（也就是 S 散列和基本 S-G 散列算法使用的机制）所采用的机制，那么所有状态的 RPF 路径可能会重新收敛，并发生相应的变化，尤其是那些之前使用了这三条路径的 RPF 路径会发生变化。这有可能会导致这些状态不必修改自己的 RPF 接口和下一跳连接。存在这种问题的原因在于，算法在选择路径时，会考虑所有可用路径的总数。这样一来，一旦路径发生变化，设备给所有状态选择的 RPF 也就会发生变化。对于基于下一跳的散列算法来说，只有那些使用了修改过的路径作为 RPF 的状态，需要重新收敛到三条剩余路径之一当中。至于那些本来就没有使用变更路径的状态则不会发生变化。如果第 4 条路径恢复，那么之前使用这条路径的状态会立刻重新收敛，并

且使用这条路径，这个过程并不会对其他状态造成影响。

注释： 基于下一跳的 S-G 散列算法会忽略双向 PIM 组。

针对 PIM 邻居查询和 RPF 路径选择的 Hello 消息执行 ECMP 组播负载分割的效果

如果不启用通过 ECMP 对 IP 组播流量执行负载分割的操作，且网络中有多条通向一个 RP 或源的等价路径，那么 IPv4 组播会首先选择 IP 地址最高的 PIM 邻居。PIM 邻居是发送 PIM hello 消息（或 PIMv1 查询消息）的设备。例如，有一台设备有两条通过 IGP 学习到的等价路径或者通过两条静态路由配置的等价路径。这两条路径的下一跳为 10.1.1.1 和 10.1.2.1。如果这两台下一跳设备发送 PIM hello 消息，那么 10.1.2.1 就应该成为最高 IP 地址的 PIM 邻居。如果只有 10.1.1.1 发送 PIM hello 消息，那么选中的就应该是 10.1.1.1。如果它们都不发送 PIM hello 消息，那么选中的就应该是 10.1.2.1。这种对 PIM hello 消息的尊重，让网络得以在只有静态组播路由的环境中建立某种动态故障切换的环境。否则，这种操作方式并不非常实用。

注释： 要想了解更多关于配置静态路由的信息，可以在 Inspur INOS IP 组播 FTP 站点参阅 Inspur INOS 中配置多静态组播路由的配置注释，获取这份文件可以访问 ftp://ftpeng.icntnetworks.com/ipmulticast /config-notes/static-mroutes.txt：。

在启用通过 ECMP 对 IP 组播流量执行负载分割的操作之后，设备不会考虑邻居发送的 PIM hello 消息。这也就是说，设备在选择 RPF 邻居时并不会考虑是否从该邻居那里接收到了 PIM hello 消息，这只取决于网络中是否存在等价路由条目。

在 PIM-DM 和双向 PIM 的 DF 选举中，对断言处理执行 ECMP 组播负载分割的效果

用户使用命令 **ip multicast multipath** 只会修改下游设备的 RPF 选择结果。这条命令对于双向 PIM 中的指定转发器选举，或者 PIM-DM 环境中上游设备的断言处理不会产生效果。

下图显示了本节中要使用的示例拓扑，这张图旨在解释 ECMP 组播负载分割对 PIM-DM 环境中的断言处理以及双向 PIM 环境中的 DF 选举所产生的效果。

注释： 虽然下图和下面的示例在配置中使用得到都是路由器，但用户可以使用任意设备（路由器和交换机）。

图 39：对 PIM-DM 环境中的断言处理以及双向 PIM 环境中的 DF 选举执行 ECMP 组播负载分割

Source 1	源 1
Source 2	源 2
Device 1	设备 1
Device 2	设备 2
Device 3	设备 3
Device 4	设备 4
Receiver	接收方

在图中，设备 2 有两条去往 S1 和 S2 的等价路径，也有设备 1 上的 RP 地址。这些路径都要通过吉比特以太网接口 1/0/0 进行发送：一条将流量发往设备 3，一条将流量发往设备 4。对于 PIM-SM 和 PIM-SSM 环境中的(*,G)和(S,G)RPF 选择，拓扑中设备 2 的行为是没有区别的。但使用 PIM-DM 和使用双向 PIM 是有区别的。

如果图中所示拓扑使用的是 PIM-DM，设备 3 和设备 4 就会开始通过吉比特以太网接口 1/0/0 为这些状态泛洪流量，并且使用 PIM 断言处理来从它们当中选举一台设备转发流量，以避免重复发送流量。由于设备 3 和设备 4 都拥有相同的路由开销，因此吉比特以太网接口 1/0/0 的 IP 地址更高的设备永远会赢得断言处理。所以，如果在这个拓扑中使用 PIM-DM，那么流量就不能通过设备 3 和设备 4 实现负载分割。

如果图中所示拓扑使用的是双向 PIM，那么设备 2、设备 3 和设备 4 之间就会在吉比特以太

网接口 1/0/0 上执行一个称为 DF 选举的进程。DF 选举的进程会针对每个 RP 选举出一台设备来通过吉比特以太网接口 1/0/0 使用这个 RP 向各个组发送流量，选举的原则是判断哪个接口配置的 IP 地址最高。即使使用了多个 RP（例如 G1 一个 RP，G2 一个 RP），那么针对这些 RP 执行的 DF 选举，结果也永远是吉比特以太网接口 1/0/0 中拥有最高 IP 地址的设备获胜（设备 3 或者设备 4）。用于 DF 选举的选举规则实际上与用于 PIM 断言进程的选择规则相同，只不过 DF 选举中协商信息的协议机制经过了改良（以便返回更加准确的结果）。因此，如果拓扑中使用的是双向 PIM，设备就会跨越吉比特以太网接口 1/0/0 实现负载分割。

ECMP 组播负载分割的确会影响 RPF 选择，但不会影响 PIM-DM 中的断言处理，也不会影响双向 PIM 中的 DF 选举，其中的原因在于断言处理和 DF 选举是一个相互协作的进程，参与设备上必须实施相同的操作。对实施的操作进行修改可能会让协议操作产生某些形式的变化，因此参与设备必须达成一致。RPF 选择是一项纯设备本地的策略，因此可以在各台设备上本地禁用和启用，并不会给协议操作带来任何变化。

对于 PIM-DM 和双向 PIM 来说，只有在等价路径不是同一个局域网中的上游 PIM 邻居，而是不同局域网或点到点链路邻居时，使用命令 **ip multicast multipath** 配置 ECMP 组播负载分割才是有效率的。

在 PIM-SM 和 PIM-SSM 中，对断言处理执行 ECMP 组播负载分割的效果

有时候，由于 PIM 断言进程替代了 RPF 选举的效果，那么即使使用 PIM-SM 来对(*,G)或(S,G)条目执行流量转发，或者使用 PIM-SSM 对(S,G)条目执行流量转发，使用命令 **ip multicast multipath** 配置 ECMP 负载分割都无法产生效果。

下图显示了本节中要使用的示例拓扑，这张图旨在解释在 PIM-SM 和 PIM-SSM 中，对断言处理执行 ECMP 组播负载分割的效果。

注释： 虽然下图和下面的示例在配置中使用得到都是路由器，但用户可以使用任意设备（路由器和交换机）。

图 40：在 PIM-SM 和 PIM-SSM 环境中对 PIM 断言处理执行 ECMP 组播负载分割

Source 1	源 1
Source 2	源 2
Device 1	设备 1
Device 2	设备 2
Device 3	设备 3
Device 4	设备 4
Device 5	设备 5
Receiver 1	接收方 1
Receiver 2	接收方 2

在上图所示拓扑中，如果设备 2 和设备 5 都是 Inspur 设备，而且都通过命令 **ip multicast multipath** 配置了 ECMP 组播负载分割，那么负载分割可以如期在网络中生效。这也就是说，两台设备都会以设备 3 和设备 4 作为等价下一跳设备，而且会按照相同的方式对等价路径进行排序（根据 IP 地址）。在应用多路径散列函数时，对于每条(S,G)或(*,G)状态，设备都会选择相同的 RPF 邻居（设备 3 或设备 4），将它们的 PIM 加入消息发送给邻居。

如果设备 5 和设备 2 上配置的命令 **ip multicast multipath** 有别，或者如果设备 5 是一台第三方设备，那么设备 2 和设备 5 有可能会对一些(*,G)或(S,G)状态选择不同的 RPF 邻居。例如，设备 2 可能会选择设备 3 作为某个(S,G)条目的 RPF 邻居，或者设备 5 选择设备 4 作为某条(S,G)条目的 RPF 邻居。在这种环境中，设备 3 和设备 4 都会开始为这个条目将流量转发给吉比特以太网接口 1/0/0，于是它们也就看到了对方转发的流量，为了避免流量重复，它们就会启动断言进程。于是，对于这个(S,G)条目，吉比特以太网接口 1/0/0 配置的 IP 地址比较高

的哪台设备就会转发流量。然而，设备 2 和设备 5 都会追踪断言选举的胜方，并且将它们针对该条目的 PIM 加入消息发送给断言选举的获胜方，即使断言选举的获胜方并不是它们在 RPF 选举中计算出来的那台设备。

因此，对于 PIM-SM 和 PIM-SSM 来说，只有同一个局域网中的下游设备采用了一致的配置，并且都是 Inspur 设备时，ECMP 组播负载分担的操作才能得到保障。

单播路由变更时的 ECMP 组播负载分割与重新收敛

当单播路由变更时，所有 IP 组播路由状态都会立刻基于单播路由表进行重新收敛。具体来说，如果一条路径断开，剩下的路径就会立刻重新收敛，如果这条路径恢复，那么组播转发也会继而重新收敛到这条路径出现故障之前的状态。无论用户是否配置了通过 ECMP 对 IP 组播流量执行负载分割，重新收敛都会发生。

将 ECMP 组播负载分割与 BGP 结合使用

ECMP 组播负载分割采用与通过 BGP 学习到的 RPF 信息，具体方式与其利用通过其他协议学习到的 RPF 信息相同：它会从协议安装的多条路径中选择一条。BGP 的主要区别在于，BGP 默认只会安装一条路径。例如，当一台 BGP 设备针对一个前缀学习到了两个相同的外部 BGP（eBGP）路径，它会选择设备 ID 最低的路径作为最佳路径。接下来，设备会将最佳路径安装到 IP 路由表中。如果设备启用了 BGP 多路径支持特性，而这些 eBGP 路径都是通过同一个相邻 AS 学习到的，那么 BGP 就会将多条路径都添加到 IP 路由表中，而不是只从中选择一条添加。在默认情况下，BGP 只会向 IP 路由表中安装一条路径。

要针对通过 BGP 学习到的路由使用 ECMP 组播负载分割，用户必须启用 BGP 多路径特性。一旦配置之后，当 BGP 向路由表中安装远端写一条信息时，设备就会通过递归的方式执行 RPF 查询，以查找去往这个 BGP 下一跳的最佳下一跳设备（这一点与单播的操作方式相同）。如果某个前缀只有一条 BGP 路径，但却有两条 IGP 路径可以达到 BGP 下一跳，那么组播 RPF 会正确地通过这两条不同的 IGP 路径来实现负载分割。

将 ECMP 组播负载分割与静态组播路由结合使用

如果对于某些源或者 RP，无法使用 IGP 来安装等价路由，那么用户可以配置静态路由来设置等价路径以实现负载分割。用户不能使用静态组播路由来配置等价路由，因为软件不支持针对每个前缀分别配置一条组播路由。针对这种限制，有一些规避的措施可以通过递归路由查询的方式来实现，但是这些规避措施不适用于等价多路径路由转发。

注释： 要想了解更多关于配置静态组播路由的信息，可以在 Inspur INOS IP 组播 FTP 站点参阅 Inspur INOS 中配置多静态组播路由的配置注释，获取这份文件可以访问 ftp://ftpeng.icntnetworks.com/ipmulticast /config-notes/static-mroutes.txt：。

在 IPv4 组播环境中，用户只能给等价多路径配置静态组播路由，但这些静态组播路由只能应用于组播，用户也可以将等价多路径同时用于单播和组播路由，这是没有限制的。用户可以给应用于纯单播路由、纯组播路由，或同时应用于单播和组播路由的静态 IPv6 组播路由配置等价多路径组播路由。

负载分割 IP 组播流量的其他方法

IP 组播流量的负载分割也可以通过将多条平行链路合并为一条隧道，然后再路由组播流量的方式来实现。这种负载分割的方式配置起来比配置 ECMP 组播负载分割更加复杂。在有一种环境下，通过使用 GRE 链路的等价路径配置负载分割可以给网络带来利好，那就是(S,G)或(*,G)条目的总数很小，而每个条目承载的带宽变化又很大，导致工程师手动配置源或 RP 地址也无法保证流量能够合理地实现负载分割的情形。

注释： 在使用 ECMP 组播负载分割时，只有需要针对每个数据包执行负载分担的环境中，用户才需要使用隧道。

用户也可以使用 IP 组播流量来通过接口束（如快速或吉比特 EtherChannel 接口、MLPPP 链

路束或多链路帧中继[FRF.16]束）分担负载。GRE 或其他隧道类型也可以组成这种二层链路束。在使用这种二层机制之前，用户有必要理解单播和组播流量是如何实现负载分割的。
在通过一条隧道实现等价路径 IP 组播流量负载分割之前，用户必须配置 CEF 来针对每个数据包执行负载分担，否则 GRE 数据包就不会按照每数据包的方式获得负载分担。

如何通过 ECMP 分割 IP 组播流量负载

启用 ECMP 组播负载分割

用户可以执行下面的配置任务，来基于源地址跨域多条等价路径对 IP 组播流量执行负载分割。
如果从源出发有两条或多条等价路径，那么设备就会通过这些路径分担单播流量。但在默认情况下，组播流量不会跨越这些等价路径进行负载分割。总地来说，组播流量会沿着 RPF 邻居向下游转发。根据 PIM 的规定，如果有多个邻居拥有相同的度量值，那么前面提到的这个邻居必须拥有最高的 IP 地址。
用户如果使用命令 **ip multicast multipath** 来配置负载分割，那么系统就会使用 S 散列算法基于源地址跨越多条等价路径来对组播流量执行负载分割。如果用户配置了命令 **ip multicast multipath**，且网络中又有多条等价路径，那么设备就会根据源 IP 地址来选择组播流量穿越的路径。从不同的源发出的组播流量会通过不同的等价路径进行发送。对于从同一个源发往不同组播组的组播流量，则不会执行负载分割。
注释： 使用命令 **ip multicast multipath** 可以对流量执行负载分割，但不会平均分配这些流量。从一个源发出的流量只会使用一条路径，即使它发出的流量远远大于其他源发出的流量也是如此。

IP 组播负载分割（ECMP）的前提条件

- 网络中必须有充足的源来启用基于源地址的 ECMP 组播负载分割；
- 网络中要有多条连接 RP 的可用路径，才能配置 ECMP 组播负载分割；

注释： 用户可以使用命令 **show ip route**（并通过参数 *ip-address*）结合源的 IP 地址或者 RP 的 IP 地址来查看是否有多条去往源或 RP 的路径。如果在命令的输出信息中没有看到多条路径，那么用户就不能配置 ECMP 组播负载分割。

- 在 PIM-SM 环境中使用最短路径树（SPT）转发时，那么对于转发所有(S,G)条目，都必须设置 T-bit；
- 在配置 ECMP 组播负载分割之前，最好的做法是使用命令 **show ip rpf** 来查看源是否可以利用 IP 组播多路径功能；
- BGP 默认不能安装多条等价路径。用户可以（例如在 BGP 中）使用命令 **maximum-paths** 来配置多路径。要想了解更多信息，可以参见将 ECMP 组播负载分割与 BGP 结合使用部分。

限制条件

- 如果从源出发有两条或多条等价路径，那么设备就会通过这些路径分担单播流量。但在默认情况下，组播流量不会跨越这些等价路径进行负载分割。总地来说，组播流量会沿着 RPF 邻居向下游转发。根据 PIM 的规定，如果有多个邻居拥有相同的度量值，那么前面提到的这个邻居必须拥有最高的 IP 地址；
- 当同一个 PIM 邻居的 IP 地址可以通过多条等价路径到达时，就不支持配置命令 **ip multicast multipath**。如果网络中使用了没有配置地址的接口，就有可能会出现这种情

况。用户应该配置命令 **ip multicast multipath** 来给所有接口使用不同的 IP 地址；

- 使用命令 **ip multicast multipath** 可以对流量执行负载分割，但不会平均分配这些流量。从一个源发出的流量只会使用一条路径，即使它发出的流量远远大于其他源发出的流量也是如此。

基于源地址启用 ECMP 组播负载分割

用户可以执行下面的配置任务，来（使用 S 散列算法）基于源地址跨域多条等价路径对 IP 组播流量执行负载分割，以利用网络中的多条路径。S 散列算法是可以预测的，因为该算法在计算散列值时不会引入随机数。但 S 散列算法容易受到极化的影响，因为对于一个特定的源来说，无论在哪台设备上计算散列值，获得的结果都是相同的。

注释： 用户要在那些通过多个入站接口接收流量的接收方设备上配置 ECMP 组播负载分割，这一点与单播路由相反。与单播相比，组播在连接多个出站接口的发送方设备上更加灵活。

在开始前

- 网络中必须有充足的源（至少两个源）才能启用基于源地址的 ECMP 组播负载分割；
- 网络中要有多条连接 RP 的可用路径，才能配置 ECMP 组播负载分割；

注释： 用户可以使用命令 **show ip route**（并通过参数 *ip-address*）结合源的 IP 地址或者 RP 的 IP 地址来查看是否有多条去往源或 RP 的路径。如果在命令的输出信息中没有看到多条路径，那么用户就不能配置 ECMP 组播负载分割。

- 在 PIM-SM 环境中使用最短路径树（SPT）转发时，那么对于转发所有(S,G)条目，都必须设置 T-bit；
- 在配置 ECMP 组播负载分割之前，最好的做法是使用命令 **show ip rpf** 来查看源是否可以利用 IP 组播多路径功能；
- BGP 默认不能安装多条等价路径。用户可以（例如在 BGP 中）使用命令 **maximum-paths** 来配置多路径。要想了解更多信息，可以参见将 ECMP 组播负载分割与 BGP 结合使用部分。

总步骤

1. enable

2. configure terminal

3. ip multicast multipath

4. 在冗余拓扑中，在所有设备上重复步骤3

5. exit

6. show ip rpf *source-address* [*group-address*]

7. show ip route *ip-address*

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip multicast multipath**	启用使用 S 散列算法的、基于源地址的 ECMP 组播

	示例： `Device(config)# ip multicast multipath`	负载分割。 • 由于这条命令会修改 RPF 邻居的选择方式，在冗余拓扑中，所有设备上关于这条命令的配置应该一直，以防止网络中出现环路； • 当同一个 PIM 邻居的 IP 地址可以通过多条等价路径到达时，就不支持配置这条命令。如果网络中使用了没有配置地址的接口，就有可能会出现这种情况。在配置了这条命令的设备上，要给每个接口配置不同的 IP 地址； • 使用这条命令可以对流量执行负载分割，但不会平均分配这些流量。从一个源发出的流量只会使用一条路径，即使它发出的流量远远大于其他源发出的流量也是如此
步骤 4	在冗余拓扑中，在所有设备上重复步骤 3	——
步骤 5	**exit** 示例： `Device(config)# exit`	离开全局配置模式并返回特权 EXEC 模式
步骤 6	**show ip rpf** *source-address* [*group-address*] 示例： `Device# show ip rpf 10.1.1.2`	（可选）显示 IP 组播路由用于 RPF 校验的信息。 • 用户可以使用这条命令来验证 RPF 的选择，以确保 IP 组播流量的负载分割执行无误
步骤 7	**show ip route** *ip-address* 示例： `Device# show ip route 10.1.1.2`	（可选）显示 IP 路由表当前的状态。 • 用户可以使用这条命令来验证有多条路径通向源或 RP，可以实现 ECMP 负载分割； • 在 *ip-address* 部分输入源的 IP 地址，以确认（对于最短路径树而言）有多条可用路径通向源或 RP 的 IP 地址，以及（对于共享树而言）有多条可用路径通向 RP

启用基于源和组地址的 ECMP 组播负载分割

用户可以执行下面的配置任务，来针对组播流量启用（使用基本 S-G 散列算法的）基于源和组地址的 ECMP 组播负载分割，以利用网络中的多条路径。基本的 S-G 散列算法是可以预测的，因为该算法在计算散列值时不会引入随机数。但基本的 S-G 散列算法容易受到极化的影响，因为对于一个特定的源和组来说，无论在哪台设备上计算散列值，获得的结果都是相同的。

基本的 S-G 散列算法可以对 ECMP 组播负载分割提供比 S 散列算法更加灵活的支持。如果为执行负载分割而使用基本的 S-G 散列算法，可以让设备在组发送数据流或者向多条频道发送广播（如 IPTV 服务器或 MPEG 视频服务器）时，能够更加有效地通过等价路径实现负载分割。

注释： 用户要在那些通过多个入站接口接收流量的接收方设备上配置 ECMP 组播负载分割，这一点与单播路由相反。与单播相比，组播在连接多个出站接口的发送方设备上更加灵

活。

总步骤

1. enable

2. configure terminal

3. ip multicast multipath s-g-hash basic

4. 在冗余拓扑中，在所有设备上重复步骤3

5. exit

6. show ip rpf *source-address* [*group-address*]

7. show ip route *ip-address*

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip multicast multipath s-g-hash basic** 示例： `Device(config)# ip multicast multipath s-g-hash basic`	启用使用基本 S-G 散列算法的、基于源和组地址的 ECMP 组播负载分割。 • 由于这条命令会修改 RPF 邻居的选择方式，在冗余拓扑中，所有设备上关于这条命令的配置应该一直，以防止网络中出现环路
步骤 4	在冗余拓扑中，在所有设备上重复步骤 3	——
步骤 5	**exit** 示例： `Device(config)# exit`	离开全局配置模式并返回特权 EXEC 模式
步骤 6	**show ip rpf** *source-address* [*group-address*] 示例： `Device# show ip rpf 10.1.1.2`	（可选）显示 IP 组播路由用于 RPF 校验的信息。 • 用户可以使用这条命令来验证 RPF 的选择，以确保 IP 组播流量的负载分割执行无误
步骤 7	**show ip route** *ip-address* 示例： `Device# show ip route 10.1.1.2`	（可选）显示 IP 路由表当前的状态。 • 用户可以使用这条命令来验证有多条路径通向源或 RP，可以实现 ECMP 负载分割； • 在 *ip-address* 部分输入源的 IP 地址，以确认（对于最短路径树而言）有多条可用路径通向源或 RP 的 IP 地址，以及（对于共享树而言）

		有多条可用路径通向 RP

基于源组和下一跳地址启用 ECMP 组播负载分割

用户可以执行下面的配置任务，来针对组播流量启用（使用基于下一跳的 S-G 散列算法的）基于源、组和下一跳地址的组播负载分割，以利用网络中的多条路径。基于下一跳的 S-G 散列算法是可以预测的，因为在计算散列值的过程中，没有任何随机数的参与。基于下一跳的 S-G 散列算法所采用的散列机制与 S-散列算法和基本的 S-G 散列算法不同，这种机制不会受到极化问题的影响。

基于下一跳的 S-G 散列算法可以对 ECMP 组播负载分割提供比 S 散列算法更加灵活的支持，以消除极化问题带来的影响。如果为执行 ECMP 负载分割而使用基于下一跳的 S-G 散列算法，可以让设备在组发送数据流或者向多条频道发送广播（如 IPTV 服务器或 MPEG 视频服务器）时，能够更加有效地通过等价路径实现负载分割。

总步骤

1. enable

2. configure terminal

3. ip multicast multipath s-g-hash next-hop-based

4. 在冗余拓扑中，在所有设备上重复步骤1到步骤3

5. end

6. show ip rpf *source-address* [*group-address*]

7. show ip route *ip-address*

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip multicast multipath s-g-hash next-hop-based** 示例： `Router(config)# ip multicast multipath s-g-hash next-hop-based`	启用使用基于下一跳的 S-G 散列算法的、基于源、组和下一跳地址的 ECMP 组播负载分割。 • 由于这条命令会修改 RPF 邻居的选择方式，在冗余拓扑中，所有设备上关于这条命令的配置应该一直，以防止网络中出现环路。 **注释：**用户要在那些通过多个入站接口接收流量的接收方设备上配置 **ip multicast multipath** 这条命令，这一点与单播路由相反。与单播相比，组播在连接多个出站接口的发送方设备上更加灵活。
步骤 4	在冗余拓扑中，在所有设备上重复步骤 1 到步骤 3	——
步骤 5	**end** 示例： `Device(config)# exit`	离开全局配置模式并返回特权 EXEC 模式

步骤 6	**show ip rpf** *source-address* [*group-address*] 示例： `Device# show ip rpf 10.1.1.2`	（可选）显示 IP 组播路由用于 RPF 校验的信息。 • 用户可以使用这条命令来验证 RPF 的选择，以确保 IP 组播流量的负载分割执行无误
步骤 7	**show ip route** *ip-address* 示例： `Device# show ip route 10.1.1.2`	（可选）显示 IP 路由表当前的状态。 • 用户可以使用这条命令来验证有多条路径通向源或 RP，可以实现 ECMP 负载分割； • 在 *ip-address* 部分输入源的 IP 地址，以确认（对于最短路径树而言）有多条可用路径通向源或 RP 的 IP 地址，以及（对于共享树而言）有多条可用路径通向 RP

通过 ECMP 分割 IP 组播流量的配置示例

基于源地址的 ECMP 组播负载分割的示例

下面的示例显示了如何在一台路由器上启用使用 S 散列算法的、基于源地址的 ECMP 组播负载分割：

```
ip multicast multipath
```

基于源和组地址的 ECMP 组播负载分割的示例

下面的示例显示了如何在一台路由器上启用使用基本 S-G 散列算法的、基于源和组地址的 ECMP 组播负载分割：

```
ip multicast multipath s-g-hash basic
```

基于源组和下一跳地址的 ECMP 组播负载分割的示例

下面的示例显示了如何在一台路由器上启用使用基于下一跳的 S-G 散列算法的、基于源、组和下一跳地址的 ECMP 组播负载分割：

```
ip multicast multipath s-g-hash next-hop-based
```

其他参考资料

相关文档

相关主题	文档名
Inspur INOS 命令	《Inspur INOS 主命令列表，所有版本》
Inspur INOS IP 组播命令	《Inspur INOS IP 组播命令参考手册》

标准与 RFC

标准/RFC	标题
RFC 4601	协议独立组播稀疏模式（PIM-SM）：协议标准

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

通过 ECMP 分割 IP 组播流量的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

IP 组播优化：基于 SSM 信道的组播过滤

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导

航系统。

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

基于 SSM 信道的组播边界过滤的前提条件

用户按照《IP 组播：PIM 配置指南》中“配置基本 IP 组播”一部分描述的方法在设备上启用了 IP 组播。

关于基于 SSM 信道的组播边界过滤的信息

组播边界的规则

基于 SSM 信道的组播边界过滤特性对命令 **ip multicast boundary** 进行了扩展，以提供对控制平面过滤的支持。用户可以在一个接口上配置多条 **ip multicast boundary** 命令。
命令 **ip multicast boundary** 的规则如下：

- 每个接口上只能配置一个关键字 **in** 和一个关键字 **out** 的实例；
- 关键字 **in** 和 **out** 可用于标准访问列表或扩展访问列表；
- 只有在配置标准访问列表时，可以使用关键字 **filter-autorp** 也可以不带关键字；
- 在一个接口上的命令最多可以包含 3 个实例：一个 **in** 实例，一个 **out** 实例，一个 **filter-autorp** 实例或无关键字实例；
- 在使用包含多实例的命令时，过滤的效果也是这些实例累加的。如果一个边界语句上没有关键字，另一个边界语句携带 **in** 关键字，在将这两个访问列表都应用于入站方向上之后，只有有一条匹配即足够；
- 命令的所有实例会同时应用于控制平面流量和数据平面流量；
- 设备会解析扩展访问列表的协议信息，以实现列表复用与流量过滤的一致性。在上面叙述的所有条件下，扩展访问列表都会过滤某个操作(S,G)，只要访问列表过滤所有协议的(S,G)流量。

基于 SSM 信道的组播边界过滤的好处

- 这个特性支持应用于源接口的入站方向上；

- 访问控制功能对于 SSM 和对于任意源组播（ASM）是相同的。

如何配置基于 SSM 信道的组播边界过滤

配置组播边界

总步骤

1. enable

2. configure terminal

3. ip access-list {**standard**| **extended**} *access-list-name*

4. permit *protocol* **host** *address* **host** *address*

5. deny *protocol* **host** *address* **host** *address*

6. 根据需要重复步骤4或步骤5

7. interface *type* **interface-number** *port -number*

8. ip multicast boundary *access-list-name* [**in**| **out** | **filter-autorp**]

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip access-list** {**standard**| **extended**} *access-list-name* 示例： `Device(config)# ip access-list 101`	配置标准或扩展访问列表
步骤 4	**permit** *protocol* **host** *address* **host** *address* 示例： `Device(config-ext-nacl)# permit ip host 181.1.2.201 host 232.1.1.11`	允许指定的 IP 主机流量
步骤 5	**deny** *protocol* **host** *address* **host** *address* 示例：	拒绝指定的组播 IP 组与源流量

	Device(config-acl-nacl)# deny ip host 181.1.2.203 host 232.1.1.1	
步骤 6	根据需要重复步骤4或步骤5	允许和拒绝指定的主机和源流量
步骤 7	**interface** *type* **interface-number** *port -number* 示例： Device(config)# interface gigabitethernet 2/3/0	进入接口配置模式
步骤 8	**ip multicast boundary** *access-list-name* [**in**| **out** | **filter-autorp**] 示例： Device(config-if)# ip multicast boundary acc_grp1 out	配置组播边界。 **注释：**扩展访问列表不支持关键字 **filter-autorp**

基于 SSM 信道的组播边界过滤的配置示例

配置组播边界来放行和拒绝流量的示例

下面的实例放行了所有去往(181.1.2.201 232.1.1.1)和(181.1.2.202 232.1.1.1)的出站流量，同时拒绝了所有其他的(S,G)。

```
configure terminal
ip access-list extended acc_grp1
permit ip host 0.0.0.0 232.1.1.1 0.0.0.255
permit ip host 181.1.2.201 host 232.1.1.1
permit udp host 181.1.2.202 host 232.1.1.1
permit ip host 181.1.2.202 host 232.1.1.1
deny igmp host 181.2.3.303 host 232.1.1.1
interface gigabitethernet 2/3/0
ip multicast boundary acc_grp1 out
```

配置组播边界来放行流量的示例

下面的实例放行了去往(192.168.2.201 232.1.1.5)和(192.168.2.202 232.1.1.5)的流量。

```
configure terminal
ip access-list extended acc_grp6
permit ip host 0.0.0.0 232.1.1.1 5.0.0.255
```

```
deny udp host 192.168.2.201 host 232.1.1.5
permit ip host 192.168.2.201 host 232.1.1.5
deny pim host 192.168.2.201 host 232.1.1.5
permit ip host 192.168.2.202 host 232.1.1.5
deny igmp host 192.2.3.303 host 232.1.1.1
interface gigabitethernet 2/3/0
ip multicast boundary acc_grp6 out
```

配置组播边界来拒绝流量的示例

下面的实例拒绝了候选 RP 宣告的一个组范围。由于这个组范围被拒绝，因此设备没有创建 pim auto-rp 映射。

```
configure terminal
ip access-list standard acc_grp10
deny 225.0.0.0 0.255.255.255
permit any
access-list extended acc_grp12
permit pim host 181.1.2.201 host 232.1.1.8
deny udp host 181.1.2.201 host 232.1.1.8
permit pim host 181.1.2.203 0.0.0.255 host 227.7.7.7
permit ip host 0.0.0.0 host 227.7.7.7
permit ip 181.1.2.203 0.0.0.255 host 227.7.7.7
permit ip host 181.1.2.201 host 232.1.1.7
ip access-list extended acc_grp13
deny ip host 181.1.2.201 host 232.1.1.8
permit ip any any
interface gigabitethernet 2/3/0
ip multicast boundary acc_grp10 filter-autorp
ip multicast boundary acc_grp12 out
ip multicast boundary acc_grp13 in
```

其他参考资料

相关文档

相关主题	文档名
Inspur INOS 命令	《Inspur INOS 主命令列表，所有版本》
Inspur INOS IP 组播命令	《Inspur INOS IP 组播命令参考手册》

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。	http://www.icntnetworks.com

用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	

基于 SSM 信道的组播边界过滤的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

IP 组播优化：PIM 密集模式状态刷新

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

PIM 密集模式状态刷新的前提条件

在配置 PIM 密集模式状态刷新特性之前，用户必须首先在一个接口启用 PIM 密集模式。

PIM 密集模式状态刷新的限制条件

- 在一个 PIM 密集模式网络中，所有路由器都必须运行支持 PIM 密集模式状态刷新特性的软件版本，才能处理和转发状态刷新控制消息；
- 在同一个局域网中，所有 PIM 路由器上使用的状态刷新控制消息原始时间间隔都必须相同。具体来说，所有与局域网直连的路由器接口上都必须配置相同的原始时间间隔。

关于 PIM 密集模式状态刷新的信息

PIM 密集模式状态刷新概述

PIM 密集模式状态刷新特性是对 PIMv2 组播路由架构的一种扩展。
PIM 密集模式会建立基于源的组播分发树，这棵树的操作原则是泛洪并修剪。从源发出的组播数据包会泛洪到 PIM 密集模式网络的所有区域中。接收到组播数据包，且没有直连组播组成员或 PIM 邻居的 PIM 路由器会沿着基于源的分发树，向数据包的源发送一条修剪消息。因此，后续的组播数据包不会再泛洪到那些已经被修剪掉的分发树的叶网络当中。然而，PIM 密集模式中被修剪掉的状态大约每 3 分钟就会超时，而整个 PIM 密集模式网络中会重新泛洪组播数据包和修剪消息。在 PIM 密集模式网络中重新泛洪不需要的流量会占用网络带宽。PIM 密集模式状态刷新特性会确保 PIM 密集模式中的修剪状态不会超时，因此这种特性会周期性地沿着基于源的分发树转发控制消息。控制消息会在共享树中每台路由器的出站接口上刷新修剪状态。

PIM 密集模式状态刷新的好处

PIM 密集模式状态刷新特性会确保 PIM 密集模式中的修剪状态不会超时，这样可以大大减少不需要的流量被重新泛洪到那些已经被修剪掉的叶网络中，因此可以节省网络带宽。这种特性可以在默认的 3 分钟状态刷新超时周期之前，让 PIM 密集模式组播网络中的 PIM 路由器发现拓扑的变化（即源加入或离开组播组）。

如何配置 PIM 密集模式状态刷新

配置 PIM 密集模式状态刷新

用户不需要通过配置来启用 PIM 密集模式状态刷新特性。在默认情况下，只要 PIM 路由器运行的 Inspur INOS XE 软件系统支持 PIM 密集模式状态刷新特性，那么这台路由器就会自动处理并转发状态刷新控制消息。
要想在一台 PIM 路由器上禁用对状态刷新控制消息的处理和转发，可以使用全局配置命令 **ip pim state-refresh disable**。如果希望在状态刷新被禁用之后再次启用这项特性，可以使用全局配置命令 **no ip pim state-refresh disable**。
发起状态刷新控制消息的操作默认是禁用的。要在 PIM 路由器上配置发起控制消息，可以

从全局配置模式下配置下列命令：

命令	目的
`Router(config)#` **interface** *type number*	指定一个接口，并且进入路由器的接口配置模式
`Router(config-if)#` **ip pim state-refresh origination-interval** [*interval*]	配置 PIM 密集模式状态刷新控制消息的发起。用户可以根据需要，使用参数 *interval* 来配置设备发送控制消息之间的秒数。默认时间间隔为 60 秒。时间间隔的取值范围是从 1 秒到 100 秒

查看 PIM 密集模式状态刷新配置

用户可以使用命令 **show ip pim interface** [*type number*] **detail** 和命令 **show ip pim neighbor** [*interface*]来验证 PIM 密集模式状态刷新特性是否配置无误。命令 **show ip pim interface** [*type number*] **detail** 的输出信息显示，状态刷新控制消息的处理、转发和发起已经启用。

```
Router# show ip pim interface fastethernet 0/1/0 detail
FastEthernet0/1/0 is up, line protocol is up
Internet address is 172.16.8.1/24
Multicast switching:process
Multicast packets in/out:0/0
Multicast boundary:not set
Multicast TTL threshold:0
PIM:enabled
PIM version:2, mode:dense
PIM DR:172.16.8.1 (this system)
PIM neighbor count:0
PIM Hello/Query interval:30 seconds
PIM State-Refresh processing:enabled
PIM State-Refresh origination:enabled, interval:60 seconds
PIM NBMA mode:disabled
PIM ATM multipoint signalling:disabled
PIM domain border:disabled
Multicast Tagswitching:disabled
```

在命令 **show ip pim neighbor** [*interface*]的输出信息中，Mode 一列的 S 表示邻居已经配置了 PIM 密集模式状态刷新特性。

```
Router# show ip pim neighbor
PIM Neighbor Table
Neighbor Interface Uptime/Expires Ver DR
Address Priority/Mode
172.16.5.1 Ethernet1/1 00:09:03/00:01:41 v2 1 / B S
```

PIM DM 状态刷新的监控与维护

下面是在用户针对组播组 239.0.0.1 配置了特权 EXEC 命令 **debug ip pim** 之后，PIM 路由器发

送和接收到的 PIM 密集模式状态刷新控制消息：

```
Router# debug ip pim 239.0.0.1
*Mar 1 00:25:10.416:PIM:Originating refresh message for
(172.16.8.3,239.0.0.1)
*Mar 1 00:25:10.416:PIM:Send SR on GigabitEthernet1/1/0 for (172.16.8.3,239.0.0.1)
TTL=9
```

下面是在修改了吉比特以太网接口 1/0/0 和组播组 239.0.0.1 的修剪计时器之后，命令 **show ip mroute** 提供的输出信息。（下面的输出信息默认路由器上已经提前配置了特权 EXEC 命令 **debug ip pim**）在第一条命令 **show ip mroute** 的输出信息中，可以看到修剪计时器读数为 00:02:06。调试消息显示了以太网接口 1/0 上接收和发送的 PIM 密集模式状态刷新控制消息，同时其他 PIM 密集模式状态刷新路由器也被发现。在第二条命令 **show ip mroute** 的输出信息中，可以看到修剪计时器被重置为了 00:02:55。

```
Router# show ip mroute 239.0.0.1
(172.16.8.3, 239.0.0.1), 00:09:50/00:02:06, flags:PT
Incoming interface:GigabitEthernet1/1/0, RPF nbr 172.16.5.2
Outgoing interface list:
GigabitEthernet1/0/0, Prune/Dense, 00:09:43/00:02:06
Router#
*Mar 1 00:32:06.657:PIM:SR on iif from 172.16.5.2 orig 172.16.8.1 for
(172.16.8.3,239.0.0.1)
*Mar 1 00:32:06.661: flags:prune-indicator
*Mar 1 00:32:06.661:PIM:Cached metric is [0/0]
*Mar 1 00:32:06.661:PIM:Keep RPF nbr 172.16.5.2
*Mar 1 00:32:06.661:PIM:Send SR on Ethernet1/0 for (172.16.8.3,239.0.0.1)
TTL=8
*Mar 1 00:32:06.661: flags:prune-indicator
Router# show ip mroute 239.0.0.1
(172.16.8.3, 239.0.0.1), 00:10:01/00:02:55, flags:PT
Incoming interface:GigabitEthernet1/1/0, RPF nbr 172.16.5.2
Outgoing interface list:
GigabitEthernet1/0/0, Prune/Dense, 00:09:55/00:02:55
```

PIM 密集模式状态刷新的配置示例

发起处理与转发 PIM 密集模式状态刷新控制消息的示例

下面的示例是让一台 PIM 路由器的快速以太网接口 0/1/0 每 60 秒发起、处理和转发 PIM 密集模式状态刷新控制消息的相关配置：

```
ip multicast-routing distributed
interface FastEthernet0/1/0
ip address 172.16.8.1 255.255.255.0
ip pim state-refresh origination-interval 60
ip pim dense-mode
```

处理和转发 PIM 密集模式状态刷新控制消息的示例

下面的实例是让一台 PIM 路由器仅在快速以太网接口 1/1/0 上处理和转发 PIM 密集模式状态刷新控制消息的配置。

```
ip multicast-routing
interface FastEthernet1/1/0
ip address 172.16.7.3 255.255.255.0
ip pim dense-mode
```

其他参考资料

相关文档

相关主题	文档名
PIM 密集模式状态刷新特性是一种对 PIMv2 组播路由架构的扩展	“配置基本 IP 组播”部分
IP 组播命令：完整的命令语法、命令模式、默认状态、命令历史、使用指南和示例	《Inspur INOS IP 组播命令参考手册》

标准

RFC	文档名
这个特性没有新的标准，也没有修订的标准。关于这个特性，对当前标准的支持也没有变化	—

RFC

RFC	文档名
这个特性没有新的 RFC，也没有修订的 RFC。关于这个特性，对当前标准的支持也没有变化	—

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

PIM 密集模式状态刷新的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

PIM 组播优化：IGMP 状态限制

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

IGMP 状态限制的前提条件

- 用户按照《IP 组播：PIM 配置指南》中“配置基本 IP 组播”一部分描述的方法在设备上启用了 IP 组播和协议独立组播（PIM）接口；
- 必须配置好所有的 ACL。要想了解更多信息，参见《安全配置指南：访问控制列表》指南的“创建一个 IP 访问列表并将其应用到接口”一部分的描述。

IGMP 状态限制的限制条件

用户在每台设备上只能配置一条全局限制，在每个接口上只能配置一条限制。

关于 IGMP 状态限制的信息

IGMP 状态限制

IGMP 状态限制特性可以让用户配置 IGMP 状态限制器（limiter），这种策略可以在全局或接口上对 IGMP 成员关系报告（IGMP 加入消息）中获得的组播状态设置限制。超过用户配置限制的成员关系报告不会再进入 IGMP 缓存当中。这个特性可以用来防止 DoS 攻击，或者在网络中提供一种组播 CAC 机制，让所有组播流量占用的带宽总量大体相当。

注释： IGMP 状态限制器可以在全局或接口上对通过 IGMP、IGMPv3lite 和 URL 汇集发现（URD）成员关系报告中获得的组播状态设置限制。

IGMP 状态限制特性的设计

- 在全局配置模式配置 IGMP 状态限制器可以对设备能够进行缓存的 IGMP 成员关系报告总数设置一个全局的限制数；
- 在接口配置模式配置 IGMP 状态限制器可以针对接口的 IGMP 成员关系报告数量设置一个限制数；
- 用户可以使用 ACL 来防止将一些组或信道计入到接口限制数量当中。在这里，用户可以设置标准 ACL 或扩展 ACL。标准 ACL 可以用来定义哪些(*,G)状态不计入接口限制数中。扩展 ACL 可以用来定义哪些(S,G)状态不计入接口限制数中。用户也可以采用将源地址和源反掩码设置为 0.0.0.0 的方式，在扩展访问列表中定义这些称为(0,G)的语句，来使用扩展 ACL 来定义哪些(*,G)状态不计入接口限制数中；
- 用户在每台设备上只能配置一条全局限制，在每个接口上只能配置一条限制。

IGMP 状态限制器的构成

IGMP 状态限制器的构成为：

- 每当路由器接收到一条某个组或信道的 IGMP 成员关系报告时，Inspur INOS 软件会对报告进行校验，以判断全局 IGMP 状态限制器或接口 IGMP 状态限制器是否已经达到了限制数量；
- 只有当用户配置的全局 IGMP 状态限制器没有达到限制时，设备才会进一步处理 IGMP 成员关系报告。在达到配置的限制数时，后续的 IGMP 成员关系报告就会被忽略（丢弃），同时设备会按照下面的格式生成一条告警消息：
  - `%IGMP-6-IGMP_GROUP_LIMIT: IGMP limit exceeded for <group (*, group address)> on <interface type number> by host <ip address>`
  - `%IGMP-6-IGMP_CHANNEL_LIMIT: IGMP limit exceeded for <channel (source address, group address)> on <interface type number> by host <ip address>`
- 只有用户针对接口配置了 IGMP 状态限制器时，设备才会在配置了限制器的接口对限制数量进行技术；
- 如果用户同时配置了全局 IGMP 状态限制器和基于接口的 IGMP 状态限制器，那么设备也会实施基于接口的 IGMP 状态限制器，但是这些限制策略也要满足全局限制数量的限制。

如何配置 IGMP 状态限制

配置 IGMP 状态限制器

注释： IGMP 状态限制器可以在全局或接口上对通过 IGMP、IGMPv3lite 和 URL 汇集发现（URD）成员关系报告中获得的组播状态设置限制。

配置全局 IGMP 状态限制器

用户可以根据需要，来针对设备配置全局 IGMP 状态限制器。

总步骤

1. enable

2. configure terminal

3. ip igmp limit *number*

4. end

5. show ip igmp groups

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**ip igmp limit** *number* 示例： `Device(config)# ip igmp limit 150`	针对从 IGMP 成员关系报告（IGMP 加入消息）中得到的组播路由条目配置总数量限制
步骤 4	**end** 示例： `Device(config-if)# end`	中断当前的配置会话并返回特权 EXEC 模式
步骤 5	**show ip igmp groups** 示例： `Device# show ip igmp groups`	（可选）显示包含（与设备直连且通过 IGMP 学习到的）接收方的组播组

针对接口配置 IGMP 状态限制器

用户可以根据需要，来针对接口配置 IGMP 状态限制器。

总步骤

1. enable

2. configure terminal

3. interface *type number*

4. ip igmp limit *number* [**except** *access-list*]

5. 执行下列操作之一：

- **exit**
- **end**

6. show ip igmp interface [*type number*]

7. show ip igmp groups

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *type number* 示例： `Device(config)# interface GigabitEthernet0/0`	选择一台与主机相连的接口，进入接口的配置模式
步骤 4	**ip igmp limit** *number* [**except** *access-list*] 示例： `Device(config-if)# ip igmp limit 100`	针对该接口从 IGMP 成员关系报告（IGMP 加入消息）中得到的组播路由条目配置数量限制

步骤 5	**执行下列操作之一：**
	• **exit**
	• **end**
	
	示例：
	`Device(config-if)# exit`
	`Device(config-if)# end`
		• （可选）中断当前的配置会话并返回全局配置模式。用户可以重复步骤 3 和步骤 4 来给另一个接口配置基于接口的限制器；
		• 中断当前的配置会话并返回特权 EXEC 模式

步骤 6	**show ip igmp interface** [*type number*] 示例： `Device# show ip igmp interface`	（可选）显示关于 IGMP 状态和配置，以及接口组播路由的信息
步骤 7	**show ip igmp groups**	（可选）显示包含（与设备直连且通过 IGMP 学习到的）接收方的组播组

	示例： `Device# show ip igmp groups`	

IGMP 状态限制的配置示例

下面的示例显示了如何通过配置 IGMP 状态限制器在网络中提供组播 CAC 机制，让所有组播流量占用的带宽总量大体相当。
这个实例使用了下图所示的拓扑。
注释： 虽然下图和下面的示例在配置中使用得到都是路由器，但用户可以使用任意设备（路由器和交换机）。

图 41：IGMP 状态限制实例的拓扑

SDTV channels	SDTV 频道
300 channels	300 个频道
Multicast Video(50%)	组播视频（50%）
Voice, Internet & VoD(50%)	语音，互联网与 VoD（50%）
250-500 users per DSLAM	每个 DSLAM 250-500 个用户

在本例中，服务提供商正在提供 300 个标准定义（SD）TV 频道。每个 TV 频道大约使用 4Mbps。服务提供商必须在与 DSLAM（数字用户线路接入复用器）相连的 PE 路由器上按照下列方式提供吉比特以太网接口：50%的链路带宽（500Mbps）必须用于给用户提供互联网、语音和按需视频（VoD）服务，剩下的 50%链路带宽则必须给用于为用户提供 SD 频道服务。
由于每条 SD 频道需要使用相同的带宽，因此用户可以使用基于接口的 IGMP 状态限制器来实现必要的 CAC，让服务提供商提供所需的服务。要计算出每个接口所需的 CAC，应该用频道的总数除以 4（因为每个频道都是 4Mbps 带宽）。因此，每个接口所需的 CAC 就是：
500Mbps/4Mbps=125 组播路由
一旦计算出来了所需的 CAC，服务提供商就可以使用计算的结果来配置 PE 路由器上每个吉比特以太网的 IGMP 状态限制器。服务提供商必须根据网络的 CAC 需求，来将可以通过吉比特以太网接口（随时）传输的 SD 频道限制为 125。给 SD 频道将基于接口的 IGMP 状态限制器配置为 125，可以让接口提供 500Mbps 的带宽，这 50%就是链路带宽当中必须给 SD 频道保留的那一部分。
下面的配置显示了服务提供商如何使用基于接口的组播路由器限制器，让吉比特以太网接口 0/0/0 确保提供给用户的 SD 频道和互联网、语音与 VoD 服务：

```
interface GigabitEthernet0/0/0
description --- Interface towards the DSLAM ---
.
.
.
ip igmp limit 125
```

其他参考资料

相关文档

相关主题	文档名

Inspur INOS 命令	《Inspur INOS 主命令列表，所有版本》
Inspur INOS IP 组播命令	《Inspur INOS IP 组播命令参考手册》

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com

IGMP 状态限制的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入该特性

第 8 部分

配置生成树协议

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com/go/cfn。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

STP 的限制条件

- 如果成为根设备必须的值小于 1，那么这台设备就无法成为根设备；
- 如果网络中由一些支持扩展系统 ID 的设备，和一些不支持扩展系统 ID 的设备组成，那么包含扩展系统 ID 的设备就不太可能成为根设备。每当 VLAN 编号大于运行老板系统的设备的优先级值，扩展系统 ID 会增加设备的优先级值；
- 对每个生成树实例来说，根设备都应该是骨干设备或分布层设备。不要将接入层设备配置为生成树的主用根；
- 用户不能混合使用 Inspur 3850 交换机和 Inspur 6650 交换机来建立交换机堆栈。

关于生成树协议的信息

生成树协议

生成树协议（STP）是一项二层管理协议，它的作用是在提供路径冗余的同时防止网络中出现环路。要想让二层以太网络能够正常工作，两个工作站之间只能有一条活动路径。终端站点之间有多条活动路径，网络中就会出现环路。如果网络中存在环路，那么终端工作站可能就会接收到重复消息。设备还有可能会通过多个二层接口学习到 MAC 地址。这种情况会导致网络不稳定。生成树操作对于终端工作站来说是透明的，也就是说终端工作站无法检测出它们是连接到了一个局域网段，还是连接到了一个包含多个网段的交换型局域网。

STP 会使用生成树算法从具有冗余连接的网络中选择出一台设备来充当生成树的根。这种算法可以依据端口在活动拓扑中发挥的作用来给端口指定一个角色。通过这种方法，算法可以在交换型二层网络中计算出最佳的无环路径。这些角色包括：

- 根：生成树拓扑中选举出来的转发端口；
- 指定：给每个交换型局域网段选举出来的转发端口；
- 替代：在生成树中提供通往根桥的替代路径的阻塞端口；
- 备份：在环回配置中的阻塞端口。

那些*所有*端口皆为指定角色或备份角色的设备即为根设备。而那些至少有一个端口为指定角色的设备则称为指定设备。

生成树会强制让冗余数据路径进入备份（即阻塞）状态。如果生成树中的一个网段失效，而网络中又存在冗余路径的话，那么生成树算法就会重新计算生成树拓扑，并且激活备份路径。设备会以固定的时间间隔发送和接收生成树数据帧，这些数据帧称为桥协议数据单元（BPDU）。设备不会把这些数据帧转发给其他设备，而是会通过这些数据帧来建立无环的路径。BDPU 中包含关于发送方设备及其端口的信息，其中包括设备和 MAC 地址、设备的优先级、端口的优先级以及路径开销。生成树会使用这些信息来给交换网络选举根设备和根端口，以及给每个交换网段选举根端口和指定端口。

当一台设备上有两个端口同处某个环路中，设备就会通过生成树和路径开销设置来判断将其中的哪个端口置入转发状态，将其中的哪个端口置入阻塞状态。生成树端口优先级值可以代表这个端口在网络拓扑中的位置，以及其位置对于转发流量的优越程度。路径靠小指则代表了媒体的速率。

注释： 在默认情况下，设备只会在那些没有安装 SFP（小型可插拔）模块的接口上发送保活消息（来确保连接的连通性）。用户可以通过输入接口配置命令（不带其他关键字）**[no] keepalive** 来修改接口的这种默认操作。

生成树拓扑与 BPDU

下列因素共同构成了一个交换型网络稳定、活动的拓扑结构：

- 每台设备各个 VLAN 所关联的唯一的桥 ID（由设备优先级和 MAC 地址组成）。在设备堆栈中，所有设备在一个生成树实例中都会使用相同的桥 ID；
- 去往根设备的生成树路径开销；
- 每个二层接口所关联的端口标识符（由端口优先级和 MAC 地址组成）。

当网络中一台设备启动时，它们都会按照根设备的方式进行操作。每台设备都会通过所有的端口发送一条配置 BPDU。BPDU 的功能是发起通信并计算这个生成树的拓扑。每个配置 BDPU 中都会包含下列信息：

- 发送方设备认为是根设备的那台设备的唯一桥 ID；

- 去往根的生成树路径开销；
- 发送方设备的桥 ID；
- 消息老化值；
- 发送方接口的标识符；
- hello、转发延迟和最大老化协议计时器值。

当一台设备接收到一个包含更优信息（即桥 ID 更低，路径开销更低等）的配置 BDPU 时，它会将该端口的信息保存下来。如果这个 BPDU 是通过设备的根端口接收到的，那么设备也会将它经过更新，通过所有其作为指定设备的直连局域网段发送出去。

当一台设备接收到一个比当前为该端口保存的配置 BPDU 包含更差信息（即桥 ID 更低，路径开销更低等）的配置 BDPU 时，它就会丢弃这个 BPDU。如果这台设备是它接收到 BPDU 的那个局域网段的指定设备，那么它就会向这个局域网段中发送一条包含为该端口保存的最新信息的 BPDU。通过这种方式，较差的信息就会被丢弃，网络中传播都是较优的信息。

交换 BPDU 可以获得下面的效果：

- 网络中的一台设备会被选举为根设备（即一个交换型网络中生成树拓扑的逻辑中心）。详见下面的拓扑图；

  对于每个 VLAN 来说，拥有最高设备优先级（即优先级数值最低）的设备会被选举为根设备。如果所有设备上使用的都是默认优先级（32768），那么这个 VLAN 中 MAC 地址最低的设备就会成为根设备。设备优先级值在桥 ID 中占据了最重要的比特位，如下图的图所示。
- （除根设备之外）每台设备会选举出根端口。当设备向根设备转发数据包时，这个端口可以提供最佳的（也就是开销值最低的）路径。

  在设备堆栈中选择根端口时，生成树执行的操作为：
  - 选择根桥 ID 最低的端口；
  - 选择去往根设备路径开销最低的端口；
  - 选择指定桥 ID 最低的端口；
  - 选择指定路径开销最低的端口；
  - 选择端口 ID 最低的端口
- 在堆栈根设备中，只有一个出站端口会被选为根端口。堆栈中剩下的设备都会成为它的指定设备，如下图（中的设备 2 和设备 3）所示；
- 每台设备会根据路径开销来计算去往根交换机的最短距离；
- 每个局域网段会选出一台指定设备。在从这个局域网向根设备转发数据包时，指定设备是路径开销最低的设备。指定设备连接局域网的端口称为指定端口。

一个堆栈成员会被选举为堆栈的根设备。堆栈根设备中会包含出站的根端口（设备 1）。

图 42：设备堆栈中的生成树端口状态

Switch stack	交换机堆栈
Switch 1	交换机 1
Outgoing RP	出站 RP
StackWise Plus port connections	StackWise Plus 端口连接
Switch 2	交换机 2
Switch 3	交换机 3
Switch A	交换机 A
Switch B	交换机 B
RP=root port	RP=根端口

DP=designated port	DP=指定端口
BP=blocked port	BP=阻塞端口

在交换网络中，所有不需要从任何途径到达根设备的路径都会进入生成树阻塞模式。

桥 ID、设备优先级与扩展系统 ID

IEEE 802.1D 白哦准要求每台设备都有一个唯一的桥标识符（即桥 ID），这个标识符控制根桥的选择。由于在 PVST+和快速 PVST+环境中，每个 VLAN 都可以理解为一个不同的逻辑桥，因此同一台设备必须针对每个配置的 VLAN 都有一个不同的桥 ID。设备的每个 VLAN 都有一个唯一的 8 字节桥 ID。其中最重要的两个字节用于设备优先级，剩下的 6 个字节则取自于设备的 MAC 地址。

设备支持 IEEE 802.1t 生成树扩展，一些之前用于设备优先级的比特位在这个标准中用于 VLAN 标识符。这样做的结果是保留给设备的 MAC 地址更少了，而支持的 VLAN ID 范围更大的，这些都是为了保证桥 ID 的唯一性。

之前用于设备优先级的前 2 字节重新分配为了 4 比特的优先级值和等同于 VLAN ID 的 12 比特扩展系统 ID 值。

表 53：设备优先级值与扩展系统 ID

优先级位				**扩展系统 ID（设置为 VLAN ID）**								
16 位	**15 位**	**14 位**	**13 位**	**12 位**	**11 位**	**10 位**	**9 位**	**8 位**	**7 位**	**6 位**	**5 位**	**4 位**
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8

3 位	**2 位**	**1 位**
4	2	2

生成树会使用扩展系统 ID、设备优先级和分配的生成树 MAC 地址来确保每个 VLAN 的桥 ID 是唯一的。由于设备堆栈在网络其他部分看来是一台设备，因此堆栈中所有设备对于一个给定生成树来说都会使用相同的桥 ID。如果堆栈主设备发生了故障，那么堆栈成员设备就会基于新堆栈主设备的 MAC 地址来给所有运行的生成树重新计算它们各自的桥 ID。

支持扩展的系统 ID 会影响用户手动配置根设备、辅助根设备和 VLAN 的设备优先级的方式。例如，在用户修改设备优先级值时，用户可以修改设备会被选举为根设备的可能性。优先级值配置得越高，可能性就越低；反之则越高。

如果特定 VLAN 中的根设备优先级值低于 24576，那么设备就会将自己对指定 VLAN 的优先级设置为比最低设备优先级低 4096。4096 是表中最不重要的那 4 位设备优先级值。

端口优先级与路径开销

如果出现环路，生成树在选择将接口设置为转发状态时，会使用端口优先级。用户可以给自己希望首先选择的接口配置最高的优先级（也就是将优先级的数值配置为最低），给自己希望最后选择的接口配置最低的优先级（也就是将优先级的数值配置为最高）。如果所有接口的优先级值都相同，那么生成树就会让编号最低的接口进入转发状态，并且阻塞其他接口。

生成树路径开销的默认值取自于接口的媒体速率。如果出现环路，生成树就会在选择要将哪个接口置入转发状态时使用开销值。用户可以给那些希望首先选择的接口分配较低的开销值，并给那些希望最后选择的接口分配较高的开销值。如果所有接口拥有相同的开销值，那么生成树就会将接口编号最低的接口置入转发状态，并且阻塞其他接口。

如果设备是一个设备堆栈的成员，那么用户必须给希望首先选择的接口分配较低的开销值，并且给希望最后选择的接口分配较高的开销值，而不应该调整它们的端口优先级。要想了解具体信息，可以参考相关主题。

相关主题

生成树接口状态

在协议信息穿过一个交换型局域网时，就会发生转发延迟。于是，在交换型网络中，网络拓扑就可以随时随地发生变更。当接口直接从没有参与生成树拓扑的状态过渡到转发状态时，它会就会形成一个临时的数据环路。接口必须等待新拓扑信息在交换网络中传播之后，才会开始转发数据帧。这些接口必须让在旧拓扑中转发的数据帧生存时间过期。

使用生成树的设备上，每个二层接口都会处于下列状态之一：

- 阻塞：接口不会参与数据帧转发；
- 侦听：在阻塞状态之后，当生成树决定让这个接口参与数据帧转发时，这个接口经历的第一个过渡状态；
- 学习：接口准备参与数据帧转发；
- 转发：接口转发数据帧；
- 禁用：因为这个端口被关闭、没有连接链路或者没有运行生成树实例，所以不会参与生成树。

接口会按照这种方式转换状态：

- 从初始化过渡到阻塞；
- 从阻塞过渡到侦听或禁用；
- 从侦听过渡到学习或禁用；
- 从学习过渡到转发或禁用；
- 从转发到禁用

接口会在状态下进行过渡。

图 43：生成树接口状态

Power-on initialization	加电 初始化
Blocking state	阻塞状态
Listening state	侦听状态
Disabled state	禁用状态
Learning state	学习状态
Forwarding state	转发状态

在设备启动时，生成树默认就会启用，设备、VLAN 或网络中的每个接口都会经历从阻塞状态进入到侦听与学习两个过渡状态的过程。每个接口都会最终稳定在转发或阻塞状态。

当生成树算法将一个二层接口置入转发状态时，就会发生下面的过程：

1 当生成树等待协议信息将接口置入阻塞状态时，接口会处于侦听状态；

2 当生成树等待转发延迟计时器过期时，生成树就会让接口进入学习状态，并且重置转发延迟计时器；

3 在学习状态下，当设备的转发数据库学习到终端站点的位置信息时，接口还是会继续阻塞数据帧的转发；

4 当转发延迟计时器超时时，生成树就会让接口进入转发状态对数据帧的学习和转发也会启用。

阻塞状态

阻塞状态的二层接口不会参与数据帧的转发。在初始化之后，设备会通过每个设备接口发送BPDU。设备最初会像根桥那样工作，直到它与其他设备交换了 BPDU 为止。交换的过程会决定网络中的哪台设备是根设备。如果网络中只有一台设备，那就不会进行消息交换，转发延迟计时器会过期，而接口也会进入侦听状态。在设备初始化之后，接口一定会进入阻塞状态。

阻塞状态下的接口会执行下列功能：

- 丢弃这个接口接收到的数据帧；
- 丢弃从另一个接口交换过来进行转发的数据帧；
- 不学习地址；
- 接收 BPDU。

侦听状态

侦听状态是阻塞状态之后，二层接口进入的第一个状态。当生成树决定一个接口应该参与数据帧转发时，这个接口就会进入到这种状态下。

侦听状态下的接口会执行下列功能：

- 丢弃这个接口接收到的数据帧；
- 丢弃从另一个接口交换过来进行转发的数据帧；
- 不学习地址；
- 接收 BPDU。

学习状态

处于学习状态下的二层接口会准备参与数据帧转发。接口会从侦听状态进入学习状态。

学习状态下的接口会执行下列功能：

- 丢弃这个接口接收到的数据帧；
- 丢弃从另一个接口交换过来进行转发的数据帧；
- 学习地址；
- 接收 BPDU。

转发状态

处于转发状态下的二层接口会转发数据帧。接口会从学习状态进入转发状态。

转发状态下的接口会执行下列功能：

- 接收并转发这个接口接收到的数据帧；
- 转发从另一个接口交换过来的数据帧；
- 学习地址；
- 接收 BPDU。

禁用状态

处于禁用状态的二层接口并不会参与数据帧转发，也不会包含在生成树当中。处于禁用状态的接口是不执行操作的。

禁用状态下的接口会执行下列功能：

- 丢弃这个接口接收到的数据帧；
- 丢弃从另一个接口交换过来进行转发的数据帧；
- 不学习地址；
- 不接收 BPDU。

设备或端口是如何成为根设备或根端口的

如果网络中的所有设备都启用了默认的生成树设置，那么拥有最低 MAC 地址的设备就会成为根设备。

设备 A 会被选举为根设备，因为所有设备的设备优先级会被设置为默认值（32768），而设备 A 的 MAC 地址最低。不过，考虑到流量模式，转发接口的数量和链路的类型，设备 A 也许

并不是理想的根设备。用户可以增加理想设备的优先级（即降低优先级的数值），让理想的设备成为根设备，用户可以强制生成树重新进行计算，来以理想设备为根形成新的拓扑。

图 44：生成树拓扑

RP=Root Port	RP=根端口
DP=Designated Port	DP=指定端口

在网络基于默认参数计算生成树拓扑时，交换网络中源与目的终端站点的路径可能并不是最理想的路径。例如，如果将速率较高的链路连接到了一个比根端口数值高的接口，有可能就会导致根端口的变更。这样做的目的在于让速率最高的链路成为根端口。

比如，若设备 B 上的一个端口为吉比特以太网链路，而设备 B 上的另一个端口（连接的是一条 10/100 链路）却是根端口。网络流量如果穿越吉比特以太网链路效率很可能更高。用户可以将吉比特以太网端口的修改生成树端口优先级修改为一个比根端口更高的优先级（也就是修改为一个更小的数），那么这个吉比特以太网就会成为新的根端口。

生成树与冗余连接

通过生成树协议，用户可以将两个设备接口连接到另一台设备，或者连接到两台不同的设备，以此创建冗余的骨干。生成树会自动禁用一个接口，但如果另一个接口出现故障时，生成树就会重新启用这个接口。如果其中一条链路是高速链路，另一条链路是低速链路，那么禁用的永远是那条低速链路。如果速率相同，那么生成树会将端口优先级和端口 ID 相加，然后禁用那个数值较高的端口。

图 45：生成树与冗余连接

Active link	活动链路
Blocked link	阻塞链路
Workstations	工作站

用户也可以使用 EtherChannel 组在设备之间创建冗余链路。

生成树地址管理

IEEE 802.1D 指定了 17 个组播地址来由不同的桥协议使用，这 17 个组播地址取值范围是从 0x00180C2000000 到 0x0180C2000010。这些地址是静态地址，无法移除。

无论生成树协议是什么状态，堆栈中的每台设备都会接收，但不会转发去往 0x00180C2000000 到 0x0180C2000010 之间的地址。

如果启用了生成树，那么堆栈中每台设备的 CPU 都会接收到去往 0x00180C2000000 到 0x0180C2000010 的数据包。如果禁用了生成树，那么设备或堆栈中的每台设备都会向那些未知组播地址转发数据包。

加速老化以保持连接

动态地址老化的默认时间为 5 分钟，这就是全局配置命令的默认设置为 **mac address-table aging-time**。不过，重新配置生成树可能会让许多站点的位置发生变化。由于这些站点有可能无法在 5 分钟或更长时间内到达，因此用户可以加速地址老化时间，让交换机将工作站地址从地址表中删除，然后再重新学习。在生成树重新计算时，加速老化与转发延迟参数值（全局配置命令 **spanning-tree vlan** *vlan-id* **forward-time** *seconds*）相同。

由于每个 VLAN 都是一个独立的生成树实例，设备会以每个 VLAN 为单位加速老化。在一个 VLAN 上执行生成树重新配置会让在这个 VLAN 中学习到的动态地址受到加速老化的影响。其他 VLAN 中的动态地址不会受到影响，这些地址仍然服从于在设备上输入的老化时间间隔。

生成树的模式与协议

设备支持下面这些生成树模式与协议：

- PVST+：这种模式是基于 IEEE 802.1D 和 Inspur 私有扩展标准的生成树标准。PVST+会在设备的每个 VLAN 上运行，直至达到最大的支持数量，这可以确保每个 VLAN 都在网络

中获得一个无环路径。

PVST+可以给它运行的 VLAN 提供二层的负载分担。用户可以使用网络中的 VLAN 来创建不同的逻辑拓扑，以确保所有链路都得到了有效地利用，但又没有链路会过载。在一个 VLAN 中，每个 PVST+实例都有一个单独的根设备。这台根设备会将与这个 VLAN 有关的生成树信息分发给网络中的所有设备。由于每台设备都拥有了关于网络的相同信息，因此这个过程可以确保网络维护能够得到维护；

- 快速 PVST+：快速 PVST+是设备上默认的 STP 模式。这种生成树模式与 PVST+相同，只不过这种模式使用了基于 IEEE 802.1w 标准的快速收敛。为了提供快速收敛，快速 PVST+会在接收到拓扑变更消息时，以端口为单位立即删除动态学习到的 MAC 地址条目。而 PVST+则对动态学习的 MAC 地址条目使用了一个比较短的老化时间。

  （除非专门指出）快速 PVST+采用了与 PVST+相同的配置方法，用户只需要在设备上进行最简单的配置。快速 PVST+的好处在于用户可以将大量的 PVST+安装库迁移到快速 PVST+当中，而不需要掌握多生成树协议（MSTP）复杂的配置方法，也不需要重新部署网络。在快速 PVST+模式中，每个 VLAN 都会运行自己的生成树实例，直至达到最大支持的生成树数量为止；
- MSTP：这种模式是基于 IEEE 802.1s 标准的生成树标准。用户可以将多个 VLAN 映射到同一个生成树实例当中，这可以减少生成树实例的数量，让设备不必为大量 VLAN 支持生成树。MSTP 采用的是 RSTP（基于 IEEE 802.1w）的做法，即通过减少转发延迟及快速将根端口与指定端口快速过渡到转发状态下，来提升生成树的收敛速率。在设备堆栈中，交叉堆栈快速转换（Cross stack rapid transition，CSRT）特性会执行与 RSTP 相同的功能。没有 RSTP 或 CSRT 就无法运行 MSTP。

支持的生成树实例

在 PVST+或快速 PVST+模式下，设备或设备堆栈支持最多 128 个生成树实例。

在 MSTP 模式下，设备或设备堆栈支持 65 个 MST 实例。每个 MST 中可以映射的 VLAN 数量是没有限制的。

生成树的互操作性与向后兼容

在混合使用 MSTP 和 PVST+的网络中，公共生成树（CST）的根必须处于 MST 骨干当中，而 PVST+设备无法连接到多个 MST 域。

当一个网络同时包含了运行快速 PVST+的设备和运行 PVST+的设备时，我们推荐将快速 PVST+设备与 PVST+设备配置为不同的生成树实例。在快速 PVST+生成树实例中，根设备必须为快速 PVST+设备。在 PVST+实例中，根设备必须为 PVST+设备。PVST+设备应该部署在网络的边缘。

所有堆栈成员都会运行相同版本的生成树（皆为 PVST+、皆为快速 PVST+或皆为 MSTP）。

表 54：PVST+、MSTP 和快速 PVST+的互操作性与兼容性

	PVST+	**MSTP**	**快速 PVST+**
PVST+	是	是（但有限制）	是（回退为 PVST+）
MSTP	是（但有限制）	是	是（回退为 PVST+）
快速 PVST+	是（回退为 PVST+）	是（回退为 PVST+）	是

STP 与 IEEE 802.1Q Trunk

对于 VLAN trunk 的 IEEE 802.1Q 标准对于网络的生成树战略施加了一些限制条件。这种标准要求给 trunk 支持的*所有* VLAN 只能运行一个生成树实例。但在一个由 Inspur 设备通过 IEEE 802.1Q trunk 相互连接而组成的网络中，这些设备还是会给 trunk 支持的每个 VLAN 各自维护一个生成树实例。

在将一台 Inspur 设备通过 IEEE 802.1Q trunk 连接到一台非 Inspur 设备时，Inspur 设备会使用

PVST+来提供生成树的互操作性。如果启用了快速 PVST+，那么设备就会使用快速 PVST+，而不使用 PVST+。设备会将 IEEE 802.1Q VLAN 的生成树实例，与非 Inspur IEEE 802.1Q 设备的生成树实例结合起来。

不过，被一个由非 Inspur IEEE 802.1Q 设备所组成的网络云相互隔开的 Inspur 设备会维护所有 PVST+或快速 PVST+信息，而隔开 Inspur 设备的非 Inspur IEEE 802.1Q 云会被视为设备之间的一条 trunk 链路。

IEEE 802.1Q trunk 上会自动启用快速 PVST+，这里不需要用户进行任何配置。Access 端口及 ISL（交换机间链路） trunk 端口上的外部生成树的操作不会受到 PVST+的影响。

VLAN 桥生成树

Inspur VLAN 桥生成树是与后退桥接特性（桥组）一起使用的，后者会在两个或多个 VLAN 桥域或路由端口之间转发非 IP 协议（如 DECnet）的流量。VLAN 桥生成树可以让桥组在每个 VLAN 生成树之上，再建立一个生成树，防止 VLAN 之间有多条连接进而形成环路。这项技术也可以防止各个被桥接的 VLAN 的生成树不会坍塌为一棵生成树。

要支持 VLAN 桥生成树，需要增加一些生成树计时器。要使用后退桥接特性，用户必须在设备上启用 IP Services 特性集。

生成树与设备堆栈

当设备堆栈工作在 PVST+或快速 PVST+模式下时：

- 设备堆栈在网络其他部分看来是一台设备，因此堆栈中所有设备对于一个给定生成树来说都会使用相同的桥 ID。其桥 ID 取自于主用交换机的 MAC 地址；
- 当一台新的设备加入堆栈时，它会将自己的桥 ID 设置为主用交换机的桥 ID。如果这台新添加的设备 ID 值最低，且所有堆栈成员的根路径开销相同，那么新添加的这台设备就会成为堆栈的根；
- 当一台堆栈成员离开堆栈时，生成树会在堆栈中（有可能也会在堆栈外）重新收敛。其余的堆栈成员设备中，拥有最低堆栈端口 ID 的设备就会成为堆栈的根；
- 如果堆栈外部的相邻设备出现故障或者掉电，网络就会执行常规的生成树处理。网络有可能会因为活动拓扑中丢失了一台设备而重新收敛；
- 如果堆栈外部添加了一台新的设备，网络就会执行常规的生成树处理。网络有可能会因为活动拓扑中增加了一台设备而重新收敛。

默认的生成树配置

表 55：默认的生成树配置

特性	**默认设置**
启用状态	在 VLAN 1 启用
生成树模式	快速 PVST+（PVST+与 MSTP）
设备优先级	32768
生成树端口优先级（可以基于接口进行配置）	128
生成树端口开销（可以基于接口进行配置）	1000Mb/s：4 100Mb/s：19 10Mb/s：100
生成树 VLAN 端口优先级（可以基于 VLAN 进行配置）	128
生成树 VLAN 端口开销（基于 VLAN 进行配置）	1000Mb/s：4

	100Mb/s：19 10Mb/s：100
生成树计时器	Hello 时间：2 秒 转发延迟时间：15 秒 最大老化时间：20 秒 传输抑制树：6 BPDU

注释： 从 Inspur INOS 15.2(4)E 版开始，默认 STP 模式为快速 PVST+。

如何配置生成树特性

修改生成树模式（CLI）

交换机支持三种生成树模式：每 VLAN 生成树加（PVST+）、快速 PVST+或多生成熟协议（MSTP）。在默认情况下，设备会运行快速 PVST+协议。

如果用户希望启用一种与默认模式不同的模式，需要执行下面的流程。

总步骤

1. enable

2. configure terminal

3. spanning-tree mode {pvst | mst | rapid-pvst}

4. interface *interface-id*

5. spanning-tree link-type point-to-point

6. end

7. clear spanning-tree detected-protocols

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**spanning-tree mode {pvst | mst | rapid-pvst}** 示例： `Device(config)#` **spanning-tree mode pvst**	配置生成树模式。所有堆栈成员运行生成树的同一个版本。 • 选择 **pvst** 可以启用 PVST+； • 选择 **mst** 可以启用 MSTP； • 选择 **rapid-pvst** 可以启用快速 PVST+
步骤 4	**interface** *interface-id* 示例：	指定要配置的接口，进入接口配置模式。有效接口包括物理端口、VLAN 和 port channels。VLAN ID 的取值范围是从 1 到 4094，port-channel 的取值范围

	`Device(config)#` **interface gigabitethernet 1/0/1**	是从 1 到 48
步骤 5	**spanning-tree link-type point-to-point** 示例： `Device(config-if)#` **spanning-tree link-type point-to-point**	将这个端口的链路类型设置为点到点。 如果将这个端口（本地端口）通过一条点到点链路连接到一个远端端口，而这个本地端口成为了指定端口的话，那么设备就会与远端端口进行协商，并且快速将本地端口修改为转发状态
步骤 6	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 7	**show ip igmp interface** [*interface-id*] 示例： `Device#` **show ip igmp interface**	设备上的任何端口与一台传统的 IEEE 802.1D 相连，这条命令都会在整台设备上重新启动协议迁移进程。 如果指定设备检测到这台设备运行的是快速 PVST+，那么这一步就是可选的操作

禁用生成树（CLI）

生成树默认会在 VLAN 1 和所有新创建的 VLAN 上启用，直至达到了生成树的限制数量为止。只有在用户十分确定网络中没有环路时，才可以禁用生成树。

注意： 如果用户禁用了生成树而网络中又仍然有环路，那么网络中的过量流量和永无休止的数据包复制操作会严重影响网络的性能。

这项操作是可选的。

总步骤

1. enable

2. configure terminal

3. no spanning-tree vlan *vlan-id*

4. end

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**no spanning-tree vlan** *vlan-id*	*vlan-id* 部分的取值范围是从 1 到 4094

	示例： `Device(config)#` **no spanning-tree vlan 300**	
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置根设备（CLI）

用户要在指定 VLAN 中将一台设备配置为根，可以使用全局配置命令 **spanning-tree vlan** *vlan-id* **root** 来将设备优先级修改为一个远远低于默认值（32768）的数值。在输入这条命令时，软件会校验每个 VLAN 中根设备的设备优先级。由于支持扩展系统 ID，因此如果 24576 这个值可以让设备成为指定 VLAN 的根，那么设备会将自己在指定 VLAN 中的优先级设置为 24576。用户可以使用关键字 **diameter** 来设置二层网络的直径（即两台终端工作在在二层网络中相隔的设备最大跳数）。在用户设置网络半径的时候，设备会自动设置优化的 hello 时间、转发延迟时间和这个网络半径下的最大老化时间，这些参数可以显著减少收敛时间。用户可以使用关键字 **hello** 来覆盖自动计算出来的 hello 时间。

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. spanning-tree vlan *vlan-id* **root primary** [**diameter** *net-diameter*

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**spanning-tree vlan** *vlan-id* **root primary** [**diameter** *net-diameter* 示例： `Device(config)#` **spanning-tree vlan 20-24 root primary diameter 4**	将一台设备配置为指定 VLAN 的根。 在 *vlan-id* 部分，用户可以通过 VLAN ID 值来输入一个 VLAN，可以用连字符输入一个 VLAN 范围，也可以用逗号相隔输入一系列的 VLAN。VLAN 的取值范围是从 1 到 4094。 • （可选）在 **diameter** *net-diameter* 部分，设置两台终端工作站之间相隔的最大设备数量。取值范围是 2 到 7
步骤 4	**end**	返回特权 EXEC 模式

	示例： `Device(config)# end`	

接下来做什么？

在将设备配置为根设备之后，我们推荐用户不要手动通过全局配置命令 **spanning-tree vlan** *vlan-id* **hello-time**、**spanning-tree vlan** *vlan-id* **forward-time** 和 **spanning-tree vlan** *vlan-id* **max-age** 来配置 hello 时间、转发延迟时间和最大老化时间。

配置辅助根设备（CLI）

在将一台设备配置为辅助根时，设备优先级会从默认值（32768）修改为 28672。如果这个 VLAN 的主用根发生了故障，那么设备使用这个优先级就更有可能会成为这个 VLAN 的根设备。这一点的前提是其他网络设备使用的都是默认的设备优先级 32768，因此这些设备就很难成为根设备。

用户可以在多台设备上执行这条命令，来将多台设备配置为备份根设备。用户也可以使用配置主用根设备时使用的命令 **spanning-tree vlan** *vlan-id* **root primary** 来设置辅助根设备的网络直径与 hello 时间值。

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. spanning-tree vlan *vlan-id* **root secondary** [**diameter** *net-diameter*

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**spanning-tree vlan** *vlan-id* **root secondary** [**diameter** *net-diameter* 示例： `Device(config)#` **spanning-tree vlan 20-24 root secondary diameter 4**	将一台设备配置为指定 VLAN 的辅助根。 • 在 *vlan-id* 部分，用户可以通过 VLAN ID 值来输入一个 VLAN，可以用连字符输入一个 VLAN 范围，也可以用逗号相隔输入一系列的 VLAN。VLAN 的取值范围是从 1 到 4094. • （可选）在 **diameter** *net-diameter* 部分，设置两台终端工作站之间相隔的最大设备数量。取值范围是 2 到 7。 应该在这里给设备配置与主用根设备相同的网络直径

步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置端口优先级（CLI）

注释： 如果用户的设备是设备堆栈的成员，那就必须使用接口配置命令 **spanning-tree** [**vlan** *vlan-id*] **cost** *cost*（而不是接口配置命令 **spanning-tree** [**vlan** *vlan-id*] **port-priority** *priority*）来选择将一个接口置入转发状态。用户可以给自己希望首先选择的接口配置一个较低的开销值，而给自己希望之后选择的接口配置一个较高的开销值。

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. spanning-tree port-priority *priority*

5. spanning-tree vlan *vlan-id* **port-priority** *priority*

6. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)# interface gigabitethernet1/0/2`	指定要配置的接口，进入接口配置模式。 有效接口包括物理端口 port channel 逻辑接口（**port-channel** *port-channel-number*）
步骤 4	**spanning-tree port-priority** *priority* 示例： `Device(config-if)# spanning-tree port-priority 0`	给一个接口配置端口优先级。 在 *priority* 部分，取值范围是从 0 到 240，增量为 16，默认值是 128。有效的取值包括 0、16、32、48、64、80、96、112、128、144、160、176、192、208、224 和 240。配置其他值系统都会拒绝。数值越低，优先级就越高
步骤 5	**spanning-tree vlan** *vlan-id* **port-priority** *priority*	给一个 VLAN 配置端口优先级。 • 在 *vlan-id* 部分，用户可以通过 VLAN ID 值来输入一个 VLAN，可以用连字符输入一个 VLAN 范

	示例： `Device(config-if)#` **spanning-tree vlan 20-25 port-priority 0**	围，也可以用逗号相隔输入一系列的 VLAN。VLAN 的取值范围是从 1 到 4094. • 在 *priority* 部分，取值范围是从 0 到 240，增量为 16，默认值是 128。有效的取值包括 0、16、32、48、64、80、96、112、128、144、160、176、192、208、224 和 240。配置其他值系统都会拒绝。数值越低，优先级就越高
步骤 6	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置路径开销（CLI）

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. spanning-tree cost *cost*

5. spanning-tree vlan *vlan-id* **cost** *cost*

6. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet1/0/1**	指定要配置的接口，进入接口配置模式。 有效接口包括物理端口 port channel 逻辑接口（**port-channel** *port-channel-number*）
步骤 4	**spanning-tree cost** *cost* 示例： `Device(config-if)#` **spanning-tree cost 250**	给一个接口配置开销。 • 如果出现环路，生成树在选择要将哪个接口置入转发状态时就会使用路径开销进行判断。路径开销越低表示传输速率越高。 • 在 *cost* 部分，取值范围是从 1 到 200000000，这个值取自于接口媒体的速率

步骤 5	**spanning-tree vlan** *vlan-id* **cost** *cost* 示例： `Device(config-if)#` **spanning-tree vlan 10,12-15,20 cost 300**	给一个 VLAN 配置开销。 如果出现环路，生成树在选择要将哪个接口置入转发状态时就会使用路径开销进行判断。路径开销越低表示传输速率越高。 • 在 *vlan-id* 部分，用户可以通过 VLAN ID 值来输入一个 VLAN，可以用连字符输入一个 VLAN 范围，也可以用逗号相隔输入一系列的 VLAN。VLAN 的取值范围是从 1 到 4094. • 在 *cost* 部分，取值范围是从 1 到 200000000，这个值取自于接口媒体的速率
步骤 6	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

特权 EXEC 命令 **show spanning-tree interface** *interface-id* 只会显示那些链路处于 up 状态的端口信息。否则，用户也可以使用特权 EXEC 命令 **show running-config** 来确认自己所作的配置。

配置一个 VLAN 的设备优先级（CLI）

用户可以配置设备的优先级，让一台独立设备或一台堆栈中的设备更有可能被选为根设备。

注释： 在使用这条命令时务请小心。在大多数情况下，我们推荐用户使用全局配置命令 **spanning-tree vlan** *vlan-id* **root primary**和**spanning-tree vlan** *vlan-id* **root secondary**来修改设备优先级。

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. spanning-tree vlan *vlan-id* **priority** *priority*

4. end

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**spanning-tree vlan** *vlan-id* **priority** *priority* 示例：	配置一个 VLAN 的设备优先级。 • 在 *vlan-id* 部分，用户可以通过 VLAN ID 值来输入一个 VLAN，可以用连字符输入一个 VLAN 范围，也可以用逗号相隔输入一系列的 VLAN。

	Device(config)# **spanning-tree vlan 20 priority 8192**	VLAN 的取值范围是从 1 到 4094. • 在 *priority* 部分，取值范围是从 0 到 61440，增量为 4096，默认值是 32768。数值越低，设备越有可能被选为根设备。 有效的值包括 4096、8192、12288、16384、20480、24576、28672、32768、36864、40960、45056、49152、53248、57344 和 61440。配置其他值系统都会拒绝。
步骤 4	**end** 示例： Device(config)# end	返回特权 EXEC 模式

配置 Hello 时间（CLI）

hello 时间是根设备生成和发送配置消息的时间间隔。
这个流程是可选的。

总步骤

1. enable

2. spanning-tree vlan *vlan-id* **hello-time** *seconds*

3. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： Device> **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**spanning-tree vlan** *vlan-id* **hello-time** *seconds* 示例： Device(config)# **spanning-tree vlan 20-24 hello-time 3**	配置一个 VLAN 的 hello 时间。hello 时间是根设备生成和发送配置消息的时间间隔。 • 在 *vlan-id* 部分，用户可以通过 VLAN ID 值来输入一个 VLAN，可以用连字符输入一个 VLAN 范围，也可以用逗号相隔输入一系列的 VLAN。VLAN 的取值范围是从 1 到 4094. • 在 *seconds* 部分，取值范围是从 1 到 10；默认值为 2
步骤 3	**end** 示例： Device(config-if)# **end**	返回特权 EXEC 模式

给一个 VLAN 配置转发延迟时间（CLI）

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. spanning-tree vlan *vlan-id* **forward-time** *seconds*

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**spanning-tree vlan** *vlan-id* **forward-time** *seconds* 示例： `Device(config)#` **spanning-tree vlan 20,25 forward-time 18**	配置一个 VLAN 的转发时间。转发延迟是接口在将自己的生成树学习和侦听状态过渡到转发状态之前，等待的秒数。 • 在 *vlan-id* 部分，用户可以通过 VLAN ID 值来输入一个 VLAN，可以用连字符输入一个 VLAN 范围，也可以用逗号相隔输入一系列的 VLAN。VLAN 的取值范围是从 1 到 4094. • 在 *seconds* 部分，取值范围是从 4 到 30，默认值是 15
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

给一个 VLAN 配置最大老化时间

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. spanning-tree vlan *vlan-id* **max-age** *seconds*

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码

步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**spanning-tree vlan** *vlan-id* **max-age** *seconds* 示例： `Device(config)# spanning-tree vlan 20 max-age 30`	配置一个 VLAN 的最大老化时间。最大老化时间是指设备从没有接收到生成树配置消息开始，会等待多久才会开始执行重新配置。 • 在 *vlan-id* 部分，用户可以通过 VLAN ID 值来输入一个 VLAN，可以用连字符输入一个 VLAN 范围，也可以用逗号相隔输入一系列的 VLAN。VLAN 的取值范围是从 1 到 4094； • 在 *seconds* 部分，取值范围是从 6 到 40，默认值是 20
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置传输抑制计时（CLI）

用户可以通过修改传输抑制计时值来配置 BPDU 突发值。

注释： 将这个参数修改为一个较高的数值会严重影响 CPU 的使用率，特别是在快速 PVST+ 模式下。降低这个参数值则会在某些情况下延迟收敛时间。我们推荐用户维持默认的设置。

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. spanning-tree transmit hold-count *value*

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**spanning-tree transmit hold-count** *value* 示例：	配置在暂停 1 秒之前可以发送的 BPDU 数量。 • 在 *value* 部分，取值范围是从 1 到 20，默认值为 6

	Device(config)# **spanning-tree transmit hold-count 6**	
步骤 4	**end** 示例： Device(config)# end	返回特权 EXEC 模式

监控生成树的状态

表 56：查看生成树状态的命令

show spanning-tree active	仅显示活动接口的生成树信息
show spanning-tree detail	显示具体的接口信息
show spanning-tree vlan *vlan-id*	显示特定 VLAN 的生成树信息
show spanning-tree interface *interface-id*	显示特定接口的生成树信息
show spanning-tree interface *interface-id* **portfast**	显示特定接口的生成树 portfast 信息
show spanning-tree summary [**totals**]	显示接口状态的汇总信息，或者显示 STP 状态部分的总行

其他关于生成树协议的参考资料

相关文档

相关主题	**文档名**
生成树协议的命令	《LAN 交换命令参考手册，Inspur INOSXE3SE 版（Inspur 6650 交换机）》

错误消息解码器

描述	**链接**
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com/icnt

标准与 RFC

标准/RFC	**标题**
无	——

技术助手

描述	**链接**
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚	http://www.icntnetworks.com/icnt

合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	

STP 的特性信息

版本	修改
Inspur INOS 11.3.1	引入该特性

配置多生成树协议

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com/go/cfn。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

MSTP 的前提条件

- 对于同一个多生成树（MSTP）域中的两台或多台设备，它们必须拥有相同的 VLAN 与实例映射关系、相同的修订版本号和相同的名称；
- 对于同一个 MSTP 域中的两个或多个堆栈交换机来说，它们必须拥有相同的 VLAN 与实例映射关系、相同的修订版本号和相同的名称；
- 要想网络能够跨越冗余路径执行负载分担，那么所有 VLAN 与实例的映射关系就必须相互匹配；否则，所有流量就都会通过同一条链路。用户可以通过手动配置路径开销，来跨越堆栈中设备实现负载分担；
- 要想在一个每 VLAN 生成树加（PVST+）和一个 MST 云之间实现负载分担，或者在一个快速 PVST+和一个 MST 云之间实现负载分担，所有 MST 边界端口都必须进行转发。MST

边界端口进行转发，同时 MST 云的内部生成树（IST）主设备是公共生成树（CST）的根。如果 MST 云由多个 MST 域组成，那么其中一个 MST 域必须包含 CST 根，而所有其他 MST 区域必须有一条比通过 PVST+云或快速 PVST+云更优的路径，可以去往 MST 云的根。用户可能需要手动配置云中的设备。

MSTP 的限制条件

- 用户不能混合使用 Inspur 3850 和 Inspur 6650 交换来建立堆栈；
- 设备堆栈支持最多 65 个 MST 实例，但可以映射到一个 MST 实例中的 VLAN 数量是不受限制的；
- 支持 PVST+、快速 PVST+和 MSTP，但一次只能使用一个版本（例如，所有 VLAN 均运行 PVST+、所有 VLAN 均运行快速 PVST+或所有 VLAN 均运行 MSTP）；
- 不支持通过 VLAN 中继协议（VTP）传播 MST 的配置。但用户可以使用命令行界面（CLI）或者通过 SNMP（简单网络管理协议），在 MST 域的各个设备上手动配置 MST 的配置（域名称、修订版本号和 VLAN 与实例的映射）；
- 不推荐将网络分为一个大的域。但如果这种情况无法避免，我们推荐用户将交换型以太网分入通过路由器或非二层设备连接的小型局域网当中；
- 一个域中可以拥有一个或多个 MST 配置相同的成员；每个成员都必须能够处理快速生成树协议（RSTP）桥协议数据单元（BPDU）。一个网络中的 MST 域没有数量限制，但每个域只能支持最多 65 个生成树实例。用户每次只能将一个 VLAN 分配给一个生成树实例；
- 在将一台设备配置为根设备之后，我们推荐用户不要通过全局配置命令 **spanning-tree mst hello-time**、**spanning-tree mst forward-time** 和 **spanning-tree mst max-age** 手动配置 hello 时间、转发延迟时间、最大老化时间。

表 57：PVST+、MSTP 和快速 PVST+的互操作性域兼容性

	PVST+	**MSTP**	**快速 PVST+**
PVST+	是	是（但有限制）	是（回退为 PVST+）
MSTP	是（但有限制）	是	是（回退为 PVST+）
快速 PVST+	是（回退为 PVST+）	是（回退为 PVST+）	是

关于 MSTP 的信息

MSTP 的配置

MSTP 使用 RSTP 来实现快速收敛，这种技术可以将多个 VLAN 进行分组，并且映射到同一个生成树实例当中，减少支持大量 VLAN 所需的生成树实例数量。MSTP 可以给数据流量提供多条转发路径，以此来实现负载分担，减少支持大量 VLAN 所需的生成树实例数量。这项技术可以提升网络的容错性，因此一个实例（转发路径）出现了故障并不会影响其他实例（转发路径）正常工作。

注释： 多生成树（MST）是基于 IEEE 802.1s 标准实施的。

MSTP 最常用的初始部署方式是部署在二层交换网络的骨干和分布层。这种部署方式可以提供能够满足服务运营商网络需求的高可用性网络。

当设备工作在 MST 模式下时，（基于 IEEE 802.1w 的）RSTP 会自动启用。RSTP 通过显式握手的方式消除了 IEEE 802.1D 标准的转发延迟，让根端口和指定端口可以迅速过渡到转发状态，因此实现生成树的快速收敛。
MSTP 和 RSTP 都提升了生成树的操作水平，同时可以对基于（原始的）IEEE 802.1D 的生成树，以及 Inspur 私有的多实例生成树（MISTP）和 Inspur PVST+与快速每 VLAN 生成树加（快速 PVST+）实现向后兼容。
对于网络的其余部分而言，设备堆栈就是一个生成树节点，所有堆栈成员都会使用相同的设备 ID。

MSTP 配置指南

在使用全局配置命令 **spanning-tree mode mst** 启用 MST 时，RSTP 也会自动启用；
关于 UplinkFast、BackboneFast 和交叉堆栈 UplinkFast 的配置指南，请参见相关主题中提到的相关内容；
当设备工作在 MST 模式下时，它会使用长路径开销计算法（32 位）来计算路径开销值。通过长路径开销计算法，设备支持下面的路径开销值：

速率	**路径开销值**
10Mb/s	2000000
100Mb/s	200000
1Gb/s	20000
10Gb/s	2000
100Gb/s	200

根交换机

设备会给映射的 VLAN 组维护一个生成树实例。每个实例都会关联一个由设备优先级和设备 MAC 地址组成的设备 ID。对于一组 VLAN 来说，设备 ID 最低的设备会成为根设备。
在将一台设备配置为根设备时，用户需要将这台设备的优先级从默认值（32768）修改为一个明显更低的数值，这样设备才能成为指定生成树实例的根设备。在输入这条命令的时候，设备会校验根设备的设备优先级。由于支持扩展系统 ID，因此如果 24576 这个值可以让设备成为指定 VLAN 的根，那么设备会将自己在指定 VLAN 中的优先级设置为 24576。
如果指定实例的根设备优先级低于 24576，那么设备就会讲自己的优先级设置得比最低设备优先级低 4096。（4096 是设备优先级值中最低 4 位的值，要想了解详细信息，可以在相关主题中选择“桥 ID、设备优先级域扩展系统 ID”的链接）
如果网络中由一些支持扩展系统 ID 的设备，和一些不支持扩展系统 ID 的设备组成，那么包含扩展系统 ID 的设备就不太可能成为根设备。每当 VLAN 编号大于运行老板系统的设备的优先级值，扩展系统 ID 会增加设备的优先级值。
对每个生成树实例来说，根设备都应该是骨干设备或分布层设备。不要将接入层设备配置为生成树的主用根。
用户可以使用关键字 **diameter**（仅可用于 MST 实例 0）来设置二层网络的半径（也就是二层网络中任意两台终端设备之间的最大设备跳数）。在用户设置网络半径的时候，设备会自动设置优化的 hello 时间、转发延迟时间和这个网络半径下的最大老化时间，这些参数可以显著减少收敛时间。用户可以使用关键字 **hello** 来覆盖自动计算出来的 hello 时间。

多生成树域

要想让交换机参与多生成树（MST）实例，用户必须给所有交换机上配置相同的 MST 配置信息。多台拥有相同 MST 配置的交换机共同组成了一个 MST 域。

MST 的配置会控制各个设备属于哪个 MST 域。配置的内容包括域名称，修订版本号和 VLAN 与实例的分配映射。用户需要在设备上设置 MST 域的配置。用户可以将多个 VLAN 映射到一个 MST 实例，并且设置域名称、设置修订版本号。要想了解详细信息和示例，可以选择相关主题中的“设置 MST 域的配置与启用 MSTP”链接。

一个域中可以有多台拥有相同 MST 配置的成员。每个成员都必须能够处理 RSTP 桥协议数据单元（BPDU）。一个网络中的 MST 域没有数量限制，但每个域只能支持最多 65 个生成树实例。实例可以使用从 0 到 4094 之间的数字进行标识。用户每次只能将一个 VLAN 分配给一个生成树实例；

IST、CIST 和 CST

在 PVST+和快速 PVST+中，每个生成树实例都是独立的。MSTP 则与此不同，它会建立和维护两类生成树：

- 一棵内部生成树（IST），即运行在 MST 域中的生成树。

  在每个 MST 域中，MSTP 都会维护多个生成树实例。实例 0 是每个域中的一个特殊实例，成为内部生成树（IST）。所有其他 MST 实例的编号则从 1 到 4094。

  IST 是唯一会发送和接收 BPDU 的生成树实例。所有其他生成树实例信息都包含在 M 记录中，而 M 记录是封装在 MSTP BPDU 中的。由于 MSTP BPDU 会携带关于所有实例的信息，因此，为了支持多生成树实例而需要处理的 BPDU 数量可以显著减少。

  同一个域中的所有 MST 实例都会共享同一个协议计时器，但每个 MST 实例都会包含自己的拓扑参数，如根设备 ID、根路径开销等等。在默认情况下，所有 VLAN 都会分配给 IST。

  MST 实例只具有区域本地意义。例如，域 A 中的 MST 实例 1 是与域 B 中的 MST 实例 1 相独立的，哪怕域 A 与域 B 相互连接也是如此。
- 一棵公共和内部生成树（CIST）。CIST 是每个 MST 域中的一系列 IST，和连接 MST 域与单个生成树的公共生成树（CST）。

  一个域中计算出来的生成树是包含整个交换域的 CST 的子树。CIST 是支持 IEEE 802.1w、IEEE 802.1s 和 IEEE 802.1D 标准的交换机共同运行的生成树算法所形成的树。一个 MST 域中的 CIST 与一个域外的 CST 相同。

MST 域内的操作

IST 连接了一个域内的所有 MSTP 交换机。当 IST 收敛时，IST 的根就会成为 CIST 的域跟（在 IEEE 802.1s 标准实现之前，称为 IST master）。这是域内拥有最低设备 ID 和去往 CIST 根最短路径开销的设备。如果网络中只有一个域的话，那么 CIST 域根也就是 CIST 的根。如果 CIST 根在域外，那么在域边界的一台 MSTP 交换机就会被选为这个 CIST 的域根。

当 MSTP 设备启动时，它会通过发送 BPDU 来声称自己是 CIST 的根和 CIST 的域根，同时将去往 CIST 根和 CIST 域根的路径开销设置为 0。设备也会启动自己所有的 MST 实例，并且声称自己是所有这些实例的根。如果设备接收到了比当前给这个端口存在的根信息更优的 MST 根信息（比如更低的设备 ID、更低的路径开销等等），它就会放弃自己作为 CIST 域根的身份。在启动过程中，一个域中可能还有很多子域，每个子域都有自己的 CIST 域根。当交换机接

收到较优的 IST 信息时，它们会离开自己过去的子域，并且加入包含了真正 CIST 域根的新子域。所有子域都会收缩，除了包含 CIST 域根的那个子域之外。

为了能够实现正常的操作，MST 域中的所有交换机必须都拥有相同的 CIST 域根。因此，域中任何两台交换机都只会在它们收敛到一个公共 CIST 域根时，针对一个实例同步它们的端口角色。

MST 域间的操作

如果网络中有多个域或传统 IEEE 802.1D 设备，那么 MSTP 就会建立并维护 CST，其中包括所有 MST 区域和网络中的所有传统 STP 设备。MST 实例会结合域边界的 IST 向结合，成为 CST。IST 连接了一个域内的所有 MSTP 交换机，并且作为包含整个交换域的 CIST 中的一个子树。这棵子树的根就是 CIST 域跟。MST 域会成为域 STP 设备和 MST 域相邻的一台虚拟设备。

一旦 CST 实例接受并发送 BPDU，MST 实例就会将它们的生成树信息添加到 BPDU 当中，来与相邻的设备进行通信，并且计算最终的生成树拓扑。正因如此，与 BPDU 传输有关的生成树参数（例如 hello 时间、转发时间、最大老化时间和最大跳数）只在 CST 实例上进行了配置，但却会影响所有 MST 实例。与生成树拓扑有关的参数（如设备优先级、端口 VLAN 开销和端口 VLAN 优先级）可以同时在 CST 实例和 MST 实例上进行配置。

MSTP 设备会使用第 3 版 RSTP BPDU 或 IEEE 802.1D STP BPDU 来域传统 IEEE 802.1D 设备进行通信。MST 设备会使用 MSTP BPDU 来与 MSTP 设备进行通信。

IEEE 802.1s 术语

在 Inspur 预标准实施方案中的一些 MST 命名方式已经进行了调整，以便定义一些*内部*或*区域*参数。这些内部参数只在一个 MST 域中有意义，而外部参数则与整个网络相关。由于 CIST 是唯一一个扩展到整个网络中的生成树实例，因此只有 CIST 参数需要用到外部（而不是内部或区域）术语。

- CIST 根是 CIST（唯一扩展到整个网络中的实例）的根设备；
- CIST 外部根路径的开销是去往 CIST 根的开销。在一个 MST 域中，开销是没有变化的。切记，对于 CIST 来说，一个 MST 域就像一台设备一样。CIST 外部根路径开销是在这些虚拟设备和不属于任何域的设备之间计算出来的根路径开销；
- CIST 域根在预标准实施方案中称为 IST master。如果 CIST 根在域中，那么 CIST 域根就是 CIST 的根。否则，CIST 域根就是距离域的 CIST 根最近的设备。CIST 域根会充当 IST 的根设备；
- CIST 内部根路径开销是去往一个域的 CIST 域根的开销。这个开销值只与 IST（实例 0）有关。

表 58：预标准与标准术语

IEEE 标准	**Inspur 预标准**	**Inspur 标准**
CIST 的域根	IST master	CIST 的域根
CIST 内部根的路径开销	IST master 的路径开销	CIST 的内部路径开销
CIST 外部根的路径开销	根的路径开销	根的路径开销
MSTI 的域根	实例的根	实例的根
MSTI 内部根的路径开销	根的路径开销	根的路径开销

MST 域的图例

这张图显示了 3 个 MST 域，和一台传统的 IEEE 802.1D 设备（D）。域 1（A）的 CIST 域跟也是 CIST 根。域 2（B）的 CIST 域根和域 3（C）的 CIST 域根分别是它们在 CIST 中对应子树的

根。RSTP 会在所有域中运行。

图 46：MST 域、CIST Master 和 CST 根

IST master and CST root	IST master 与 CST 的根
MST Region 1	MST 域 1
Legacy IEEE 802.1D	传统 IEEE 802.1D
MST Region 2	MST 域 2
MST Region 3	MST 域 3

跳数

IST 和 MST 实例不会使用配置 BPDU 中的消息老化和最大老化信息来计算生成树拓扑。它们使用的是去往根的路径开销，和一种类似于 IP 生存时间（TTL）机制的跳数机制。

用户可以使用全局配置命令 **spanning-tree mst max-hops** 来配置域内的最大跳数，并且将其应用于 IST 和这个域中的所有 MST 实例。跳数会得到与消息老化信息（触发重新配置）相同的结果。实例的根设备会始终以开销值 0 发送 BPDU（或 M 记录），并且将跳数设置为最大值。当一台设备接收到这个 BPDU，它就会把接收到的消息跳数减 1，然后将这个值作为它在这个 BPDU 中生成的剩余跳数。当跳数值为 0 时，设备就会丢弃这个 BPDU，然后将端口保存的信息老化。

在整个域中，BPDU RSTP 部分中的消息老化与最大老化信息都会保持不变，域边界的指定端口也会传播相同的值。

边界端口

在 Inspur 预标准的实施方案中，边界端口会将一个 MST 域连接到一个运行 RSTP 的生成树域，一个运行 PVST+或快速 PVST+的生成树域，或者另一个采用了不同 MST 配置的 MST 域。边界端口也会连接一个局域网，这个局域网中的指定路由器要么是一台生成树设备，要么是一台包含不同 MST 配置的设备。

在 IEEE 802.1s 标准中并没有关于边界端口的定义。IEEE 802.1Q-2002 标准定义了一个端口可以接收到的两类消息：

- （来自同一个域的）内部消息
- （来自另一个域的）外部消息

当消息是内部消息时，这个消息只能通过 CIST 接收到。如果 CIST 的角色为根端口或替代端口，或者如果外部 BPDU 是一个拓扑变更，这有可能会给 MST 实例构成影响。

MST 域中包含设备和局域网。网段属于指定端口的域。因此，与指定端口所在网段处于不同域中的端口就是边界端口。根据这种定义，域的两个内部端口可以通过属于不同域的那个端口共享一个网段，因此一个端口也就有可能同时接收到内部消息和外部消息。

Inspur 预标准实施方案的一大变化在于，CIST 域根设备 ID 字段现在被插入到了 RSTP 或传统 IEEE 802.1Q 设备标记发送方设备 ID 的地方。整个域会向虚拟设备一样执行操作，它会连续向邻居设备发送发送方设备 ID。在这个示例中，设备 C 会接收到带有同一个一致发送发设备 ID 的 BPDU，无论 A 或 B 是不是这个网段的指定设备。

IEEE 802.1s 的实施

在 Inspur 对 IEEE MST 标准的实施方案中，包含了需要满足这一标准的特性，以及一些尚未结合到已发布标准中的必备预标准功能。

端口角色的命名变化

边界角色已经没有再重现在最终的 MST 标准中，但这种边界的概念在 Inspur 的实施方案中得到了保留了。但是，在域边界的 MST 实例端口可能不会按照 CIST 端口的状态操作。当前存在两种边界角色：

- 边界端口是 CIST 域根的根端口——当 CIST 实例端口接收到 Proposal，并且已经同步时，它会向回发送 agreement，而且只有在所有对应的 MSTI 端口都同步后才会进入转发状态。MSTI 端口现在有一种特殊的 master 角色。
- 边界端口不是 CIST 域根的根端口——MSTI 端口会按照 CIST 端口的状态进行操作。这种标准提供的信息比较少，在 MSTI 端口接收不到 BPDU 时，它也许很难理解为什么 MSTI 端口会被阻塞。此时，虽然边界的角色已经不复存在，但用户可以在 **show** 命令输出信息中的 *type* 一列中看到将端口标识为边界（boundary）端口。

与传统和标准设备的互操作

由于对预标准设备进行自动检测有可能会失败，因此可以用以使用一条接口配置命令来设置预标准端口。一个域不能由标准设备和预标准设备组成，但它们可以通过使用 CIST 来进行互操作。只有在有些情况下，无法通过不同的实例实现负载分担。当端口接收到预标准的 BPDU 时，CLI 会根据端口的配置显示不同的标记。当设备通过一个没有配置预标准 BPDU 传输的端口上接收到了一条预标准的 BPDU 时，系统日志消息也会出现。

假设 A 是一台标准设备，而 B 是一台预标准的设备，这两台设备都配置在了同一个域中。A 是 CIST 的根设备，而 B 在网段 X 上有一个根端口（BX）而在网段 Y 上有一个替代端口（BY）。如果网段 Y 出现翻动，而 BY 上的端口会在对外发送一个预标准的 BPDU 之前成为替代端口，那么 AY 就无法检测到预标准设备连接到 Y，因此会继续发送标准 BPDU。在边界上，端口 BY 是固定的，A 和 B 之间不可能执行负载分担。网段 X 上也存在相同的问题，但 B 可能会传输拓扑的变化。

表 47：标准和预标准设备的互操作

Segement X	网段 X
MST Region	MST 域
Switch A	交换机 A
Switch B	交换机 B
Segement Y	网段 Y

注释： 我们推荐用户尽可能减少标准和预标准 MST 实施方案的互动。

检测单向链路失效

这种特性还没有添加到 IEEE MST 标准当中，但包含在了这个 Inspur INOS 版本中。软件系统会通过接收到的 BPDU 来校验端口角色和状态的连续性，以检测网路中是否出现了有可能引发桥接环路的单向链路失效。

当指定端口检测到冲突时，它会保持自己的端口角色，但同时回退到转发状态。因为在出现不连续的情况时，网络是希望打破连通性来打开桥接环路的。

下面这张图显示了一个导致了桥接环路的单向链路失效问题。设备 A 是根设备，它在通向设备 B 的链路上丢失了 BPDU。RSTP 和 MST BPDU 包含了发送方端口的角色和状态。通过这些信息，设备 A 可以检测到设备 B 没有对自己发送的更优 BPDU 作出响应，而设备 B 在所连网

段充当的是指定设备，而不是根设备。于是，设备 A 阻塞了这个端口，这就避免了桥接环路的出现。

图 48：检测单向链路失效

Switch A	交换机 A
Switch B	交换机 B
Superior BPDU	更优 BDPU
Inferiror BPDU Designated+Learning bit set	较差 BPDU 指定角色+学习位置位

MSTP 与设备堆栈

设备堆栈在网络其他部分看来是一台设备，所有堆栈成员对于一棵给定的生成树使用相同的桥 ID。其桥 ID 取自于主用交换机的 MAC 地址。

如果一台不支持 MSTP 的设备被添加到了支持 MSTP 的设备堆栈中，或者一台支持 MSTP 的设备被添加到了不支持 MSTP 的设备堆栈中，那么这台设备都会进入一种版本不匹配状态。如有可能，此时这台设备会自动升级或降级到设备堆栈运行的那个软件版本。

与 IEEE 802.1D STP 的互操作性

运行 MSTP 的设备支持一种内置的协议迁移机制，这种机制可以让设备与传统的 IEEE 802.1D 设备进行互操作。如果这台设备接收到了一个传统的 IEEE 802.1D 配置 BPDU（协议版本设置为 0 的 BPDU），那么这台设备就只会在这个端口发送 IEEE 802.1D BPDU。MSTP 设备也可以在接收到一个传统 BPDU，一个来自不同域的 MSTP BPDU（第 3 版）或者一个 RSTP BPDU（第 2 版）时，检测到一个端口位于区域边界。

不过，当设备没有再接收到 IEEE 802.1D BPDU 时，它并不会自动回退到 MSTP 模式，因为它无法检测出传统的设备是否已经从链路上被移除，除非这台传统设备是指定设备。当连接到一个端口的设备加入到这个域时，设备可能也会继续给这个端口分配边界角色。用户要想重新启动协议迁移进程（强制与邻居设备重新协商），可以输入特权 EXEC 命令 **clear spanning-tree detected-protocols**。

如果链路上的所有传统设备都是 RSTP 设备，那么它们可以像处理 RSTP BPDU 那样处理 MSTP BPDU。因此，MSTP 设备要么会在一个边界端口发送版本 0 的配置和 TCN BPDU，要么则会发送版本 3 的 MSTP BPDU。边界端口会连接到一个局域网，而这个局域网中的指定设备要么是一台单生成树设备，要么是一台拥有不同 MSTP 配置的设备。

RSTP 概述

RSTP 利用了点到点的布线方式，并且提供了生成树快速收敛。因此，生成树的重新配置可以在 1 秒之内完成（而 IEEE 802.1D 生成树中的默认设置则需要 50 秒的时间）。

端口角色与活动拓扑

RSTP 通过分配端口角色和学习活动拓扑的方式，实现了生成树的快速收敛。RSTP 建立在 IEEE

802.1D STP 的基础上，它会将设备优先级最高的设备（也就是优先级值最小的设备）选为根设备。接下来，RSTP 会给各个端口分配下列端口角色之一：

- 根端口：在设备向根设备转发数据包提供最佳路径（即路径开销最小）的端口；
- 指定端口：连接指定设备的端口。在从局域网向根设备转发数据包时，这种端口拥有最小的路径开销。这种指定设备连接局域网的端口称为指定端口；
- 替代端口：提供当前路径之外，另一条通往根设备替代路径的端口；
- 备份端口：提供另一条通往生成树叶网络备份路径的端口。只有当两个端口通过一条链路彼此相连，或者当一台设备域共享 LAN 网段有两条或多条连接时，网络中才会出现备份端口；
- 禁用端口：没有在生成树的操作中扮演任何角色。

根端口或指定端口角色的端口会包含在活动拓扑当中。而角色为替代端口或备份端口的端口则不会包含在活动拓扑当中。

在一个端口角色连续的稳定拓扑中，RSTP 可以确保每个根端口和指定端口立刻过渡到转发状态，而替代端口和备份端口则永远处于丢弃状态（相当于 IEEE 802.1D 中的阻塞状态）。端口状态会控制转发和学习进程的操作。

表 59：端口状态的比较

操作状态	**STP 端口状态（IEEE 802.1D）**	**RSTP 端口状态**	**这种端口是否会包含在活动拓扑中**
启用	阻塞	丢弃	否
启用	侦听	丢弃	否
启用	学习	学习	是
启用	转发	转发	是
禁用	禁用	丢弃	否

为了保证 Inspur 实施方案的一致性，这种指导方针也将端口状态定义为了*阻塞*，而不是*丢弃*。指定端口启用时为侦听状态。

快速收敛

RSTP 提供了设备、设备端口或局域网失效后，连接的快速恢复机制。这种协议为边缘端口、新根端口和通过点到点链路连接的端口提供了快速收敛机制：

- 边缘端口：如果用户在 RSTP 设备上使用接口配置命令 **spanning-tree portfast** 将一个端口配置为了边缘端口，那么边缘端口就会立刻过渡到转发状态。边缘端口相当于启用了 PortFast 的端口，用户应该在连接到终端工作在的端口上配置这条命令；
- 根端口：如果 RSTP 选择了一个新的根端口，它会阻塞老的根端口，并且立刻将新的根端口过渡到转发状态；
- 点到点链路：如果用户用一个端口通过一条点到点链路连接了另一个端口，而这个本地端口成为了指定端口，那么这个端口就会使用 Proposal-Agreement 握手机制来与其他端口协商快速过渡的方法，以确保拓扑是无环的。

  设备 A 通过一条点到点链路与设备 B 相连，所有所有端口都处于阻塞状态。假设设备 A 的优先级数值小于设备 B 的优先级值，那么设备 A 会向设备 B 发送一条 Proposal 消息（设置了 Proposal 标记的配置 BPDU），提议自己为指定设备。

  在接收到这个 Proposal 消息之后，设备 B 会从接收到 Proposal 消息的端口选择出新的根端口，并且强制所有非边缘端口进入阻塞状态，并且通过新的根端口发送一条 Agreement 消息（设置了 Agreement 标记的配置 BPDU）。

  在接收到设备 B 的 Agreement 消息之后，设备 A 也会立刻将其指定端口过渡到转发状态。此时网络中不会形成环路，是因为设备 B 阻塞了它的所有非边缘端口，因为设备 A

和设备 B 之间有一条点到点链路。

当设备 C 连接到设备 B 时，它们之间也会交换一系列类似的握手消息。设备 C 会将连接设备 B 的端口选择为根端口，而这两段会立刻过渡到转发状态。在每次重复握手进程时，都会有另一台设备加入活动拓扑。当网络收敛时，Proposal-Agreement 握手进程会从根一直向生成树的叶网络扩散。

交叉快速过渡（CSRT）特性可以确保设备堆栈中的堆栈成员在 Proposal-Agreement 握手期间从所有堆栈成员那里接收到了确认消息，然后才会将端口过渡到转发状态。当设备进入 MST 模式时，CSRT 就会自动启用。

设备会从端口双工模式学习到链路类型，全双工端口会被视为是一条点到点链路，而半双工端口则会被视为是一条共享连接。用户可以使用接口配置命令 **spanning-tree link-type** 来覆盖通过双工设置学习到的默认设置。

图 49：Proposal 与 Agreement 握手以实现快速收敛

Switch A	交换机 A
Switch B	交换机 B
Root	根交换机
Designated switch	指定交换机
Designated switch	指定交换机
Root	根交换机
Switch C	交换机 C
Designated switch	指定交换机
DP=designated port	DP=指定端口
RP=root port	RP=根端口
F=forwarding	F=转发
Root	根交换机

同步端口角色

当设备通过自己的一个端口接收到一个 proposal 消息，而这个端口又被选为了新的根端口时，那么 RSTP 就会强制所有其他端口用新端口的信息进行同步。

如果所有其他端口都进行了同步，设备就会使用根端口接收到的更优的根信息来进行同步。在发生下列情况时，设备的一个端口会进行同步：

- 这个端口处于阻塞状态；
- 这是一个边缘端口（用户配置为网络边界的端口）

如果指定端口处于转发状态，但用户又没有将它配置为边缘端口，那么当 RSTP 强制它使用新的根信息进行同步时，它就会过渡为阻塞状态。总地来说，当 RSTP 强制一个端口用根信息进行同步时，而这个端口又不满足任何上述条件，那么它的端口状态就会被设置为阻塞状态。

在确认了所有端口都已经同步之后，设备会向其根端口连接的指定设备发送一条 agreement 消息。当通过点到点链路连接的设备都同意了它们的端口角色时，RSTP 会立刻将端口过渡到转发状态。

图 50：快速收敛过程中的事件发生顺序

5. Forward	5.转发
Edge port	边缘端口
2.Block 9.Forward	2.阻塞 9.转发
3.Block	3.阻塞

11.Forward	11.转发
Root port	根端口
Designated port	指定端口

桥协议数据单元的格式与处理

RSTP 的 BPDU 格式与 IEEE802.1D BPDU 的格式相同，只不过协议版本变为了 2。有一个新的 1 字节版本 1 长度（Version 1 Length）字段会被设置为 0，这表示不存在第 1 版协议信息。

表 60：RSTP BPDU 标记

位	功能
0	拓扑变更（TC）
1	Proposal
2-3： 00 01 10 11	端口角色： 未知 替代端口 根端口 指定端口
4	学习
5	转发
6	Agreement
7	拓扑变化确认（TCA）

发送方设备会设置 RSTP BPDU 中的 proposal 标记，其目的是宣告自己是这个网段的指定设备。在 proposal 消息中的端口角色永远会被设置为指定端口。

发送方设备会设置 RSTP BPDU 中的 agreement 标记，其目的接受之前的提议（proposal）。在 agreement 消息中的端口角色永远会被设置为根端口。

RSTP 没有独立的拓扑变更通告（TCN）BPDU。它会使用拓扑变更（TC）标记来显示拓扑的变更。不过，为了和 IEEE 802.1D 设备进行互操作，RSTP 设备会处理和生成 TCN BPDU。

RSTP 会根据发送端口的状态来设置学习和转发标记。

处理更优的 BPDU 信息

如果一个端口接收到了比当前给这端口保存的根信息更优的根信息（设备 ID 较低、路径开销较低等），RSTP 就会触发重配置。如果这个端口经过提议被选为了新的根端口，那么 RSTP 就会强制所有其他端口进行同步。

如果 BPDU 接收到一条设置了 proposal 标记的 RSTP BPDU 消息，那么设备就会在其他端口同步之后发送一条 agreement 消息。如果 BPDU 是一个 IEEE 802.1D BPDU，那么设备就不会设置 proposal 标记，并且对这个端口启动转发延迟计时器。新的根端口需要两倍的转发延迟时间才能过渡到转发状态。

如果端口接收到的更优信息导致这个端口成为了备份端口或替代端口，那么 RSTP 就会将这个端口置入阻塞状态，而不会发送 agreement 消息。指定端口会继续发送设置了 proposal 标记的 BPDU，直至转发延迟计时器过期为止，此时这个端口就会过渡到转发状态。

处理较差的 BPDU 信息

如果一个指定端口接收到了一个较差的 BPDU（比如其设备 ID 更高，或者路径开销值大于当前给这个端口保存的路径开销值），它会立刻用自己的信息作出响应。

拓扑变更

在这一部分，我们会介绍 RSTP 和 IEEE 802.1D 在处理生成树变更时的区别：

- 检测：在 IEEE 802.1D 环境中，*一切*状态变化都会导致拓扑变更。而在 RSTP 环境中，只有从阻塞状态过渡到转发状态才会导致拓扑变更（也就是说，只有连接数量增加才会被

视为拓扑变更）。边缘端口的状态变更不会导致拓扑变更。当 RSTP 设备检测到拓扑变更时，它会删除所有在非边缘端口上学习到的信息——除了接收到 TC 通告的那个端口之外。

- 通告：IEEE 802.1D 使用的是 TCN BPDU，但 RSTP 没有使用这种 BPDU。但为了支持与 IEEE 802.1D 进行互操作，RSTP 设备会处理和生成 TCN BPDU；
- 确认：当 RSTP 设备接在指定端口从一台 IEEE 802.1D 设备那里接收到一条 TCN 消息时，它会使用 TCA 位置位的 IEEE 802.1D 配置 BPDU 进行响应。不过，如果在与 IEEE 802.1D 设备直连的根端口上，TC-while 计时器（与 IEEE 802.1D 中的拓扑变更计时器相同）是活动的，同时该端口接收到了一个 TCA 位置位的配置 BPDU，那么 TC-while 计时器就会被重置。

  只有在支持 IEEE 802.1D 设备才需要执行这种操作。RSTP BPDU 永远不会置位 TCA 位；
- 传播：当 RSTP 通过指定端口或根端口从另一台设备那里接收到一条 TC 消息时，它会将这个变化传播给所有非边缘的指定端口，以及根端口（不包括接收到 TC 消息的那个端口）。设备会对所有这类端口启动 TC-while 计时器，并冲刷掉在这些端口学习到的信息；
- 协议迁移：为了能够向后兼容 IEEE 802.1D 设备，RSTP 会有选择地从不同端口发送 IEEE 802.1D 配置 BPDU 和 TCN BPDU。

  当一个端口启动时，迁移延迟计时器就会启动（设置 RSTP BPDU 发送的最小时间）。当计时器处于活动状态时，设备会处理所有从这个端口接收到的 BPDU，同时忽略协议类型。

  如果设备在迁移延迟计时器过时之后接收到了一条 IEEE 802.1D BPDU，它会假设自己连接的是一台 IEEE 802.1D 设备，并自此只使用 IEEE 802.1D BPDU。但是，如果 RSTP 设备在计时器过期之后，在一个端口使用 IEEE 802.1D BPDU，同时又接收到了 RSTP BPDU，那么它就会重新启动计时器，并且开始在这个端口使用 RSTP BPDU。

协议迁移进程

运行 MSTP 的设备会支持内置的协议迁移机制，这种机制可以让设备与传统的 IEEE 802.1D 设备进行互操作。如果这台设备接收到了一条传统的 IEEE 802.1D 配置 BPDU（即协议版本设置为 0 的 BPDU），它就只会通过这个端口发送 IEEE 802.1D BPDU。MSTP 设备也可以在接收到传统 BPDU、不同域的 MST BPDU（第 3 版）或 RST BPDU（第 2 版）时，检测到这个端口位于域的边界。

不过，当一台设备没有再接收到 IEEE 802.1D BPDU 时，它并不会自动回退到 MSTP 模式，因为它无法检测出传统的设备是否已经从链路上被移除，除非这台传统设备是指定设备。当连接到一个端口的设备加入到这个域时，设备可能也会继续给这个端口分配边界角色。

默认的 MSTP 配置

表 61：默认的 MSTP 配置

特性	**默认设置**
生成树模式	MSTP
设备优先级（可以基于 CIST 端口进行配置）	32768
生成树端口优先级（可以基于 CIST 端口进行配置）	128

生成树端口开销（可以基于 CIST 端口进行配置）	1000Mb/s：20000 100Mb/s：20000 10Mb/s：20000 1000Mb/s：20000 100Mb/s：20000 10Mb/s：20000
Hello 时间	3 秒
转发延迟时间	20 秒
最大老化时间	20 秒
最大跳数	20 跳

如何配置 MSTP 特性

设置 MST 域的配置与启用 MSTP（CLI）

对于同一个 MST 域中的两台或多台设备，它们必须拥有相同的 VLAN 与实例映射关系、相同的修订版本号和相同的名称。
一个域中可以拥有一个或多个 MST 配置相同的成员；每个成员都必须能够处理 RSTP BPDU。
一个网络中的 MST 域没有数量限制，但每个域只能支持最多 65 个生成树实例。用户每次只能将一个 VLAN 分配给一个生成树实例。

总步骤

1. enable
2. configure terminal
3. spanning-tree mst configuration
4. instance *instance-id* **vlan** *vlan-range*
5. name *name*
6. revision *version*
7. show pending
8. exit
9. spanning-tree mode mst
10. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**spanning-tree mst**	进入 MST 配置模式

	configuration 示例： `Device(config)#` **spanning-tree mst configuration**	
步骤 4	**instance** *instance-id* **vlan** *vlan-range* 示例： `Device(config-mst)#` **instance 1 vlan 10-20**	将 VLAN 映射为一个 MST 实例。 • 在 *instance-id* 部分，取值范围为 0 到 4094； • 在 **vlan** *vlan-range* 部分，取值范围为 1 到 4094；在将 VLAN 映射为一个 MST 实例时，映射是递增的，命令中指定的 VLAN 会被添加到之前映射的 VLAN 当中，或者从之前映射的 VLAN 中移除。 要指定 VLAN 范围，可以使用连字符。例如，**instance 1 vlan 1-63** 是将 VLAN 1 到 63 映射为 MST 实例 1；要指定VLAN范围，可以使用逗号。例如，**instance 1 vlan 10, 20, 30**是将VLAN 10、20和30映射为MST实例1
步骤 5	**name** *name* 示例： `Device(config-mst)#` **name region1**	指定配置名。*name* 这个字符串的最大长度为 32 个字符，而且是区分大小写的
步骤 6	**revision** *version* 示例： `Device(config-mst)#` **revision 1**	指定配置修订版本号。取值范围是从 0 到 65535
步骤 7	**show pending** 示例： `Device(config-mst)#` **show pending**	通过显示未定信息来验证配置
步骤 8	**exit** 示例： `Device(config-mst)#` **exit**	应用所有的修改，并返回全局配置模式
步骤 9	**spanning-tree mode mst** 示例： `Device(config)#` **spanning-tree mode mst**	启用 MSTP，则 RSTP 也会启用。 修改生成树的模式会导致流量中断，因为所有生成树实例都会停止执行之前的模式，并且在新的模式下重新启动。 用户不能同时运行 MSTP 与 PVST+，或者同时运行 MSTP 与快速 PVST+
步骤 10	**end**	返回特权 EXEC 模式

	示例： `Device(config)# end`	

配置根设备（CLI）

这个流程是可选的。

在开始前

设备上必须设置和启用多生成树（MST）。要想了解详细信息，可以参见相关主题。

用户必须清楚设置的 MST 实例 ID。示例中的步骤 2 以 0 作为实例 ID，是因为这是按照相关主题下面列出的方法设置的实例 ID。

总步骤

1. enable

2. configure terminal

3. spanning-tree mst *instance-id* **root primary**

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**spanning-tree mst** *instance-id* **root primary** 示例： `Device(config)#` **spanning-tree mst 0 root primary**	将一台设备配置为根设备。 • 在 *instance-id* 部分，用户可以设置一个实例，可以用连字符设置一个实例范围，也可以用逗号相隔输入一系列的实例。实例的取值范围是从 0 到 4094
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置辅助根设备（CLI）

在将一台支持扩展的系统 ID 的设备配置为辅助根时，设备优先级会从默认值（32768）修改为 28672。如果这个实例的主用根发生了故障，那么设备使用这个优先级就更有可能会成为这个实例的根设备。这一点的前提是其他网络设备使用的都是默认的设备优先级 32768，因此这些设备就很难成为根设备。

用户可以在多台设备上执行这条命令，来将多台设备配置为备份根设备。用户也可以使用配

置主用根设备时使用的命令 **spanning-tree mst** *instance-id* **root primary** 来设置辅助根设备的网络直径与 hello 时间值。

这个流程是可选的。

在开始前

设备上必须设置和启用多生成树（MST）。要想了解详细信息，可以参见相关主题。

用户必须清楚设置的 MST 实例 ID。示例中以 0 作为实例 ID，是因为这是按照相关主题下面列出的方法设置的实例 ID。

总步骤

1. enable

2. configure terminal

3. spanning-tree mst *instance-id* **root secondary**

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**spanning-tree mst** *instance-id* **root secondary** 示例： `Device(config)# spanning-tree mst 0 root secondary`	将一台设备配置为辅助根。 • 在 *instance-id* 部分，用户可以设置一个实例，可以用连字符设置一个实例范围，也可以用逗号相隔输入一系列的实例。实例的取值范围是从 0 到 4094
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置端口优先级（CLI）

当环路出现时，MSTP 会使用端口优先级来选择一个接口，并将其置入转发状态。用户可以给自己希望首先被选中的接口设置较高的（数值较小的）优先级值，给自己希望之后被选中的接口设置较低的（数值较大的）优先级值。如果所有接口的优先级值相同，那么 MSTP 就会将接口编号最低的接口置入转发状态，并阻塞其他接口。

注释： 如果用户的设备是设备堆栈的成员，那就必须使用接口配置命令 **panning-tree mst** [*instance-id*] **cost** *cost*（而不是接口配置命令 **spanning-tree mst** [*instance-id*] **port-priority** *priority*）来选择将一个接口置入转发状态。用户可以给自己希望首先选择的接口配置一个较低的开销值，而给自己希望之后选择的接口配置一个较高的开销值。要想了解详细信息，可以参见相

关主题。

这个流程是可选的。

在开始前

设备上必须设置和启用多生成树（MST）。要想了解详细信息，可以参见相关主题。

用户必须清楚设置的 MST 实例 ID 和接口。示例中以 0 作为实例 ID，以 GigabitEthernet1/0/1 为接口，是因为这是按照相关主题下面列出的方法设置的实例 ID。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. spanning-tree mst *instance-id* **port-priority** *priority*

5. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)# interface GigabitEthernet1/0/1`	指定要配置的接口，进入接口配置模式。
步骤 4	**spanning-tree mst** *instance-id* **port-priority** *priority* 示例： `Device(config-if)# spanning-tree mst 0 port-priority 64`	配置端口优先级。 • 在 *instance-id* 部分，用户可以设置一个实例，可以用连字符设置一个实例范围，也可以用逗号相隔输入一系列的实例。实例的取值范围是从 0 到 4094； • 在 *priority* 部分，取值范围是从 0 到 240，增量为 16，默认值是 128。有效的取值包括 0、16、32、48、64、80、96、112、128、144、160、176、192、208、224 和 240。配置其他值系统都会拒绝
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

特权 EXEC 命令 **show spanning-tree mst interface** *interface-id* 只会显示那些链路处于 up 状态的端口信息。否则，用户也可以使用特权 EXEC 命令 **show running-config interface** 来确认自己所作的配置。

配置路径开销（CLI）

MSTP 路径开销的默认值取自于接口的媒体速率。当环路出现时，MSTP 会使用开销来选择一个接口，并将其置入转发状态。用户可以给自己希望首先被选中的接口设置较低的开销值，给自己希望之后被选中的接口设置较高的开销值。如果所有接口的开销值相同，那么 MSTP 就会将接口编号最低的接口置入转发状态，并阻塞其他接口。
这个流程是可选的。

在开始前

设备上必须设置和启用多生成树（MST）。要想了解详细信息，可以参见相关主题。
用户必须清楚设置的 MST 实例 ID 和接口。示例中以 0 作为实例 ID，以 GigabitEthernet1/0/1 为接口，是因为这是按照相关主题下面列出的方法设置的实例 ID。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. spanning-tree mst *instance-id* **cost** *cost*

5. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet1/0/1**	指定要配置的接口，进入接口配置模式。有效接口包括物理端口 port channel 逻辑接口。port-channel 的取值范围是从 1 到 48
步骤 4	**spanning-tree mst** *instance-id* **cost** *cost* 示例： `Device(config-if)#` **spanning-tree mst 0 cost 17031970**	配置开销。 如果出现环路，MSTP 在选择要将哪个接口置入转发状态时就会使用路径开销进行判断。路径开销越低表示传输速率越高。 • 在 *instance-id* 部分，用户可以设置一个实例，可以用连字符设置一个实例范围，也可以用逗号相隔输入一系列的实例。实例的取值范围是从 0 到 4094； • 在 *cost* 部分，取值范围是从 1 到 200000000，这个值取自于接口媒体的速率
步骤 5	**end**	返回特权 EXEC 模式

	示例： `Device(config)# end`	

特权 EXEC 命令 **show spanning-tree mst interface** *interface-id* 只会显示那些链路处于 up 状态的端口信息。否则，用户也可以使用特权 EXEC 命令 **show running-config** 来确认自己所作的配置。

配置设备优先级（CLI）

用户可以配置设备的优先级，让一台独立设备或一台堆栈中的设备更有可能被选为根设备。

注释： 在使用这条命令时务请小心。在正常的网络配置中，我们推荐用户使用全局配置命令**spanning-tree mst** i*nstance-id* **root primary**和**spanning-tree mst** *instance-id* **root secondary**来将一台设备设置为根或辅助根设备。只有当这些命令没有生效时，用户才应该考虑修改设备优先级。

这个流程是可选的。

在开始前

设备上必须设置和启用多生成树（MST）。要想了解详细信息，可以参见相关主题。

用户必须清楚设置的 MST 实例 ID。示例中以 0 作为实例 ID，是因为这是按照相关主题下面列出的方法设置的实例 ID。

总步骤

1. enable

2. configure terminal

3. spanning-tree mst *instance-id* **priority** *priority*

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**spanning-tree mst** *instance-id* **priority** *priority* 示例： `Device(config)#` **spanning-tree mst 0 priority 40960**	配置设备优先级。 • 在 *instance-id* 部分，用户可以设置一个实例，可以用连字符设置一个实例范围，也可以用逗号相隔输入一系列的实例。实例的取值范围是从 0 到 4094； • 在 *priority* 部分，取值范围是从 0 到 61440，增量为 4096，默认值是 32768。数值越低，设备越有可能被选为根设备。 有效的值包括 4096、8192、12288、16384、

		20480、24576、28672、32768、36864、40960、45056、49152、53248、57344 和 61440。只有输入这些值设备才会接受
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置 Hello 时间（CLI）

hello 时间是根设备生成和发送配置消息的时间间隔。
这个流程是可选的。

在开始前

设备上必须设置和启用多生成树（MST）。要想了解详细信息，可以参见相关主题。

总步骤

1. enable

2. configure terminal

3. spanning-tree mst hello-time *seconds*

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**spanning-tree mst hello-time** *seconds* 示例： `Device(config)#` **spanning-tree mst hello-time 4**	配置所有 MST 实例的 hello 时间。hello 时间是根设备生成和发送配置消息的时间间隔。这些消息表示这台设备当前仍然正常工作。 在 *seconds* 部分，取值范围是从 1 到 10；默认值为 3
步骤 4	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式

配置转发延迟时间（CLI）

在开始前

设备上必须设置和启用多生成树（MST）。要想了解详细信息，可以参见相关主题。

总步骤

1. enable

2. configure terminal

3. spanning-tree mst forward-time *seconds*

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**spanning-tree mst forward-time** *seconds* 示例： `Device(config)# spanning-tree mst forward-time 25`	配置所有 MST 实例的转发时间。转发延迟是接口在将自己的生成树学习和侦听状态过渡到转发状态之前，等待的秒数。 在 *seconds* 部分，取值范围是从 4 到 30，默认值是 20
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置最大老化时间（CLI）

在开始前

设备上必须设置和启用多生成树（MST）。要想了解详细信息，可以参见相关主题。

总步骤

1. enable

2. configure terminal

3. spanning-tree mst max-age *seconds*

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device#` **configure terminal**	
步骤 3	**spanning-tree mst forward-time** *seconds* 示例： `Device(config)#` **spanning-tree mst forward-time 25**	配置所有 MST 实例的最大老化时间。转发延迟是端口从生成树状态从学习和侦听状态过渡到转发状态之前，等待的最大秒数。 在 *seconds* 部分，取值范围是从 4 到 30，默认值是 20
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置最大老化时间（CLI）

在开始前

设备上必须设置和启用多生成树（MST）。要想了解详细信息，可以参见相关主题。

总步骤

1. enable

2. configure terminal

3. spanning-tree mst max-age *seconds*

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**spanning-tree mst max-age** *seconds* 示例： `Device(config)#` **spanning-tree mst max-age 40**	配置所有 MST 实例的最大老化时间。最大老化时间是指设备从没有接收到生成树配置消息开始，会等待多久才会开始执行重新配置。 在 *seconds* 部分，取值范围是从 6 到 40，默认值是 20
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置最大跳数（CLI）

这个流程是可选的。

在开始前

设备上必须设置和启用多生成树（MST）。要想了解详细信息，可以参见相关主题。

总步骤

1. enable

2. configure terminal

3. spanning-tree mst max-hops *hop-count*

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**spanning-tree mst max-hops** *hop-count* 示例： `Device(config)#` **spanning-tree mst max-hops 25**	配置 BPDU 被丢弃，且针对端口保存的信息老化之前，在域中可以转发的跳数，。 在 *hop-count* 部分，取值范围是从 1 到 255，默认值是 20
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

设置链路类型以确保快速过渡（CLI）

如果用户用一个端口通过一条点到点链路连接了另一个端口，而这个本地端口成为了指定端口，那么 RSTP 就会使用 Proposal-Agreement 握手机制来与其他端口协商快速过渡的方法，以确保拓扑是无环的。

在默认情况下，设备的链路类型是受双工模式控制的：全双工端口会被视为是一条点到点链路，而半双工端口则会被视为是一条共享连接。如果用户用一条半双工链路物理地点到点连接到一台运行 MSTP 的远程设备，用户可以覆盖默认设置的链路类型，让端口快速过渡到转发状态。

这个流程是可选的。

在开始前

设备上必须设置和启用多生成树（MST）。要想了解详细信息，可以参见相关主题。

用户必须清楚设置的 MST 实例 ID 和接口。示例中以 0 作为实例 ID，以 GigabitEthernet1/0/1 为接口，是因为这是按照相关主题下面列出的方法设置的实例 ID。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. spanning-tree link-type point-to-point

5. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)# interface gigabitethernet1/0/1`	指定要配置的接口，进入接口配置模式。有效接口包括物理端口、VLAN 和 port channel 逻辑接口。VLAN ID 的取值范围是 1 到 4094，port-channel 的取值范围是从 1 到 48
步骤 4	**spanning-tree link-type point-to-point** 示例： `Device(config-if)# spanning-tree link-type point-to-point`	将端口的链路类型设置为点到点
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

指定邻居类型（CLI）

拓扑应该同时包含符合预标准和 IEEE 802.1s 标准的设备。在默认情况下，端口可以自动检测到预标准设备，但它们仍然可以同时接收标准的和预标准的 BPDU。只要设备与邻居之间出现了不匹配的情况，那接口上就只能运行 CIST。

用户可以选择设置一个端口，让它只发送预标准的 BPDU。即使端口工作在匹配 STP 的模式下，**show** 命令还是会显示出预标准的标记。

这个流程是可选的。

在开始前

设备上必须设置和启用多生成树（MST）。要想了解详细信息，可以参见相关主题。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. spanning-tree mst pre-standard

5. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet1/0/1**	指定要配置的接口，进入接口配置模式。有效接口包括物理端口
步骤 4	**spanning-tree mst pre-standard** 示例： `Device(config-if)#` **spanning-tree mst pre-standard**	设置端口，让它只发送预标准的 BPDU
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

重新启动协议迁移进程（CLI）

这个流程会重新启动协议迁移进程，并且强制与邻居设备进行重新协商。它会让设备回退到 MST 模式。当设备在接收到 IEEE 802.1D BPDU 之后，没有继续接收到 IEEE 802.1D BPDU，它就需要执行这个流程。

用户可以按照下面的步骤在设备上重新启动协议迁移进程（强制与邻居设备进行重新协商）。

在开始前

设备上必须设置和启用多生成树（MST）。要想了解详细信息，可以参见相关主题。

如果用户想要使用接口版本命令，就必须清楚设置的 MS 接口。示例中以 GigabitEthernet1/0/1 为接口，是因为这是按照相关主题下面列出的方法设置的接口。

总步骤

1. enable

2. 输入下面命令之一：

- **clear spanning-tree detected-protocols**
- **clear spanning-tree detected-protocols interface** *interface-id*

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**输入下面命令之一：** • **clear spanning-tree detected-protocols** • **clear spanning-tree detected-protocols interface** *interface-id* 示例： `Device#` **clear spanning-tree detected-protocols** 或 `Device#` **clear spanning-tree detected-protocols interface GigabitEthernet1/0/1**	设备回退到 MSTP 模式，协议迁移进程会重新启动

接下来做什么

如果设备接收到了更多传统的 IEEE 802.1D 配置 BPDU，那么这个流程有可能需要重复执行（协议版本设置为 0 的 BPDU）。

其他 MSTP 的参考资料

相关文档

相关主题	文档名
生成树协议的命令	《LAN 交换命令参考手册，Inspur INOSXE3SE 版（Inspur 6650 交换机）》

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com/icnt

标准与 RFC

标准/RFC	标题

无	——

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com/icnt

MSTP 的特性历史

版本	修改
Inspur INOS 11.3.1	引入该特性

配置可选的生成树特性

关于可选生成树特性的信息

PortFast

PortFast 可以让一个配置为 Access 或 trunk 模式的接口立刻从阻塞状态进入转发状态，跳过侦听状态和学习状态。

用户可以在一个连接到一台工作站或服务器的接口上使用 PortFast 特性，让这些设备能够立刻连接到网络，而无需等待生成树的收敛。

图 51：启用了 PortFast 的接口

Port Fast-enabled	启用了 PortFast 的端口

ports	
Port Fast-enabled ports	启用了 PortFast 的端口
Workstations	工作站
Workstations	工作站
Server	服务器

当交换机重新启动时，启用了 PortFast 的接口会按照正常的生成状态变化流程进行状态迁移。
用户可以在这个接口或者所有非中继端口上启用这个特性。

BPDU 防护

桥协议数据单元（BPDU）防护特性可以在交换机上全局启用，也可以在个别端口上启用，但协议的操作方式存在一些区别。
当用户在启用了 PortFast 边缘的端口上全局启用 BPDU 防护时，如果处于 PortFast 边缘操作状态的端口接收到了任何 BPDU，那么生成树就会关闭这些端口。在有效的配置方案中，启用了 PortFast 边缘的端口不会接收到 BPDU。如果在启用了 PortFast 边缘的端口上接收到了 BPDU，表示设备上存在无效的配置（譬如有未授权设备连接），那么 BPDU 防护特性就会将这个端口置入 error-disabled 状态。在发生这件事时，交换机就会关闭发生错误的端口。
当用户在没有启用 PortFast 边缘特性的接口上启用 BPDU 防护，那么当这个端口接收到 BPDU 时，它就会进入 error-disabled 状态。
BPDU 防护特性给无效配置提供了一种安全的响应方式，因为 BPDU 防护特性必须由用户手动在接口上进行配置。在服务提供商网络中使用 BPDU 防护特性可以防止 access 端口参与到生成树当中。

BPDU 过滤

用户可以在交换机全局启用 BPDU 过滤特性，也可以在各个接口上启用 BPDU 过滤特性，但协议的操作方式存在一些区别。
在启用了 PortFast 边缘的接口上全局启用 BPDU 过滤，可以让这些处于 PortFast 边缘操作状态的接口不会发送和接收 BPDU。这些接口会在开始过滤出站 BPDU 之前，通过链路发送几个 BPDU。用户应该在交换机上全局启用 BPDU 过滤，让与这些接口直连的主机无法接收到 BPDU。如果设备通过一个启用了 PortFast 边缘的接口接收到了 BPDU，那么这个接口就会丢失它的 PortFast 边缘操作状态，而 BPDU 过滤也会被禁用。
如果用户在没有启用 PortFast 边缘特性的接口上启用了 BPDU 过滤，那么这些接口就不会发送或接收 BPDU 了。
注意： 在接口启用 BPDU 过滤相当于在接口上禁用生成树，因此有可能会导致生成树环路。
用户可以针对整台交换机启用 BPDU 过滤特性，也可以针对一个接口来启用 BPDU 过滤特性。
UplinkFast
在分层网络中的交换机可以分类为骨干交换机、分布层交换机和接入层交换机。下面这个复杂的网络包含了分布层交换机和接入层交换机，它们都有至少一条冗余链路被生成树阻塞，以防网络中出现环路。

图 52：分层网络中的交换机

Backbone switches	骨干交换机
Root bridge	根桥
Distribution switches	分布层交换机
Access switches	接入层交换机
Active link	活动链路
Blocked link	阻塞链路

如果交换机丢失了链路，它就会在生成树选出新的根端口之后立刻开始使用替代路径。用户可以通过启用 UplinkFast，在链路或交换机出现故障，或者在生成树重新配置时，加速新根端口的选择。根端口会立刻过渡到转发状态，而不需要经历侦听或学习状态，这是它与常规生成树处理流程的不同之处。

当生成树重新配置了新的根端口时，其他接口就会在网络中泛洪组播数据包，向每个在这个接口上学习到的地址发送一个组播数据包。用户可以通过减少最大更新速率这个参数（默认值为每秒 150 个数据包）来限制组播流量的突发值。但如果输入 0，那么设备就不会创建工作站学习的数据帧，因此丢失连接之后生成树拓扑的收敛时间也会变慢。

注释： UplinkFast 最适合配置在那些部署在网络接入层或边缘的那些配线柜中的交换机上。这项特性不适合配置在骨干交换机上。这些特性可能对于其他类型的应用用处不大。

在直连链路出现故障时，UplinkFast 可以提供快速收敛，并且使用上行链路组通过冗余的二层链路提供负载分担。一个上行链路组是指（一个 VLAN 中的）多个二层接口，其中只有一个二层接口时钟处于转发状态，具体来说，除了自环端口之外，上行链路组是由（处于转发状态的）根端口、和一系列阻塞端口组成的。在当前转发链路出现故障时，这个上行链路组可以提供替代路径。

这个拓扑当前没有链路出现故障，交换机 A，即根交换机与交换机 B 通过链路 L1 直连，与交换机 C 通过链路 L2 相连。交换机 C 上域交换机 B 直连的二层接口目前处于阻塞状态。

图 53：直连链路故障之前的 UplinkFast 示例

Switch A (Root)	交换机 A （根）
Switch B	交换机 B
Switch C	交换机 C
Blocked port	阻塞端口

如果交换机 C 在连接当前活动链路 L2 的根端口上检测到了链路故障（直连链路故障），那么 UplinkFast 就会启动与交换机 C 之间的阻塞接口，让它过渡到转发状态，而不需要经历侦听和学习状态。这个变更需要经历 1 到 5 秒的时间。

图 54：直连链路故障之后的 UplinkFast 示例

Switch A (Root)	交换机 A （根）
Switch B	交换机 B
Switch C	交换机 C
Link failure	链路故障
UplinkFast transitions port directly to forwarding state.	UplinkFast 直接将端口过渡到转发状态

交叉堆栈 UplinkFast

交叉堆栈 UplinkFast（CSUF）可以跨越堆栈中的交换机提供快速生成树过渡（在正常网络条件下，快速收敛可以在 1 秒之内完成）。在快速过渡期间，交换机堆栈中会有一条替代冗余链路被堆栈置于转发状态，而不会引发临时生成树环路或者丢失与骨干之间的连接。通过这项特性，用户可以通过配置获得一个冗余的，能够快速复原的网络。当用户启用 UplinkFast 特性时，CSUF 也会自动启用。

CSUF 可能无法随时提供快速过渡。在有些情况下，设备也会执行普通的生成树过渡，这需要消耗 30 到 40 秒的时间。要想了解具体信息，可以参考相关主题。

交叉堆栈 UplinkFast 是如何工作的

交叉堆栈 UplinkFast（CSUF）可以确保堆栈中有一条链路会被选为去往根的路径。

交换机 1 上的堆栈根端口提供了去往生成树根的路径。交换机 2 和 3 上的替代堆栈根端口可以在当前堆栈根交换机发生故障，或者去往生成树根的链路故障时，提供去往根的替代路径。

根链路链路 1 处于生成树转发状态。链路 2 和链路 3 则是冗余链路，它们处于生成树阻塞状态。如果交换机 1 发生了故障，如果交换机 1 所在的堆栈根端口发生了故障，或者链路 1 发生了故障，那么 CSUF 就会从交换机 2 和交换机 3 的替代堆栈根端口中选择一个，并在 1 秒之内将其置于转发状态。

图 55：交叉堆栈 UplinkFast 拓扑

Backbone	骨干
Spanning- tree root	生成树根
Forward	转发
Forward	转发
Forward	转发
Link 1 (Root link)	链路 1 （根链路）
Link 2 (Alternate redundant link)	链路 2 （替代冗余链路）
Link 3 (Alternate redundant link)	链路 3 （替代冗余链路）
100 or 1000 Mb/s	100 或 1000Mb/s
100 or 1000 Mb/s	100 或 1000Mb/s
100 or 1000 Mb/s	100 或 1000Mb/s
Stack-root port	堆栈根端口
Alternate stack- root port	替代堆栈根端口
Alternate stack- root port	替代堆栈根端口
Switch 1	交换机 1
Switch 2	交换机 2
Switch 3	交换机 3
StackWise Plus	StackWise Plus

port connections	端口连接
StackWise Plus port connections	StackWise Plus 端口连接
StackWise Plus port connections	StackWise Plus 端口连接
Switch stack	交换机堆栈

当一条链路丢失，或者发生了生成树事件（在下一个主题中进行介绍），那么快速上行链路过渡协议就会使用邻居列表来向堆栈成员发送快速过渡请求。

发送快速过渡请求的交换机需要将选择为根端口的那个端口快速过渡到转发状态，同时它必须从每个堆栈交换机那里获得确认才能执行快速过渡。

堆栈中的每台交换机都会判断发送方交换机是不是比自己更适合成为这个生成树实例的根，它们会通过比较根、开销、桥 ID 来进行判断。如果发送方交换机是成为堆栈根的最佳选择，那么堆栈中的每台交换机都会返回一条确认；否则，它就会发送一条快速过渡请求。此时，发送方交换机还没有从所有堆栈交换机那里接收到确认。

当发送方交换机从所有堆栈交换机那里接收到确认时，发送方交换机上运行的快速上行链路过渡协议就会立刻将它的替代堆栈根端口过渡到转发状态。如果发送方交换机没有从所有堆栈交换机那里接收到确认，那么交换机就会执行普通的生成树过渡（从阻塞、到侦听、到学习、再到转发），而生成树拓扑也会按照正常的速率进行收敛（2 倍转发延迟+最大老化时间）。快速上行链路过渡协议是基于每个 VLAN 实施的，每次只会影响一个生成树实例。

导致快速收敛的事件

根据网络事件或者故障的不同，CSUF 有可能会执行快速收敛，也有可能不会执行快速收敛。在下列情况下，会发生快速收敛（在正常网络条件下，快速收敛可以在 1 秒之内完成）：

- 堆栈根端口链路发生故障。
- 如果堆栈中有两台交换机拥有去往根的替代路径，但只有一台交换机执行了快速过渡；
- 失效链路（连接堆栈根与生成树根的链路）恢复；
- 网络重配置导致网络中选择出了新的堆栈根交换机；
- 网络重配置导致当前堆栈根交换机上的一个端口被选为堆栈根端口。

  注释：如果很多事件同时发生，有可能不会发生快速过渡。例如，如果一个堆栈成员掉电，同时连接堆栈根与生成树根的链路恢复，那么网络就会执行普通的生成树收敛。

在下列情况下，会发生普通的生成树收敛（耗时 30-40 秒）：

- 堆栈根交换机掉电，或者软件运行失败；
- 掉电或故障的堆栈根交换机恢复；
- 有可能成为堆栈根的新交换机加入了堆栈。

BackboneFast

BackboneFast 可以检测出非直连的骨干网核心故障。BackboneFast 是对 UplinkFast 特性的一项补充技术，后者可以对直连的接入交换机的故障作出响应。BackboneFast 优化了最大老化计时器，这个计时器会控制交换机在接口保存接收到的协议信息的总时长。当交换机从另一台交换机的指定端口那里接收到一个较差的 BPDU，那么这个 BPDU 可能表示对方已经失去了去往根的路径，BackboneFast 会尝试找到去往根的替代路径。

当交换机上的一个根端口或阻塞的接口从其指定交换机那里接收到交叉 BPDU 时，BackboneFast 就会启动。交叉 BPDU 表示有一台交换机正在将自己宣称为根桥和指定交换机。

当交换机接收到较差 BPDU 时，这表示有一条与这台交换机并不直连的链路发生了故障（也就是说，指定交换机丢失了去往根交换机的连接）。根据生成树的规则，交换机会在最大老化时间内（默认为 20 秒）忽略较差 BPDU。

交换机会尝试发现是否有去往根交换机的替代路径。如果较差 BPDU 到达被阻塞的接口，那么交换机上的根端口和其他被阻塞的端口就会成为去往根交换机的替代路径。（自环端口不会成为去往根交换机的替代路径）如果较差 BPDU 到达的是根端口，那么所有阻塞的接口都会成为去往根交换机的替代路径。如果较差 BPDU 到达的是根端口，而交换机上又没有被阻塞的接口，那么交换机就会认为自己已经丢失了与根交换机的连接，因此根端口上的最大老化时间就会超时，而交换机也会根据常规的生成树规则成为根交换机。

如果交换机拥有去往根交换机的替代路径，它就会使用这些替代路径来发送根链路查询（RLQ）请求。交换机会在所有替代路径发送 RLQ 请求，来学习是否有堆栈成员有通往根交换机的替代根，并等待网络和堆栈中的其他交换机发送 RLQ 响应。交换机会在所有替代路径发送 RLQ 请求，并等待网络中的其他交换机发送 RLQ 响应。

当堆栈成员通过阻塞的接口从非堆栈成员那里接收到一条 RLQ 响应消息时，而这个响应消息的目的是另一台非堆栈交换机时，它会转发这个响应数据包，无论这个接口的生成树状态为何。

当堆栈成员通过阻塞的接口从非堆栈成员那里接收到一条 RLQ 响应消息时，而这个响应消息的目的是堆栈时，它会转发这个响应数据包，让堆栈中的其他成员都能够接收到这条消息。

如果交换机发现自己仍然有一条通向根的替代路径，那么它会让接收到较差 BPDU 的接口最大老化时间超时。如果所有去往根交换机的替代路径都显示交换机已经丢失了与根交换机之间的连接，那么交换机就会让接收到 RLQ 响应消息的接口的最大老化时间超时。如果有一两条替代路径仍然可以连接到根交换机，那么交换机就会让所有接收到交叉 BPDU 的接口成为指定端口，然后让它们从阻塞状态（如果它们之前是阻塞状态的话）经历侦听和学习状态过渡到转发状态。

这个拓扑当前没有链路出现故障，交换机 A，即根交换机与交换机 B 通过链路 L1 直连，与交换机 C 通过链路 L2 相连。交换机 C 上域交换机 B 直连的二层接口目前处于阻塞状态。

图 53：非直连链路故障之前的 BackboneFast 示例

Switch A (Root)	交换机 A （根）
Switch B	交换机 B
Switch C	交换机 C
Blocked port	阻塞端口

如果链路 1 出现了故障，交换机 C 是无法检测到这个故障的，因为交换机 C 没有直接与链路 1 相连。不过，由于交换机 B 通过 L1 直接连接到了根交换机，因此它会检测到这个故障，它会将自己选举为根，并且开始向交换机 C 发送 BPDU，声称自己是根。当交换机 C 从交换机 B 那里接收到较差 BPDU 时，交换机 C 会认为网络中有非直连链路出现了故障。此时，BackboneFast 会让交换机 C 上被阻塞的端口立刻进入侦听状态，而不需要等待接口的最大老化时间超时。接下来，BackboneFast 会将交换机 C 上的二层接口过渡到转发状态，这就提供了一条从交换机 B 去往交换机 A 的路径。根交换机选举需要消耗大约 30 秒的时候，如果用户没有修改默认转发延迟时间的 15 秒，那么这个时间就是转发延迟的 2 倍。BackboneFast 会重新配置拓扑，此时链路 L1 的故障也会被考虑在内。

图 54：非直连链路故障之后的 BackboneFast 示例

Switch A (Root)	交换机 A （根）

Switch B	交换机 B
Switch C	交换机 C
Link failure	链路故障
BackboneFast changes port through listening and learning states to forwarding state.	UBackboneFast 将端口经过侦听状态和学习状态过渡到了转发状态

如果有一台新的交换机添加到了这个共享媒介拓扑当中，BackboneFast 就不会启动，因为交换机无法识别出这些较差 BPDU 是由指定交换机（交换机 B）发送的。新的交换机会开始发送交叉 BPDU，自称是根交换机。但其他交换机会忽略这些较差 BPDU，而新交换机也会学习到交换机 B 是去往交换机 A（根交换机）的指定交换机。

图 55：向共享媒介拓扑中添加一台交换机

Switch A (Root)	交换机 A （根）
Switch B (Designated bridge)	交换机 B （指定网桥）
Switch C	交换机 C
Blocked port	阻塞端口
Added switch	新增交换机

EtherChannel 防护

用户可以使用 EtherChannel 防护来检测交换机与直连设备之间是否有 EtherChannel 的误配置。如果交换机接口配置为 EtherChannel，而另一台设备的相应接口却没有配置为 EtherChannel，就是 EtherChannel 误配置的情形。如果 EtherChannel 通道两端的参数不匹配，也属于 EtherChannel 误配置。

如果交换机检测到另一端的设备存在误配置，EtherChannel 防护就会让交换机接口进入 error-disabled 状态，同时显示错误消息。

根防护

服务提供商（SP）的二层网络可以包含很多通向交换机的连接，这些连接并不属于 SP 所有。在这样的拓扑环境中，生成树有可能会对自己进行重新配置，选择客户交换机为根交换机。用户可以在与客户网络中交换机相连的那些 SP 交换机接口上启用根防护，以避免这种情况的发生。如果生成树计算的结果是客户网络中的接口被选举为根端口，那么根防护就会这个接口进入不连续根（阻塞）状态，以防止客户交换机成为根交换机，或者客户交换机出现在去往根的路径上。

图 59：服务提供商网路中的根防护、

Customer network	客户网络
Service-provider network	服务提供商网络
Potential spanning-tree root without root guard enabled	没有启用根防护的潜在生成树根
Desired root switch	根交换机

Enable the root-guard feature on theses interfaces to prevent switches in the customer network from becoming the root switch or being the path to the root	在这些接口上启用根防护，防止客户交换机成为根交换机，或者客户交换机出现在去往根的路径上

如果 SP 网络之外的交换机成为了根交换机，那么接口就会被阻塞（进入不连续根状态），而生成树会重新选择一台新的根交换机。客户的交换机不会成为根交换机，也不会出现在去往根的路径上。

如果交换机工作在多生成树（MST）模式下，那么根防护就会强制接口成为指定端口。如果在一个内部生成树（IST）实例中的边界端口因为根防护特性而被阻塞，这个接口也会在所有 MST 实例中被阻塞。边界端口是连接一个局域网段的边界端口，这个局域网段的指定交换机要么是一台 IEEE 802.1D 交换机，要么是一台拥有不同 MST 域配置的交换机。

如果用户在一个接口上启用根防护，那么根防护就会应用于这个接口所属的所有 VLAN。VLAN 可以分组并映射到一个 MST 实例当中。

注意： 根防护特性使用不当可能会导致网络丢失连接。

环路防护

用户可以使用环路防护来防止替代端口或根端口因出现单向链路故障而成为指定端口。在整个交换网络中都配置这个特性是最有效的。环路防护可以防止替代端口和根端口成为指定端口，而生成树也不会在根端口或替代端口发送 BPDU。

当交换机工作在 PVST+或快速 PVST+模式下时，环路防护会防止替代端口和根端口成为指定端口，而生成树也不会在根端口或替代端口发送 BPDU。

如果交换机工作在 MST 模式下，那么只有当接口在所有 MST 实例中都被环路防护特性阻塞时，这个非边界端口才不会发送 BPDU。在边界端口上，环路防护会将接口在所有 MST 实例中进行阻塞。

如何配置可选生成树特性

启用 PortFast（CLI）

启用了 PortFast 特性的接口会直接进入生成树转发状态，而不需要等待标准的转发时间延迟。

如果启用语音 VLAN 特性，那么 PortFast 特性也会自动启用。但在禁用语音 VLAN 时，PortFast 并不会自动被禁用。

如果交换机运行的是 PVST+、快速 PVST+或 MSTP，那么用户可以启用这个特性。

注意： 用户只可以在连接一个终端工作站的 access 端口或 trunk 端口上使用 PortFast。在连接到交换机或集线器的接口上启用这个特性会妨碍生成树检测和禁用网络中的环路，而这会导致网络风暴和地址学习问题。

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. spanning-tree portfast [**trunk**]

5. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet1/0/1**	指定要配置的接口，进入接口配置模式
步骤 4	**spanning-tree portfast** [**trunk**] 示例： `Device(config-if)#` **spanning-tree portfast trunk**	在连接到一个工作站或服务器的 access 端口上启用 PortFast。用户可以使用 **trunk** 这个关键字，在 trunk 端口上启用 PortFast。 **注释**：要在 trunk 端口上启用 PortFast，用户必须配置接口配置命令 **spanning-tree portfast trunk**。而在 trunk 端口上输入命令 **spanning-tree portfast** 则不会生效。 用户务必确保 trunk 端口与工作站和服务器之间没有网络环路，然后才能在 trunk 端口上启用 PortFast。 在默认情况下，PortFast 在所有接口上都是禁用的
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

接下来做什么？

用户可以使用全局配置命令 **spanning-tree portfast default** 来在所有非中继端口上全局启用 PortFast 特性。

启用 BPDU 防护（CLI）

如果交换机运行的是 PVST+、快速 PVST+或 MSTP，那么用户就可以启用 BPDU 防护特性。

注意： 用户只可以在连接终端工作站的端口上配置 PortFast 边缘特性；否则，网络就有可能会因为意料之外的拓扑环路而产生数据环路，进而打断交换机和网络的操作。

这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. spanning-tree portfast edge bpduguard default

4. interface *interface-id*

5. spanning-tree portfast edge

6. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**spanning-tree portfast edge bpduguard default** 示例： `Device(config)#` **spanning-tree portfast edge bpduguard default**	在全局启用 BPDU 防护。 在默认情况下，BPDU 防护是禁用的
步骤 4	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet1/0/2**	选择与终端工作站相连的接口，并进入接口配置模式
步骤 5	**spanning-tree portfast edge** 示例： `Device(config-if)#` **spanning-tree portfast edge**	启用 PortFast 边缘特性
步骤 6	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

接下来做什么？

要防止端口关闭，用户可以使用全局配置命令 **errdisable detect cause bpduguard shutdown vlan** 来单独关闭发生了违背事件的端口所在的 VLAN。

用户也可以使用接口配置命令 **spanning-tree bpduguard enable** 在没有启用 PortFast 边缘特性的接口上启用 BPDU 防护。当这个端口接收到 BPDU 时，它就会进入 error-disabled 状态。

启用 BDPU 过滤（CLI）

用户也可以使用接口配置命令 **spanning-tree bpdufilter enable** 在没有启用 PortFast 边缘特性的接口上启用 BPDU 过滤。这条命令可以防止接口发送或接收 BPDU。

注意： 在接口启用 BPDU 过滤相当于在接口上禁用生成树，因此有可能会导致生成树环路。

如果交换机运行的是 PVST+、快速 PVST+或 MSTP，那么用户就可以启用 BPDU 过滤特性。

注意： 用户只可以在连接终端工作站的端口上配置 PortFast 边缘特性；否则，网络就有可能会因为意料之外的拓扑环路而产生数据环路，进而打断交换机和网络的操作。

这个流程是可选的。

总步骤

1. **enable**
2. **configure terminal**
3. **spanning-tree portfast edge bpdufilter default**
4. **interface** *interface-id*
5. **spanning-tree portfast edge**
6. **end**

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**spanning-tree portfast edge bpdufilter default** 示例： `Device(config)#` **spanning-tree portfast edge bpdufilter default**	在全局启用 BPDU 过滤。 在默认情况下，BPDU 过滤是禁用的
步骤 4	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet1/0/2**	选择与终端工作站相连的接口，并进入接口配置模式
步骤 5	**spanning-tree portfast edge** 示例： `Device(config-if)#` **spanning-tree portfast edge**	在这个接口上启用 PortFast 边缘特性

步骤 6	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

为使用冗余链路启用 UplinkFast（CLI）

注释： 在启用 UplinkFast 时，它会影响交换机或交换机堆栈上的所有 VLAN。用户不能单独在一个 VLAN 上配置 UplinkFast

用户可以针对快速 PVST+或 MSTP 配置 UplinkFast 或交叉堆栈 UplinkFast（CSUF）特性，但是在用户将生成树协议的模式修改为 PVST+之前，这个特性还是会处于禁用（未生效）的状态。这个流程是可选的。用户可以按照下面的步骤来启用 UplinkFast 和 CSUF。

在开始前

在配置了交换机优先级的 VLAN 上是无法启用 UplinkFast 的。要给一个配置了交换机优先级的 VLAN 启用 UplinkFast，要首先使用全局配置命令 **no spanning-tree vlan** *vlan-id* **priority** 将这个 VLAN 上的交换机优先级恢复为默认值。

总步骤

1. enable

2. configure terminal

3. spanning-tree uplinkfast [**max-update-rate** *pkts-per-second*]

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**spanning-tree uplinkfast** [**max-update-rate** *pkts-per-second*] 示例： `Device(config)#` **spanning-tree uplinkfast max-update-rate 200**	启用 UplinkFast。 （可选）*pkts-per-second* 部分的取值范围为 0 到 32000 个数据包每秒；默认值为 150。 如果将速率设置为 0，那么工作站学习的数据帧就不会被创建出来，在连接丢失之后，生成树拓扑会收敛得更慢。 在输入这条命令时，CSUF 也会在所有非堆栈端口上启用
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

在启用 UplinkFast 时，所有 VLAN 的交换机优先级都会被设置为 49152。如果用户将路径开销设置为一个小于 3000 的值，同时启用 UplinkFast 或者 UplinkFast 已经启用，那么所有接口和 VLAN trunk 的路径开销都会增加为 3000（如果用户已经将路径开销增加为 3000 或者更高的值，那么路径开销就不会变化）。变更交换机优先级和路径开销会降低一台交换机成为根交换机的几率。
如果禁用 UplinkFast，那么所有 VLAN 的交换机优先级和所有接口的路径开销都会被设置为默认值（如果用户没有将它们修改为默认值的话）。
在用户使用这些方法来启用 UplinkFast 特性时，CSUF 也会在所有非堆栈端口上启用。

禁用 UplinkFast（CLI）

这个流程是可选的。
用户可以按照下面的步骤来禁用 UplinkFast 和交叉堆栈 UplinkFast（CSUF）。
在开始前
用户必须先启用 UplinkFast。
总步骤
1. enable
2. configure terminal
3. no spanning-tree uplinkfast
4. end
具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**no spanning-tree uplinkfast** 示例： `Device(config)# no spanning-tree uplinkfast`	在交换机和所有其 VLAN 上禁用 UplinkFast 与 CSUF
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

如果禁用了 UplinkFast，那么所有 VLAN 的交换机优先级和所有接口的路径开销都会被设置为默认值（如果用户没有将它们修改为默认值的话）。
在用户使用这些方法来禁用 UplinkFast 特性时，CSUF 也会在所有非堆栈端口上禁用。

启用 BackboneFast（CLI）

用户可以启用 BackboneFast 来检测非直连链路的故障，并且让生成树重新配置更快启动。用户可以针对快速 PVST+或 MSTP 配置 BackboneFast 特性，但是在用户将生成树协议的模式修改为 PVST+之前，这个特性还是会处于禁用（未生效）的状态。
这个流程是可选的。用户可以按照下面的步骤来启用 BackboneFast。

在开始前

如果使用 BackboneFast，用户必须在网络中的所有交换机上启用这项特性。令牌环 VLAN 上是不支持 BackboneFast 的。这项特性可以在包含第三方交换机的网络环境中使用。

总步骤

1. enable

2. configure terminal

3. spanning-tree backbonefast

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**spanning-tree backbonefast** 示例： `Device(config)# spanning-tree backbonefast`	启用 BackboneFast
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

启用 EtherChannel 防护（CLI）

如果设备正在运行 PVST+、快速 PVST+或 MSTP，那么用户可以启用 EtherChannel 防护来检测 EtherChannel 的误配置。
这个流程是可选的。
用户可以按照下面的步骤在设备上启用 EtherChannel 防护。

总步骤

1. enable

2. configure terminal

3. spanning-tree etherchannel guard misconfig

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**spanning-tree etherchannel guard misconfig** 示例： `Device(config)# spanning-tree etherchannel guard misconfig`	启用 EtherChannel 防护
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

接下来做什么？

用户可以使用特权 EXEC 命令 **show interfaces status err-disabled** 来查看哪些设备端口因 EtherChannel 误配置而被禁用。在远程设备上，用户可以输入特权 EXEC 命令 **show etherchannel summary** 来验证 EtherChannel 的配置。

在配置验证完毕之后，用户可以在之前误配置的 port-channel 接口上输入接口配置命令 **shutdown** 和 **no shutdown**。

启用根防护（CLI）

在接口上启用的根防护会应用于这个接口属于的每一个 VLAN。用户不要在使用 UplinkFast 特性的接口上启用根防护。通过 UplinkFast，当根端口出现故障时，（阻塞状态下的）备份接口会替代根端口。但如果用户同时也启用了根防护，那么所有 UplinkFast 特性使用的备份接口就会进入不连续根状态（会被阻塞），设备不会让这些端口过渡到转发状态。

注释： 用户不能同时启用根防护和环路防护。

如果设备正在运行 PVST+、快速 PVST+或 MSTP，那么用户可以启用这项特性。

这个流程是可选的。

用户可以按照下面的步骤在设备上启用根防护。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. spanning-tree guard root

5. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)# interface gigabitethernet1/0/2`	选择要配置的接口，并进入接口配置模式
步骤 4	**spanning-tree guard root** 示例： `Device(config-if)# spanning-tree guard root`	在接口上启用根防护。 在默认情况下，根防护是在所有接口上禁用的
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

启用环路防护（CLI）

用户可以使用环路防护来防止替代端口或根端口由于单向链路失效而成为指定端口。在整个交换网络中都配置这个特性是最有效的。环路防护只会在那些被生成树视为点到点的接口上生效。

注释： 用户不能同时启用环路防护和根防护。

如果设备正在运行 PVST+、快速 PVST+或 MSTP，那么用户可以启用这项特性。

这个流程是可选的。用户可以按照下面的步骤在设备上启用环路防护。

总步骤

1. 输入下面命令之一：

• show spanning-tree active

• show spanning-tree mst

2. configure terminal

3. spanning-tree loopguard default

4. end

具体步骤

	命令或操作	目的
步骤 1	**输入下面命令之一：** **• show spanning-tree active** **• show spanning-tree mst** 示例： `Device#` **show spanning-tree active** 或 `Device#` **show spanning-tree mst**	验证哪些接口是替代端口，哪些是根端口
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**spanning-tree loopguard default** 示例： `Device(config)#` **spanning-tree loopguard default**	启用环路防护。 在默认情况下，环路防护是禁用的
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

监控生成树的状态

表 62：监控生成树状态的命令

命令	目的
show spanning-tree active	仅显示活动接口的生成树信息
show spanning-tree detail	显示具体的接口信息
showspanning-tree interface *interface-id*	显示特定接口的生成树信息
show spanning-tree mst interface *interface-id*	显示特定接口的生成树信息
show spanning-tree summary [**totals**]	显示接口状态的汇总信息，或者显示 STP 状态部分的总行
show spanning-tree mst interface *interface-id* **portfast edge**	显示特定接口的生成树 portfast 信息

其他可选生成树特性的参考资料

相关文档

相关主题	文档名
生成树协议的命令	《LAN 交换命令参考手册，Inspur INOSXE3SE 版（Inspur 6650 交换机）》

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com/icnt

标准与 RFC

标准/RFC	标题
无	——

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com/icnt

可选生成树特性的特性历史

版本	修改
Inspur INOS 11.3.1	引入该特性

配置 EtherChannel

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com/go/cfn。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

EtherChannel 的限制条件

下面是 EtherChannel 的限制条件：

- EtherChannel 中的所有端口都必须分配给同一个 VLAN，或者配置为 trunk 端口；
- 如果运行 LAN Base 许可证的特性集，那么设备不支持使用三层的 EtherChannel；
- 用户不能混合使用 Inspur 3850 交换机和 Inspur 6650 交换机来建立交换机堆栈。

EtherChannel 概述

EtherChannel 可以在交换机、路由器和服务器之间提供拥有容错功能的高速链路。用户可以使用 EtherChannel 增加配线柜与数据中心之间的带宽，用户也可以在网络中最有可能出现瓶颈的位置部署这项技术。EtherChannel 可以通过各个链路来对负载进行重分布，以便当一些链路出现故障时，能够自动恢复连接。如果链路出现了故障，EtherChannel 会将故障链路发来的流量重定向给信道中的剩余链路，而不会出现流量中断。
一条 EtherChannel 是由多条以太网链路捆绑成的一条逻辑链路。

图 60：典型的 EtherChannel 配置

Catalyst switch	Catalyst 交换机
10/100 Switched links	10/100 交换链路
10/100 Switched links	10/100 交换链路
Workstations	工作站
Workstations	工作站

EtherChannel 可以在交换机与另一台交换机或主机之间提供最大 8Gb/s（Gigabit EtherChannel）或 80Gb/s（10-Gigabit EtherChannel）的全双工带宽。
每条 EtherChannel 都可以由最多 8 条可兼容的 Ethernet 端口组成。
EtherChannel 的数量最多为 128 条。
LAN Base 特性集支持最多 24 条 EtherChannel。
每条 EtherChannel 中的所有端口都必须要么配置为二层端口，要么配置为三层端口。EtherChannel 三层端口都是由路由端口组成的。路由端口即为使用接口配置命令 **no switchport** 设置为三层模式的物理端口。要想了解具体信息，可以参考配置接口特征一章。

Ethernet 模式

用户可以将 EtherChannel 配置为下列模式之一：端口汇聚协议（PAgP）、链路汇聚控制协议（LACP）或 On。用户要将 EtherChannel 两端配置为同一种模式：

若将 EtherChannel 一端配置为 PAgP 或 LACAP 模式，系统就会与信道另一端进行协商，以判断哪些端口应该处于活动（active）状态。如果远端端口不能协商 EtherChannel，那么本地端口就会进入独立状态，并且继续向一条链路一样承载数据流量。端口的配置不会更改，但是这个端口不会参与 EtherChannel；

若将 EtherChannel 配置为 **on** 模式，那就不会发生协商。交换机会强制所有兼容端口在 EtherChannel 中处于活动状态。此时，信道的另一端也必须配置为 **on** 模式，否则就会出现丢包。

设备上的 EtherChannel

用户可以在一台设备、堆栈中的一台设备或者堆栈中的多台设备（称为交叉堆栈 EtherChannel）上创建一条 EtherChannel。

图 61：单交换机上的 EtherChannel

Swtich stack	交换机堆栈
Switch 1	交换机 1
Switch 2	交换机 2
Switch 3	交换机 3
Switch A	交换机 A
StackWise Plus port connections	StackWise Plus 端口连接

图 62：交叉堆栈 EtherChannel

Swtich stack	交换机堆栈
Switch 1	交换机 1
Switch 2	交换机 2
Switch 3	交换机 3
Switch A	交换机 A
StackWise Plus port connections	StackWise Plus 端口连接

EtherChannel 链路故障切换

如果 EtherChannel 中的链路出现了故障，之前通过故障链路传输的流量就会迁移到 EtherChannel 中剩余的链路进行传输。如果交换机上启用了 trap，那么在出现故障时，设备就会发送一条 trap 来标识交换机、EtherChannel 和故障链路。在 EtherChannel 中通过一条链路入站的广播和组播数据包不能通过 EtherChannel 中另一条链路返回。

Channel Group 与 Port-Channel 接口

EtherChannel 包含一个 channel group 和一个 port-channel 接口。在 channel group 中，物理端口会绑定到 port-channel 接口。应用到 port-channel 接口上的配置变更也会应用到 channel group 中的所有物理端口。

命令 **channel-group** 会将物理端口和 port-channel 接口进行绑定。每个 EtherChannel 都有一

个编号为 1 到 128 之间的 port-channel 逻辑接口。这个 port-channel 接口编号对应的是接口配置命令 **channel-group** 设置的编号。

图 63：物理端口、Channel Group 和 Port-Channel 接口之间的关系

Logical port-channel	逻辑 port-channel
Channel-group binding	Channel-group 绑定关系
Physical ports	物理端口

- 对于二层端口，用户可以使用接口配置命令 **channel-group** 来动态创建 port-channel 接口；

  用户也可以使用全局配置命令 **interface port-channel** *port-channel-number* 来手动创建 port-channel 接口，但接下来用户必须使用命令 **channel-group** *channel-group-number* 来给物理端口绑定逻辑接口。其中，*channel-group-number* 和 *port-channel-number* 可以使用同一个数值，也可以使用一个新的数。如果使用新的数值，那么命令 **channel-group** 就会动态创建出新的 port channel。
- 对于三层端口，用户应该使用全局配置命令 **interface port-channel** 加上接口配置命令 **no switchport** 来创建逻辑接口。接下来，用户可以使用接口配置命令 **channel-group** 来手动将一个接口分配给 EtherChannel。
- 对于三层端口，用户应该使用接口配置命令 **no switchport** 来将接口配置为三层接口。接下来，用户可以使用接口配置命令 **channel-group** 来手动将一个接口分配给 EtherChannel。

端口汇聚协议

端口汇聚协议（PAgP）是一个 Inspur 私有协议，这个协议只能在 Inspur 设备上，和获准支持 PAgP 的厂商设备上运行。PAgP 支持在以太网端口之间交换 PAgP 数据包，以动态创建 EtherChannel。

通过 PAgP，设备或设备堆栈可以学习到能够支持 PAgP 的对端身份，以及每个端口的功能。接下来，它就可以动态将（堆栈中一台设备上）配置类似的端口分组为一条逻辑链路（信道或汇聚端口）。配置类似的端口可以根据硬件、管理和端口参数的限制进行分类。例如，PAgP 会将拥有相同速率、双工模式、native VLAN、VLAN 范围和中继状态和类型的端口进行分组。在将链路分组进一个 EtherChannel 之后，PAgP 就会将这个组作为一个设备端口添加到生成树当中。

PAgP 模式

PAgP 模式指定了一个端口是否可以发送 PAgP 数据包，哪个端口发起 PAgP 协商，或者端口是否仅对接收到的 PAgP 数据包进行响应。

表 63：EtherChannel PAgP 模式

模式	**描述**
auto	让一个端口进入被动协商的模式。在这种模式下，端口只会响应它接收到的 PAgP 数据包，但不会发起 PAgP 数据包协商。这种设置可以将 PAgP 数据包的传输降至最低
desirable	让一个端口进入主动协商的模式。在这种模式下，端口会通过发送 PAgP 数据包来发起与其他端口的协商。当 EtherChannel 成员来自交换机堆栈中的不同交换机时，可以使用这种模式

交换机端口只会与配置在 **auto** 或 **desirable** 模式下的端口交换 PAgP 数据包。配置在 **on** 模式

下的端口不会交换 PAgP 数据包。

工作在 **auto** 和 **desirable** 模式下的端口会基于诸如端口速率这类的标准，来与对端进行协商，以建立 EtherChannel。对于二层 EtherChannel 来说，双方则会基于 trunk 状态和 VLAN 编号进行协商。

当两边的端口处于不同的 PAgP 模式下时，只要这些模式相互兼容，那么这两边的端口还是可以建立 EtherChannel。例如：

- 工作在 **desirable** 模式下的端口，可以与另一端工作在 **desirable** 或 **auto** 模式下的端口建立 EtherChannel；
- 工作在 **auto** 模式下的端口，可以与另一端工作在 **desirable** 模式下的端口建立 EtherChannel；

工作在 **auto** 模式下的端口，不能与另一端工作在 **auto** 模式下的端口建立 EtherChannel，因为这两个端口都不会发起 PAgP 协商。

静默模式

如果交换机连接到了一个支持 PAgP 的设备，用户可以使用关键字 **non-silent** 配置交换机端口，让其执行非静默操作。如果用户没有在配置 **auto** 或 **desirable** 模式时添加关键字 **non-silent**，交换机也会默认支持静默模式。

如果交换机连接的是不支持 PAgP 的设备，那么在配置了静默模式之后，交换机就几乎不会通过这个端口发送数据包。可以充当静默对端的设备包括文件服务器，或者不会生成流量的数据包分析器等等。在本例中，在连接静默设备的物理端口上运行 PAgP 会防止这个交换机端口进入操作状态。不过，静默设置可以让 PAgP 正常工作，可以让将这个端口划入 channel group，并且使用这个端口传输流量。

PAgP 学习方式与优先级

网络设备可以归类为 PAgP 物理学习设备，或者汇聚端口学习设备。如果一台设备通过物理端口学习地址，并且直接依据这些内容来传输流量，那么这台设备就是物理学习设备。如果一台设备通过汇聚（逻辑）端口学习地址，那么这台设备就是汇聚端口学习设备。在链路两端，学习方法必须采取相同的配置。

当设备和对端都是汇聚端口学习设备，那么它们就会通过这条逻辑 port-channel 来学习地址。设备会使用 EtherChannel 中的端口来向源发送数据包。通过汇聚端口进行学习的话，数据包具体是通过哪个物理端口学习到的就并不重要了。

当对端设备是物理学习设备，而本地设备则是汇聚端口学习设备，那么 PAgP 就无法执行自动检测。因此，用户必须在本地设备上手动设置学习方法，让设备通过物理端口学习地址。用户还必须将负载分发方式设置为基于源的分发，让任何给定的源 MAC 地址都在同一个物理端口上进行发送。

用户也可以在组中配置一个端口，让它执行所有的传输，然后使用其他端口进行热备份。如果选择的端口检测不到硬件信号，那么组中未使用的端口在几秒之内就可能会切换到操作状态。用户可以使用接口配置命令 **pagp port-priority** 来修改端口的优先级，让所选接口执行所有的数据传输。优先级越高，选择这个端口的可能性就越高。

注释： 即使用户通过 CLI 输入了命令 **physical-port**，设备还是支持仅通过汇聚端口来学习地址。命令 **pagp learn-method** 和命令 **pagp port-priority** 对于设备的硬件没有效果，但用户需要输入这些命令，来让设备与那些只能通过物理端口学习地址的设备（如 Inspur 1900 交换机）进行 PAgP 互操作。

当设备在链路上的对端是一台物理学习设备，我们推荐用户使用接口配置命令 **pagp learn-method physical-port** 来将这台设备配置为一个物理端口学习设备。用户可以使用全局配置命令 **port-channel load-balance src-mac** 来设置基于源 MAC 地址的负载分发方式。接下来，

设备就会使用 EtherChannel 中学习到这个源地址的端口来发送数据包。用户只应在这种情况下使用命令 **pagp learn-method**。

PAgP 与其他特性的互动

动态中继协议（DTP）和 Inspur 发现协议（CDP）会通过 EtherChannel 中的物理端口来发送和接收数据包。Trunk 端口会在编号最低的 VLAN 中发送和接收协议数据单元（PDU）。

在二层 EtherChannel 中，信道中第一个启用的端口会将自己的 MAC 地址提供给 EtherChannel。如果用户将这个端口从接口束中移除，那么接口束中剩余的接口就会将自己的 MAC 地址提供给 EtherChannel。对于三层 EtherChannel，一旦用户（通过全局配置命令 **interface port-channel**）将接口创建出来，活动设备就会分配 MAC 地址。

PAgP 只会从启动，且启用了（auto 或 desirable 模式的）PAgP 的端口上发送和接收 PAgP PDU。

链路汇聚控制协议

LACP 定义在 IEEE 802.3 当中，它可以让 Inspur 设备管理（符合 IEEE 802.3ad 协议的）设备之间的以太网信道。LACP 可以交换以太网端口之间的 LACP 数据包，来自动创建 EtherChannel。通过 LACP，设备或设备堆栈可以学习到能够支持 LACP 的对端身份，以及每个端口的功能。接下来，它就可以动态将配置类似的端口分组为一条逻辑链路（信道或汇聚端口）。配置类似的端口可以根据硬件、管理和端口参数的限制进行分类。例如，LACP 会将拥有相同速率、双工模式、native VLAN、VLAN 范围和中继状态和类型的端口进行分组。在将链路分组进一个 EtherChannel 之后，LACP 就会将这个组作为一个设备端口添加到生成树当中。

在 port channel 中，端口独立模式操作发生了变化。在默认情况下，通过 CSCtn96950，设备会启用独立模式。当设备无法从 LACP 对等体那里接收到响应消息时，port channel 中的端口就会进入暂缓（suspended）状态。

LACP 模式

LACP 模式指定了一个端口是可以发送 LACP 数据包，还是只能接收 LACP 数据包。

表 64：EtherChannel LACP 模式

模式	描述
active	让一个端口进入主动协商的模式。在这种模式下，端口会通过发送 LACP 数据包来发起与其他端口的协商。
passive	让一个端口进入被动协商的模式。在这种模式下，端口只会响应它接收到的 LACP 数据包，但不会发起 LACP 数据包协商。这种设置可以将 LACP 数据包的传输降至最低

工作在 **active** 和 **passive** 模式下的端口会基于诸如端口速率这类的标准，来与对端进行协商，以建立 EtherChannel。对于二层 EtherChannel 来说，双方则会基于 trunk 状态和 VLAN 编号进行协商。

当两边的端口处于不同的 LACP 模式下时，只要这些模式相互兼容，那么这两边的端口还是可以建立 EtherChannel。例如：

- 工作在 **active** 模式下的端口，可以与另一端工作在 **active** 或 **passive** 模式下的端口建立 EtherChannel；
- 工作在 **passive** 模式下的端口，不能与另一端同样工作在 **passive** 模式下的端口建立 EtherChannel，因为这两个端口都不会发起 PAgP 协商。

LACP 与链路冗余

用户可以通过 LACP port-channel min-link 和 LACP max-bundle 特性，来进一步改善 LACP port-channel 的操作、带宽可用性和链路冗余。

LACP port-channel min-link（最小链路）特性：

- 配置必须启用并绑定到 LACP port channel 的最少端口数量；
- 防止低带宽的 LACP 端口变为活动状态；
- 如果活动成员端口的数量太少，达不到所需的最小带宽，则让 LACP port channel 成为不活动状态。

LACP max-bundle（最大绑定）特性：

- 定义 LACP port channel 中绑定的端口数量上限；
- 支持包含更少绑定端口的热备份端口。例如，在包含 5 个端口的 LACP port channel 当中，用户可以将 max-bundle 设置为 3，将剩下两个 2 端口指定为热备份端口。

LACP 与其他特性的互动

DTP 和 CDP 会通过 EtherChannel 中的物理端口来发送和接收数据包。Trunk 端口会在编号最低的 VLAN 中发送和接收协议数据单元（PDU）。

在二层 EtherChannel 中，信道中第一个启用的端口会将自己的 MAC 地址提供给 EtherChannel。如果用户将这个端口从接口束中移除，那么接口束中剩余的接口就会将自己的 MAC 地址提供给 EtherChannel。对于三层 EtherChannel，一旦用户（通过全局配置命令 **interface port-channel**）将接口创建出来，活动设备就会分配 MAC 地址。

LACP 只会从启动，且启用了（auto 或 desirable 模式的）LACP 的端口上发送和接收 LACP PDU。

EtherChannel On 模式

EtherChannel 的 **on** 模式可以用来手动配置 EtherChannel。这种 **on** 模式可以强制一个端口不经过协商直接加入 EtherChannel。如果远端设备不支持 PAgP 或 LACP，那么 **on** 模式就会相当实用。在 **on** 模式下，只有当链路两端的设备上都配置了 **on** 模式时，双方才会建立一条可用的 EtherChannel。

对于配置在同一个 channel group 的 **on** 模式下的端口，它们必须拥有匹配的端口特征，包括速率和双工模式。不兼容的端口会进入暂缓（suspended）状态，即使用户将它们配置在 **on** 模式下也是如此。

注意： 在使用 **on** 模式时务虚小心。这是手动配置，EtherChannel 两端的端口必须拥有相同的配置。如果组配置有误，网络中就会出现丢包或生成树环路。

负载分担和转发方式

EtherChannel 会通过各个链路来分担流量，它会将数据帧中地址里面的二元组减少到一个值，来选择信道中的一条链路使用。用户可以从几种不同的负载分担模式中选择一种，包括基于 MAC 地址的负载分发、基于 IP 地址的负载分发、基于源地址的负载分发、基于目的地址的负载分发、或基于源和目的地址的负载分发。选择的模式会应用于设备上配置的所有 EtherChannel。

注释： 三层等价多路径（ECMP）负载分担可以基于源 IP 地址、目的 IP 地址、源端口、目的端口和四层协议执行负载分担。分片的数据包可以在两条不同的链路上，基于这些参数运行算法进行处理。其中一项参数发生变化都会影响负载分担。

用户会使用全局配置命令 **port-channel load-balance** 和 **port-channel load-balance extended** 来配置负载分担和转发方式。

相关主题

MAC 地址转发

如果采用源 MAC 地址转发的方式，那么当数据包被转发到 EtherChannel 时，它们会基于入站数据包的源 MAC 地址在信道的各个端口中进行分发。因此，对于实现负载分担，来自不同主机的数据包会使用信道中的不同端口，但是从同一个主机发来的数据包就会使用信道中的同一个端口进行转发。

如果采用目的 MAC 地址转发的方式，那么当数据包被转发到 EtherChannel 时，它们会基于入站数据包的目的主机的 MAC 地址在信道的各个端口中进行分发。因此，去往同一个目的的数据包会使用信道中的同一个端口，但去往不同目的的数据包就会通过信道中的不同端口进行转发。

如果采用源和目的 MAC 地址转发的方式，那么当数据包被转发到 EtherChannel 时，它们会基于入站数据包的源和目的主机 MAC 地址在信道的各个端口中进行分发。通过这种方法，设备会结合源 MAC 地址和目的 MAC 地址的方式来提供负载分发。如果用户不清楚在某台设备上更适合使用源 MAC 地址转发还是目的 MAC 地址转发，就可以使用这种转发方式。通过源和目的 MAC 地址转发，从主机 A 发送给主机 B、从主机 A 发送给主机 C 和从主机 C 发送给主机 B 的数据包，会使用信道中的不同端口进行转发。

IP 地址转发

如果采用基于源 IP 地址转发的方式，那么数据包就会基于入站数据包的源 IP 地址在信道的各个端口中进行分发。因此，对于实现负载分担，来自不同 IP 地址的数据包会使用信道中的不同端口，但是从同一个 IP 地址发来的数据包就会使用信道中的同一个端口进行转发。

如果采用基于目的 IP 地址转发的方式，那么数据包就会基于入站数据包的目的 IP 地址在信道的各个端口中进行分发。因此，从同一个源 IP 地址发往不同目的 IP 地址的数据包会使用信道中的不同端口，而从不同的源 IP 地址发往同一个目的 IP 地址的数据包就会通过信道中的同一个端口进行转发。

如果采用源和目的 IP 地址转发的方式，那么数据包会基于入站数据包的源和目的主机 IP 地址在信道的各个端口中进行分发。通过这种方法，设备会结合源 IP 地址和目的 IP 地址的方式来提供负载分发。如果用户不清楚在某台设备上更适合使用源 IP 地址转发还是目的 IP 地址转发，就可以使用这种转发方式。通过源和目的 IP 地址转发，从 IP 地址 A 发送给 IP 地址 B、从 IP 地址 A 发送给 IP 地址 C 和从 IP 地址 C 发送给 IP 地址 B 的数据包，会使用信道中的不同端口进行转发。

负载分担的优势

不同的负载分担方式拥有不同的优势。用户应该根据设备在网络中的位置，和需要进行负载分担的流量类型来选择负载分担的方式。

在下图中，EtherChannel 有 4 台工作站在与一台路由器进行通信。由于路由器是单 MAC 地址设备，因此在 EtherChannel 设备上实施基于源的转发，可以确保设备会使用所有通向路由器的可用带宽。这台路由器上配置了基于目的的转发，因为大量工作站可以确保流量可以通过路由器的 EtherChannel 进行平均地分发。

图 64：负载分发的转发方式

Switch with source-based forwarding enabled	启用了基于源转发的交换机
Cisco router with destination-based forwarding enabled	启用了基于目的转发的 Cisco 路由器

用户在配置时，应该采用能够尽可能利用网络资源的方式。例如，如果信道中的流量都是去往同一个 MAC 地址，那么使用目的 MAC 地址就会让设备使用信道中的相同链路，因此使用源 IP 地址获得的负载分担效果更好。

EtherChannel 与设备堆栈

如果堆栈成员中有参与 EtherChannel 的端口出现了故障，或者离开了堆栈，那么活动设备就会将故障的堆栈成员设备端口从 EtherChannel 中移除。而 EtherChannel 中剩余的端口（如果还有剩余端口的话）会继续提供连通性。

当一台设备被添加到当前的堆栈中时，这台新的设备会从活动设备那里接收到运行配置，并且使用 EtherChannel 相关的堆栈配置来更新自己的配置。堆栈成员也会接收到操作信息（处于 up 状态的端口列表和信道的成员）。

当两个相互配置了 EtherChannel 连接的堆栈进行融合时，就会出现自环。生成树会检测到这种情况，并且执行相应的操作。在获胜的设备堆栈上，所有 PAgP 或 LACP 配置都不受到影响，而在没有获胜的设备堆栈上，所有 PAgP 或 LACP 配置则会在堆栈重启后全部被删除。

设备堆栈与 PAgP

在使用 PAgP 时，如果主用设备出现了故障或者离开了堆栈，那么备份设备就会成为新的主用设备。除非 EtherChannel 带宽发生了变化，否则生成树就不会重新收敛。新的主用设备会同步堆栈成员的配置。在主用设备变更之后，PAgP 的配置不会受到影响，除非 EtherChannel 有端口位于老的主用设备上。

设备堆栈与 LACP

在使用 LACP 时，系统 ID 会使用主用设备的堆栈 MAC 地址。当主用设备出现了故障或者离开了堆栈，而备份设备成为新的主用设备时，LACP 系统 ID 并不会变化。在默认情况下，在主用设备变更之后，LACP 的配置不会受到影响。

默认的 EtherChannel 配置

下表描述了默认的 EtherChannel 配置。

表 65：默认的 EtherChannel 配置

特性	**默认设置**
Channel group	未分配
Port-channel 逻辑接口	未定义
PAgP 模式	无默认设置
PAgP 学习方式	所有端口上皆为汇聚端口学习
PAgP 优先级	所有端口皆为 128
LACP 模式	无默认设置
LACP 学习方式	所有端口上皆为汇聚端口学习
LACP 端口优先级	所有端口皆为 32768
LACP 系统优先级	32768
LACP 系统 ID	LACP 系统优先级和设备或堆栈的 MAC 地址
负载分担	设备上的负载分发会基于入站数据包的源 MAC 地址来执行

EtherChannel 配置指南

如果配置正确的话，有些 EtherChannel 端口会自动禁用或者避免网络环路及其他问题。用户

可以按照下面的指导方针来避免出现配置问题：

- 不要尝试在设备或设备堆栈上配置超过 128 个 EtherChannel；
- 配置 PAgP EtherChannel 时，最多添加 8 个同一种类型的以太网端口；
- 配置 LACP EtherChannel 时，最多添加 16 个同一种类型的以太网端口。其中最多可有 8 个端口处于活动状态，另外 8 个端口则处于备用模式；
- 将 EtherChannel 中的所有端口配置为相同的速率和双工模式；
- 启用 EtherChannel 中的所有端口。如果 EtherChannel 中有端口因为接口配置命令 shutdown 而被禁用，那么这会被视为是这条链路出现了故障，其流量会通过 EtherChannel 中剩余的端口进行传输；
- 在第一次创建出一个组时，所有端口都会按照针对添加到组中第一个端口的参数进行设置。如果修改其中一项参数的配置，那么也必须对组中所有端口的配置进行修改：
  - 允许的 VLAN 列表
  - 每个 VLAN 的生成树路径开销
  - 每个 VLAN 的生成树端口优先级
  - 生成树 PortFast 设置
- 不要将一个端口配置为多个 EtherChannel 组的成员端口；
- 不要将一个 EtherChannel 同时配置在 PAgP 和 LACP 模式下。运行 PAgP 和 LACP 的 EtherChannel 组可以在同一台设备上共存，也可以在堆栈中的不同设备上共存。其中每个 EtherChannel 组可以或运行 PAgP，或运行 LACP，但这两种模式是不能实现互操作的；
- 不要将一个安全端口配置为 EtherChannel 的一部分，反之亦然；
- 不要将一个 EtherChannel 中的活动成员端口，或者以后会成为活动成员的端口配置为 IEEE 802.1x 端口。如果在一个 EtherChannel 端口上启用 IEEE 802.1x，设备就会出现错误消息，而 IEEE 802.1x 也不会启用；
- 如果用户在设备接口上配置了 EtherChannel，要先把 EtherChannel 的配置从接口上删除，然后再在设备上使用全局配置命令 **dot1x system-auth-control** 在全局启用 IEEE 802.1x；
- 如果用户配置了交叉堆栈 EtherChannel，和设备堆栈分离，那就有可能引发环路和转发问题。

二层 EtherChannel 配置指南

用户在配置二层 EtherChannel 时，可以参考下面的配置指南：

- 给 EtherChannel 中的所有端口分配相同的 VLAN 或者将它们都配置为 trunk 端口。拥有不同 native VLAN 的端口无法组成 EtherChannel；
- 在中继（trunking）二层 EtherChannel 中，所有端口支持的 VLAN 范围都是相同的。如果支持的 VLAN 范围不同，这些端口就无法组成 EtherChannel，即使将 PAgP 设置为 **auto** 或 **desirable** 模式也是这样；
- 只要其他配置是兼容的，那么生成树路径开销不同的端口也可以组成 EtherChannel。设置不同的生成树路径开销本身并不会让端口无法组成 EtherChannel。

三层 EtherChannel 配置指南

用户在配置三层 EtherChannel 时，可以参考下面的配置指南：

- 对于三层 EtherChannel，要向 port-channel 逻辑接口分配 IP 地址，而不是给信道中的各个物理端口分配 IP 地址；

Auto-LAG

auto-LAG 特性可以让设备在连接一台交换机的端口上自动创建 EtherChannel。在默认情况下，

设备上是全局禁用 auto-LAG，但又在所有端口上启用这项特性的。当用户在全局启用 auto-LAG 时，交换机就会执行这项特性。

一旦在全局启用了 auto-LAG，有可能会出现下列情形：

- 所有端口都会参与创建 EtherChannel 的端口，前提是对端设备的端口上配置了 EtherChannel。要想了解详细信息，可以参考下表“*本地和对端设备上支持的 auto-LAG 配置*”；
- 已经是手动 EtherChannel 成员的端口不会再参与自动创建 EtherChannel；
- 如果在已经参与了自动创建 EtherChannel 的端口上禁用 auto-LAG，这个端口会从自动创建的 EtherChannel 中被解绑出来。

下表显示了本地设备和对端设备上支持的 auto-LAG 配置。

表 66：本地和对端设备上支持的 auto-LAG 配置

本地/对端设备	**Active**	**Passive**	**Auto**
Active	是	是	是
Passive	是	否	是
Auto	是	是	是

如果在全局禁用 auto-LAG，那么所有自动创建的 EtherChannel 都会成为手动的 EtherChannel。用户不能在当前的自动创建的 EtherChannel 中添加任何配置，要想这样做需要首先执行命令 **port-channel**<*channel-number*>**persistent**，将其转换为手动的 EtherChannel。

注释： Auto-LAG 会使用 LACP 协议来创建自动 EtherChannel。与每台对端设备之间只能自动创建一条 EtherChannel。

Auto-LAG 配置指南

用户在配置 Auto-LAG 特性时，可以参考下面的配置指南：

- 在全局和在端口上启用 auto-LAG 时，如果不希望端口成为自动 EtherChannel 的成员，那就在端口上禁用 auto-LAG；
- 如果一个端口已经是手动 EtherChannel 成员，那它就不会再参与自动创建 EtherChannel。如果希望将它捆绑到自动 EtherChannel 当中，首先需要将这个端口从手动 EtherChannel 中解绑；
- 在启用 auto-LAG 并且创建自动 EtherChannel 时，用户可以手动与同一台对端设备之间手动配置多条 EtherChannel 信道。但是在默认情况下，端口会尝试与对端设备创建自动 EtherChannel；
- auto-LAG 只支持二层 EtherChannel。三层接口和三层 EtherChannel 不支持 auto-LAG；
- 交叉堆栈 EtherChannel 支持 auto-LAG。

如何配置 EtherChannel

在配置了一条 EtherChannel 之后，对 port-channel 接口所作的配置变更会作用于这个 port-channel 接口中的所有物理端口，而针对物理端口所作的配置变更则只会作用于应用配置的那个端口。

配置二层 EtherChannel（CLI）

在配置二层 EtherChannel 时，用户可以使用接口配置命令 **channel-group** 来将端口划分到 channel group。这条命令会自动创建出 port-channel 逻辑接口。

总步骤

1. configure terminal

2. interface *interface-id*

3. switchport mode {**access** | **trunk**}

4. switchport access vlan *vlan-id*

5. channel-group *channel-group-number* **mode** {**auto** [**non-silent**] | **desirable** [**non-silent**] | **on** } | { **active** | **passive**}

6. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 2	**interface** *interface-id* 示例： `Device(config)# interface gigabitethernet2/0/1`	选择要配置的接口，并进入接口配置模式 有效接口为物理端口。 对于 PAgP EtherChannel，用户可以将最多 8 个相同类型和速率的端口配置为一个组； 对于 PAgP EtherChannel，用户可以将最多 16 个相同类型和速率的端口配置为一个组。其中最多 8 个活动端口，8 个处于备份模式
步骤 3	**switchport mode** {**access**	**trunk**} 示例： `Device(config-if)# switchport mode access`	将所有端口配置为同一个 VLAN 中的静态 access 端口，或者将它们配置为 trunk 端口。 如果将端口配置为静态 access 端口，那就只能给它分配一个 VLAN。VLAN 的取值范围是 1 到 4094
步骤 4	**switchport access vlan** *vlan-id* 示例： `Device(config-if)# switchport access vlan 22`	（可选）如果将端口配置为静态 access 端口，那就只能给它分配一个 VLAN。VLAN 的取值范围是 1 到 4094
步骤 5	**channel-group** *channel-group-number* **mode** {**auto** [**non-silent**]	**desirable** [**non-silent**]	**on** }	{ **active**	**passive**} 示例： `Device(config-if)# channel-group 5 mode auto`	将端口分配给一个 channel group，并设置 PAgP 或 LACP 模式。 对于 **mode** 部分，选择下列关键字之一： • **auto**：仅当检测到 PAgP 设备时启用 PAgP。这个关键字会让端口进入被动协商状态，在这种状态下，端口只会响应它接收到的 PAgP 数据包，但不会发起 PAgP 数据包协商。如果 EtherChannel 成员位于堆栈中的不同设备上，则不支持配置这个关键字； • **desirable**：无条件启用 PAgP。这个关键字会让端口进入主动协商状态，在这种状态下，端口

		会通过发送 PAgP 数据包来与对端端口发起协商。如果 EtherChannel 成员位于堆栈中的不同设备上，则不支持配置这个关键字； • **on**：不使用 PAgP 或 LACP，强制端口建立信道。在 **on** 模式下，只有连接的对端端口组也是 **on** 模式时，EtherChannel 才会建立起来； • **non-silent**：（可选）如果设备连接到一台启用了 PAgP 的设备时，如果这个设备端口的模式为 **auto** 或 **desirable**，可以将该端口配置为非静默操作。如果不设置 **non-silent**，设备会默认执行静默。静默设置适用于与文件服务器或数据包分析设备之间的连接。这种设置可以让 PAgP 实现操作、将端口关联到 channel group，并且使用这个端口来执行传输； • **active**：仅当检测到 LACP 设备时启用 LACP。这个关键字会让端口进入主动协商状态，在这种状态下，端口会通过发送 LACP 数据包来与对端端口发起协商； • **passive**：在端口上启用 LACP，并让端口进入被动协商状态，在这种状态下，端口只会响应它接收到的 LACP 数据包，但不会发起 LACP 数据包协商
步骤 6	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置三层 EtherChannel（CLI）

用户可以执行下面的步骤来将一个以太网端口分配给三层 EtherChannel。这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. no ip address

5. no switchport

6. channel-group *channel-group-number* **mode** { **auto** [**non-silent**] | **desirable** [**non-silent**] | **on** } | { **active** | **passive** }

7. end

具体步骤

	命令或操作	目的
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密码

	示例： `Device>` **enable**	
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/2**	选择要配置的接口，并进入接口配置模式 有效接口为物理端口。 对于 PAgP EtherChannel，用户可以将最多 8 个相同类型和速率的端口配置为一个组； 对于 PAgP EtherChannel，用户可以将最多 16 个相同类型和速率的端口配置为一个组。其中最多 8 个活动端口，8 个处于备份模式
步骤 4	**no ip address** 示例： `Device(config-if)#` **no ip address**	确保物理端口上没有分配 IP 地址
步骤 5	**no switchport** 示例： `Device(config-if)#` **no switchport**	让这个端口进入三层模式
步骤 6	**channel-group** *channel-group-number* **mode** {**auto** [**non-silent**] | **desirable** [**non-silent**] | **on** } | { **active** | **passive**} 示例： `Device(config-if)#` **channel-group 5 mode auto**	将端口分配给一个 channel group，并设置 PAgP 或 LACP 模式。 对于 **mode** 部分，选择下列关键字之一： • **auto**：仅当检测到 PAgP 设备时启用 PAgP。这个关键字会让端口进入被动协商状态，在这种状态下，端口只会响应它接收到的 PAgP 数据包，但不会发起 PAgP 数据包协商。如果 EtherChannel 成员位于堆栈中的不同设备上，则不支持配置这个关键字； • **desirable**：无条件启用 PAgP。这个关键字会让端口进入主动协商状态，在这种状态下，端口会通过发送 PAgP 数据包来与对端端口发起协商。如果 EtherChannel 成员位于堆栈中的不同设备上，则不支持配置这个关键字； • **on**：不使用 PAgP 或 LACP，强制端口建立信道。在 **on** 模式下，只有连接的对端端口组也是 **on** 模式时，EtherChannel 才会建立起来； • **non-silent**：（可选）如果设备连接到一台启用了 PAgP 的设备时，如果这个设备端口的模式为 **auto** 或 **desirable**，可以将该端口配置为非静默操作。如果不设置 **non-silent**，设备会默

		认执行静默。静默设置适用于与文件服务器或数据包分析设备之间的连接。这种设置可以让 PAgP 实现操作、将端口关联到 channel group，并且使用这个端口来执行传输； • **active**：仅当检测到 LACP 设备时启用 LACP。这个关键字会让端口进入主动协商状态，在这种状态下，端口会通过发送 LACP 数据包来与对端端口发起协商； • **passive**：在端口上启用 LACP，并让端口进入被动协商状态，在这种状态下，端口只会响应它接收到的 LACP 数据包，但不会发起 LACP 数据包协商
步骤 7	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置 EtherChannel 负载分担（CLI）

用户可以使用下面几种不同的转发方式之一，来配置 EtherChannel 负载分担。
这个流程是可选的。

总步骤

1. configure terminal

2. port-channel load-balance { **dst-ip** | **dst-mac** | **dst-mixed-ip-port** | **dst-port** | **extended** [**dst-ip** | **dst-mac** | **dst-port** | **ipv6-label** | **l3-proto** | **src-ip** | **src-mac** | **src-port**] | **src-dst-ip** | **src-dst-mac** **src-dst-mixed-ip-port src-dst-portsrc-ip** | **src-mac** | **src-mixed-ip-port** | **src-port**}

3. end

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**port-channel load-balance** { **dst-ip**	**dst-mac**	**dst-mixed-ip-port**	**dst-port**	**extended** [**dst-ip**	**dst-mac**	**dst-port**	**ipv6-label**	**l3-proto**	**src-ip**	**src-mac**	**src-port**]	**src-dst-ip**	**src-dst-mac**	配置 EtherChannel 负载分担方法。 默认设置为 **src-mac**。 选择下列负载分发方式之一： • **dst-ip**：设置目的主机 IP 地址； • **dst-mac**：设置入站数据包的目的主机 MAC 地址； • **dst-mixed-ip-port**：设置主机 IP 地址和 TCP/UDP 端口； • **dst-port**：设置目的 TCP/UDP 端口； • **extended**：设置扩展的负载分担方式——结合

	src-dst-mixed-ip-port src-dst-portsrc-ip | src-mac | src-mixed-ip-port | src-port} 示例： `Device(config)#` **port-channel load-balance src-mac**	源和目的的方式； • **ipv6-label**：设置 IPv6 流标签； • **l3-proto**：设置三层协议； • **src-dst-ip**：设置源和目的主机 IP 地址； • **src-dst-mac**：设置源和目的主机 MAC 地址； • **src-dst-mixed-ip-port**：设置源和目的主机 IP 地址和 TCP/UDP 端口； • **src-dst-port**：设置源和目的 TCP/UDP 端口； • **src-ip**：设置源主机 IP 地址； • **src-mac**：设置入站数据包的源 MAC 地址； • **src-mixed-ip-port**：设置源主机 IP 地址和 TCP/UDP 端口； • **src-port**：设置源 TCP/UDP 端口
步骤 3	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置 EtherChannel 扩展的负载分担（CLI）

在想要使用组合负载分担方式时，可以配置 EtherChannel 扩展的负载分担。
这项操作是可选的。

总步骤

1. configure terminal

2. port-channel load-balance extended [dst-ip | dst-mac dst-port | ipv6-label | l3-proto | src-ip | src-mac | src-port]

3. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**port-channel load-balance extended [dst-ip | dst-mac dst-port | ipv6-label | l3-proto | src-ip | src-mac | src-port]** 示例： `Device(config)#` **port-channel load-balance extended dst-ip dst-mac src-ip**	配置 EtherChannel 扩展的负载分担方法。 默认设置为 **src-mac**。 选择下列负载分发方式之一： • **dst-ip**：设置目的主机 IP 地址； • **dst-mac**：设置入站数据包的目的主机 MAC 地址； • **dst-port**：设置目的 TCP/UDP 端口； • **ipv6-label**：设置 IPv6 流标签； • **l3-proto**：设置三层协议； • **src-ip**：设置源主机 IP 地址；

		• **src-mac**：设置入站数据包的源 MAC 地址； • **src-port**：设置源 TCP/UDP 端口
步骤 3	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置 PAgP 学习方式与优先级（CLI）

这项操作是可选的。

总步骤

1. configure terminal

2. interface *interface-id*

3. pagp learn-method physical-port

4. pagp port-priority *priority*

5. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 2	**interface** *interface-id* 示例： `Device(config)# interface gigabitethernet 1/0/2`	选择用于传输流量的接口，并进入接口配置模式
步骤 3	**pagp learn-method physical-port** 示例： `Device(config-if)# pagp learn-method physical port`	选择 PAgP 学习方法。 在默认情况下，选择的是 **aggregation-port learning**，这表示设备会使用 EtherChannel 中的任意端口向源发送数据包。若使用汇聚端口学习，数据从哪个物理接口到达设备并不重要。 选择 **physical-port** 来连接另一台物理学习设备。一定要保证将全局配置命令 **port-channel load-balance** 配置为了 **src-mac**。 EtherChannel 两端必须配置相同的学习方法
步骤 4	**pagp port-priority** *priority* 示例： `Device(config-if)# pagp port-priority 200`	通过分配优先级来让设备使用用户所选的端口传输数据。 *priority* 的取值范围是从 1 到 255，默认值为 128。优先级越高，这个端口越有可能用来执行 PAgP 传输
步骤 5	**end**	返回特权 EXEC 模式

	示例： `Device(config)# end`	

配置 LACP 热备份端口

在启用了 LACP 时，软件默认会尝试配置一个信道中 LACP 兼容端口的最大数量，最大值为 16 个端口。只有 8 条 LACP 链路可以同时处于活动状态，剩下的 8 条链路则会处于热备份模式。如果活动链路之一的状态变为不活动，那么处于热备份模式的链路就会变为活动状态。用户可以设置一个信道中活动端口的最大数量，以此来覆盖默认的操作。此时，剩余端口就会成为热备份端口。例如，如果用户将信道中最大的端口数量设置为 5 个，那么就会有最多 11 个端口成为热备份端口。

如果用户给一条 EtherChannel 组中配置了多于 8 条链路，那么软件就会基于 LACP 优先级来自动决定将哪些热备份端口置于活动状态。对于 LACP 系统之间的每条链路，软件都会分配一个专门的优先级，优先级中最多会包含下面几项因素（下面因素按照优先级顺序排列）：

- LACP 系统优先级；
- 系统 ID（设备 MAC 地址）；
- LACP 端口优先级；
- 端口号

在比较优先级时，数值越低即表示优先级越高。当硬件限制防止所有兼容端口进行汇聚时，设备会使用优先级来决定应该将哪些端口置于备份模式。

判断哪些端口应该处于活动状态，哪些端口处于热备份状态，是一个两步的流程。首先，系统优先级和系统 ID 数值较低的系统会被列入考虑。接下来，系统会基于端口优先级和端口号的数值，来判断哪些端口应该处于活动状态，哪些端口则处于热备份状态。其他系统的端口优先级和端口号则不会使用。

用户可以修改 LACP 系统优先级和 LACP 端口优先级的默认值，来影响软件对活动链路和备份链路的选择。

配置 LACP Max Bundle 特性（CLI）

在设置一个 port channel 中可以捆绑的 LACP 端口最大数量时，port channel 中剩余的端口就会被指定为热备份端口。

用户可以从特权 EXEC 模式开始，按照下面的步骤来配置一个 port channel 中支持的 LACP 端口最大数量。这个流程是可选的。

总步骤

1. configure terminal

2. interface port-channel *channel-number*

3. lacp max-bundle *max-bundle-number*

4. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 2	**interface port-channel**	进入 port channel 的接口配置模式，取值范围是从

	channel-number 示例： `Device(config)#` **interface port-channel 2**	1 到 128
步骤 3	**lacp max-bundle** *max-bundle-number* 示例： `Device(config-if)#` **lacp max-bundle 3**	设置 port-channel 接口束中，最大的 LACP 端口数量。数量取值范围是从 1 到 8
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置 LACP Port-Channel 独立禁用

要在一个 port channel 上禁用独立 EtherChannel 成员端口状态，可以在 port channel 接口上执行下面的操作：

总步骤

1. configure terminal

2. interface port-channel *channel-group*

3. port-channel standalone-disable

4. end

5. show etherchannel

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**interface port-channel** *channel-group* 示例： `Device(config)#` **interface port-channel** *channel-group*	选择要配置的 port channel 接口
步骤 3	**port-channel standalone-disable** 示例： `Device(config-if)#` **port-channel standalone-disable**	在这个 port-channel 接口上禁用独立模式
步骤 4	**end**	返回特权 EXEC 模式

	示例： `Device(config)# end`	
步骤 5	**show etherchannel** 示例： `Device#` **show etherchannel** *channel-group* **port-channel** `Device#` **show etherchannel** *channel-group* **detail**	验证所作的配置

配置 LACP Port Channel Min-Link 特性（CLI）

用户可以设置必须处于链路 up 状态，并绑定到 LACP port channel 作为 EtherChannel 的最少端口数量，允许绑定了这个数量的 port channel 接口可以过渡到链路 up 状态。使用 EtherChannel min-link 特性，可以防止低带宽的 LACP 端口变为活动状态。Port channel min-links 也会在活动成员端口的数量太少，达不到所需的最小带宽时，让 LACP port channel 成为不活动状态。

要配置 port channel 所需的最少链路数量，可以执行下面的操作。

总步骤

1. enable

2. configure terminal

3. interface port-channel *channel-number*

4. port-channel min-links *min-links-number*

5. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface port-channel** *channel-number* 示例： `Device(config)#` **interface port-channel 2**	进入一个 port-channel 的接口配置模式。 *channel-number* 的取值范围是从 1 到 63
步骤 4	**port-channel min-links** *min-links-number* 示例： `Device(config-if)#` **port-channel min-links 3**	设置必须处于链路 up 状态，并绑定到 LACP port channel 作为 EtherChannel 的最少端口数量，允许绑定了这个数量的 port channel 接口可以过渡到链路 up 状态。 *min-links-number* 的取值范围是 2 到 8

步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置 LACP 系统优先级（CLI）

用户可以使用全局配置命令 **lacp system-priority**，给所有启用了 LACP 的 EtherChannel 配置系统优先级，但不能给每个配置了 LACP 的信道配置系统优先级。把这个数值修改为默认参数之外的值，可以影响软件选择哪些链路作为活动链路，选择哪些链路作为备份链路。

用户可以使用特权 EXEC 命令 **show etherchannel summary** 来查看哪些端口处于热备份模式（这类端口的端口状态标记为 H）。

用户可以使用下面的步骤来配置 LACP 系统优先级。这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. lacp system-priority *priority*

4. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**lacp system-priority** *priority* 示例： `Device(config)# lacp system-priority 32000`	配置 LACP 系统优先级。 这个参数的取值范围是 1 到 65535，默认值为 32768。 取值越低，系统优先级越高
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置 LACP 端口优先级（CLI）

在默认情况下，所有端口使用的是相同的端口优先级。如果本地系统的系统优先级和系统 ID 取值比远端系统低，那么用户可以将 LACP EtherChannel 的端口优先级修改为一个低于默认值的数值，来让这些热备链路首先成为活动链路。用户可以使用特权 EXEC 命令 **show etherchannel summary** 来查看哪些端口处于热备份模式（这类端口的端口状态标记为 H）。

注释： 如果 LACP 不能汇聚所有兼容的端口（比如，远端系统的硬件限制比本地系统更严格），那么所有不能主动包含在 EtherChannel 中的端口都会被置入热备份状态，只有在捆绑的端口出现故障时，才会使用这些端口。

用户可以使用下面的步骤来配置 LACP 系统优先级。这个流程是可选的。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. lacp port-priority *priority*

5. end

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/2**	选择要进行配置的端口，并进入接口配置模式
步骤 4	**lacp port-priority** *priority* 示例： `Device(config-if)#` **lacp port-priority 32000**	配置 LACP 系统优先级。 这个参数的取值范围是 1 到 65535，默认值为 32768。取值越低，这个端口越有可能用于数据传输
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

配置 LACP 快速计时器

用户可以修改 LACP 计时器速率，来修改 LACP 超时的时间周期。用户可以使用命令 **lacp rate** 来设置支持 LACP 的接口会以什么样的速率接收 LACP 控制数据包。用户可以将这个超时速率从默认速率（30 秒）修改为一个更快的速率（1 秒）。这条命令只能在启用了 LACP 的接口上输入。

总步骤

1. enable

2. configure terminal

3. interface {**fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port*

4. lacp rate {**normal** | **fast**}

5. end

6. show lacp internal

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** {**fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* 示例： `Device(config)# interface gigabitEthernet 2/1`	配置接口，并进入接口配置模式
步骤 4	**lacp rate {normal | fast}** 示例： `Device(config-if)# lacp rate fast`	设置支持 LACP 的接口会以什么样的速率接收 LACP 控制数据包。 • 要将超时速率重置为默认值，需要输入命令 **no lacp rate**
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式

步骤 6	**show lacp internal**
	
	示例：
	`Device# show lacp internal`
	`Device# show lacp counters`
		验证所作的配置

在全局配置 Auto-LAG

总步骤

1. enable

2. configure terminal

3. [**no**] **port-channel auto**

4. end

5. show etherchannel auto

具体步骤

	命令或操作	目的

步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	[**no**] **port-channel auto** 示例： `Device(config)#` **interface gigabitEthernet 2/1**	在一台交换机上全局启用 auto-LAG。使用这条命令的 no 形式在交换机上全局禁用 auto-LAG 特性。 **注释**：在默认情况下，端口上的 auto-LAG 特性是启用的
步骤 4	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show etherchannel auto** 示例： `Device#` **show etherchannel auto**	显示自动创建的 EtherChannel

在端口上配置 Auto-LAG

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. [**no**] **channel-group auto**

5. end

6. show etherchannel auto

具体步骤

	命令或操作	**目的**
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id*	选择要启用 auto-LAG 的端口，并进入接口配置模

	示例： `Device(config)#` **interface gigabitethernet 1/0/1**	式
步骤 4	**[no] port-channel auto** 示例： `Device(config)#` **interface gigabitEthernet 2/1**	（可选）在个别接口上启用 auto-LAG。使用这条命令的 no 形式可以在个别端口上禁用 auto-LAG 特性。 **注释：**在默认情况下，端口上的 auto-LAG 特性是启用的
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show etherchannel auto** 示例： `Device#` **show etherchannel auto**	显示自动创建的 EtherChannel

接下来做什么？

在配置 Auto-LAG 时配置持续功能

用户可以使用 persistence 命令将自动创建的 EtherChannel 转换为手动 EtherChannel，以便向当前的 EtherChannel 中添加配置。

总步骤

1. enable

2. port-channel *channel-number* **persistent**

3. show etherchannel summary

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device>` **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**port-channel** *channel-number* **persistent** 示例： `Device#` **port-channel 1 persistent**	将自动创建的 EtherChannel 转换为手动 EtherChannel，以便向当前的 EtherChannel 中添加配置
步骤 3	**show etherchannel summary** 示例：	显示 EtherChannel 的信息

	Device# **show etherchannel summary**	

监控 EtherChannel、PAgP 和 LACP 的状态

用户可以使用表中所示的命令来查看 EtherChannel、PAgP 和 LACP 的状态。

表 67：监控 EtherChannel、PAgP 和 LACP 状态的命令

命令	描述
clear lacp { *channel-group-number* **counters** | **counters** }	清除 LACP channel-group 信息和流量计数器
clear pagp { *channel-group-number* **counters** | **counters** }	清除 PAgP channel-group 信息的流量计数器
show etherchannel [*channel-group-number* { **detail** | **load-balance** | **port** | **port-channel** | **protocol** | **summary** }] [**detail** | **load-balance** | **port** | **port-channel** | **protocol** | **auto** | **summary**]	用简化的形式、详细的形式或者一行信息的形式，显示 EtherChannel 信息。同时显示负载分担和数据帧分发机制、端口、port-channel、协议和 Auto-LAG 的信息
show pagp [*channel-group-number*] { **counters** | **internal** | **neighbor** }	显示 PAgP 信息，譬如流量信息、内部 PAgP 信息和邻居信息
show pagp [*channel-group-number*] **dual-active**	显示双向活动检测状态
show lacp [*channel-group-number*] { **counters** | **internal** | **neighbor** | **sys-id**}	显示 LACP 信息，譬如流量信息、内部 PAgP 信息和邻居信息
show running-config	验证配置条目
show etherchannel load-balance	显示 port channel 中端口的负载分担或数据帧分发机制

EtherChannel 的配置示例

配置二层 EtherChannel：示例

这个示例显示了如何在堆栈中的一台设备上配置 EtherChannel。在这个示例中，两个端口被配置为了 VLAN 10 中的静态 access 端口，并且将它添加到了 PAgP 模式 **desirable** 的 channel 5 中。

```
Device# configure terminal
Device(config)# interface range gigabitethernet2/0/1 -2
Device(config-if-range)# switchport mode access
Device(config-if-range)# switchport access vlan 10
Device(config-if-range)# channel-group 5 mode desirable non-silent
Device(config-if-range)# end
```

这个示例显示了如何在堆栈中的一台设备上配置 EtherChannel。在这个示例中，两个端口被配置为了 VLAN 10 中的静态 access 端口，并且将它添加到了 LACP 模式 **active** 的 channel 5

中。

```
Device# configure terminal
Device(config)# interface range gigabitethernet2/0/1 -2
Device(config-if-range)# switchport mode access
Device(config-if-range)# switchport access vlan 10
Device(config-if-range)# channel-group 5 mode active
Device(config-if-range)# end
```

这个示例显示了如何配置交叉堆栈 EtherChannel。在这个示例中，用户使用了 LACP 被动模式，用户将堆栈成员 1 中的两个端口和堆栈成员 2 中的一个端口配置为了 VLAN 10 中的静态 access 端口，并且将它们添加到了 channel 5 当中：

```
Device# configure terminal
Device(config)# interface range gigabitethernet2/0/4 -5
Device(config-if-range)# switchport mode access
Device(config-if-range)# switchport access vlan 10
Device(config-if-range)# channel-group 5 mode passive
Device(config-if-range)# exit
Device(config)# interface gigabitethernet3/0/3
Device(config-if)# switchport mode access
Device(config-if)# switchport access vlan 10
Device(config-if)# channel-group 5 mode passive
Device(config-if)# exit
```

如果用户配置交换机上的两个端口，让它们连接接入点（AP），那么网络中有可能会出现 PoE 或 LACP 协商错误。如果 port channel 配置在交换机上，这个问题就可以避免。要想了解详细信息，可以参考下面的示例：

```
interface Port-channel1
switchport access vlan 20
switchport mode access
switchport nonegotiate
no port-channel standalone-disable <--this one
spanning-tree portfast
```

注释： 如果在端口翻动时，端口报告了 LACP 错误，用户也应该输入这条命令：**no errdisable detect cause pagp-flap**。

配置三层 EtherChannel：示例

这个示例显示了如何配置交叉堆栈三层 EtherChannel。在这个示例中，用户将两个端口添加到了 LACP 模式为 **active** 的 channel 5 当中：

```
Device# configure terminal
Device(config)# interface range gigabitethernet2/0/1 -2
Device(config-if-range)# no ip address
Device(config-if-range)# no switchport
Device(config-if-range)# channel-group 5 mode active
Device(config-if-range)# end
```

这个示例显示了如何配置交叉堆栈三层 EtherChannel。在这个示例中，用户将堆栈成员 2 中

的两个端口和堆栈成员 3 中的一个端口添加到了 LACP 模式为 **active** 的 channel 7 当中：

```
Device# configure terminal
Device(config)# interface range gigabitethernet2/0/4 -5
Device(config-if-range)# no ip address
Device(config-if-range)# no switchport
Device(config-if-range)# channel-group 7 mode active
Device(config-if-range)# exit
Device(config)# interface gigabitethernet3/0/3
Device(config-if)# no ip address
Device(config-if)# no switchport
Device(config-if)# channel-group 7 mode active
Device(config-if)# exit
```

配置 LACP 热备份端口：示例

这个示例显示了如何通过配置，让 EtherChannel（port channel 2）在 port channel 中至少有 3 个活动端口的前提下进入活动状态，这个示例中包含 7 个活动端口，剩余端口（最多有 9 个）则为热备份端口：

```
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# port-channel min-links 3
Device(config-if)# lacp max-bundle 7
```

这个示例显示了如何在 port channel 42 上禁用独立 EtherChannel 成员端口状态：

```
Device(config)# interface port-channel channel-group
Device(config-if)# port-channel standalone-disable
```

这个示例显示了如何验证前面所作的配置：

```
Device# show etherchannel 42 port-channel | include Standalone
Standalone Disable = enabled
Device# show etherchannel 42 detail | include Standalone
Standalone Disable = enabled
```

配置 Auto LAG：示例

这个示例显示了如何在一台交换机上配置 Auto-LAG。

```
device> enable
device# configure terminal
device (config)# port-channel auto
device (config-if)# end
device# show etherchannel auto
```

下面的示例显示了自动创建的 EtherChannel 汇总信息。

```
device# show etherchannel auto
Flags: D - down P - bundled in port-channel
I - stand-alone s - suspended
```

```
H - Hot-standby (LACP only)
R - Layer3 S - Layer2
U - in use f - failed to allocate aggregator
M - not in use, minimum links not met
u - unsuitable for bundling
w - waiting to be aggregated
d - default port
A - formed by Auto LAG
Number of channel-groups in use: 1
Number of aggregators: 1
Group Port-channel Protocol Ports
------+-------------+-----------+-----------------------------------------------
1 Po1(SUA) LACP Gi1/0/45(P) Gi2/0/21(P) Gi3/0/21(P)
```

下面的示例显示了执行命令 **port-channel** 1 **persistent** 之后，自动 EtherChannel 的汇总信息

```
device# port-channel 1 persistent
device# show etherchannel summary
Switch# show etherchannel summary
Flags: D - down P - bundled in port-channel
I - stand-alone s - suspended
H - Hot-standby (LACP only)
R - Layer3 S - Layer2
U - in use f - failed to allocate aggregator
M - not in use, minimum links not met
u - unsuitable for bundling
w - waiting to be aggregated
d - default port
A - formed by Auto LAG
Number of channel-groups in use: 1
Number of aggregators: 1
Group Port-channel Protocol Ports
------+-------------+-----------+-----------------------------------------------
1 Po1(SU) LACP Gi1/0/45(P) Gi2/0/21(P) Gi3/0/21(P)
```

其他关于 EtherChannel 的参考资料

相关文档

相关主题	文档名
二层命令参考	《第 2/3 层命令参考手册（Inspur 6650 交换机）》

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com/icnt

标准与 RFC

标准/RFC	标题
无	——

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com/icnt

EtherChannel 的特性信息

版本	修改
Inspur INOS 11.3.1	引入该特性

配置弹性以太网协议

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com/go/cfn。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

REP 概述

弹性以太网协议（REP）是 Inspur 私有的协议，它提供了一种替代生成树协议（STP）的机制，来控制网络环路，解决链路故障，改善收敛时间。REP 可以控制一组连接到同一个网段的端口，确保这个网段中不会出现任何桥接环路，并且对网段中的链路故障进行响应。REP 为搭建更加复杂的网络、支持 VLAN 负载分担提供了基础。

注释： 运行 IP Base 和 IP Services 的 Inspur 交换机都可以支持 REP，但运行 LAN Base 许可证的交换机是不支持 REP 的。

REP 网段是一系列彼此相连、且由用户配置了一个网段 ID 的端口。每个网段由多个标准（即非边缘）网段端口和两个用户配置的边缘端口组成。一台路由器上，不能有超过两个端口属于同一个网段，每个网段的端口只能有一个外部邻居。

一个网段可以通过一段共享媒介相连，但每条链路上都只能有两个端口属于同一个网段。只有 Trunk 以太网流点（EFP）接口支持 REF。

下图显示了由 6 个通过 4 台交换机相连的端口，其中端口 E1 和 E2 被配置为了边缘端口。当所有端口均处于操作状态时（如左图所示），有一个端口会被阻塞，如图中斜线所示。当网络中出现故障时，阻塞的端口就会回到转发状态，以减少网络中断的时间。

图 65：REP 断开网段

Edge port	边缘端口
Blocked port	阻塞端口
Link failure	链路故障

图中所示网段是一个断开的网段。两个边缘端口之间是不连通的。REP 网段无法产生桥接环路，因此用户可以安全地将网段边缘连接到任何网络。网段中所有连接路由器的主机都可以通过两个边缘端口连接到网络的其余部分，但每次只能通过其中一条链路访问网络。如果任何网段出现了故障，或者 REP 网段中的任何端口出现了故障，REP 都会开放所有端口，来确保流量可以通过其他网关建立连接。

下图显示的是一个环形网段，其两个边缘端口位于同一台路由器上。如果采用这种配置方式，那么用户可以在这个网段中的任意两台路由器上创建一条冗余连接。

图 66：REP 环形网段

REP 网段包含了如下特征：

- 如果一个网段中的所有端口都处于操作状态，那么有一个端口（这个端口称为*替代*端口）应该对于各个 VLAN 均处于阻塞状态。如果用户配置了负载分担，那么网段中的两个端口会控制 VLAN 的阻塞状态；
- 如果一个网段中有一个或多个端口没有处于操作状态，那么所有端口都会在所有 VLAN 中转发流量，这样才能确保网络的连通性；
- 如果链路出现了故障，那么替代端口立刻就会开放。当故障链路启动时，网络会给每个 VLAN 选出一个逻辑阻塞的端口，以确保网络中断时间最小。

用户可以基于 REP 网段，构建出几乎所有类型的网络。REP 也支持 VLAN 负载分担，这是由主用边缘端口进行控制的，负载分担可以在网段中的任何端口执行。

在接入环形拓扑中，相邻交换机可能不支持 REP，如下图所示。在这个环境中，用户可以将那些不面向 REP 的端口（E1 和 E2）配置为非邻居端口。这些端口会继承所有边缘端口的属性，用户可以按照配置边缘端口的方式来配置这些端口，包括让它们向汇聚交换机发送 STP 或 REP 拓扑变更通告。在本例中，发送的 STP 拓扑变更通告（TCN）是多生成树（MST）STP 消息。

图 67：边缘非邻居端口

REP not supported	不支持 REP
E1 and E2 are configured as edge no-neighbor ports	E1 和 E2 被配置为了边缘非邻居端口
REP ports	REP 端口

REP 拥有下列限制条件：

- 用户必须配置每一个网段端口；配置不当会导致网络中产生转发环路；
- 在网段中，REP 只能管理一个端口出现故障的情形；如果 REP 网段中有多个端口出现故障，网络有可能会失去连接；
- 用户只应该在包含冗余的网络中配置 REP。如果在没有冗余的网络中配置 REP，网络也有可能会失去连接；

链路完整性

REP 不会在边缘端口之间使用端到端的轮询功能来验证链路的完整性。REP 会实施本地链路失效检测。REP 链路状态层（LSL）会检测出可以感知 REP 的邻居，并在网段内建立连通性。一个接口支持的所有 VLAN 都会被阻塞，直至它检测出邻居为止。在找到邻居之后，REP 会判断哪些邻居端口应该成为替代端口，哪些端口则应该转发流量。

一个网段中的每个端口都有一个独立的端口 ID。端口 ID 的格式类似于生成树算法中使用的端口 ID 格式：一个端口号（在整个网桥上是唯一的），一个对应的 MAC 地址（在网络中是唯一的）。当一个网段端口启动时，其 LSL 会开始发送数据包，数据包中会包含网段 ID 和端口 ID。在与同一个网段的邻居执行了三次握手之后，这个端口就会被宣告为可操作状态。

在出现下列情况时，网段端口不会进入可操作状态：

- 没有邻居拥有相同的网段 ID；
- 超过一个邻居拥有相同的网段 ID；
- 邻居没有确认本地端口是对等体；

每个端口都会与直接的邻居之间建立邻接关系。一旦创建出邻居的连接关系，端口就会进行协商，来判断这个网段中要阻塞的端口，也就是替代端口。所有其他端口则不会被阻塞。在默认情况下，REP 数据包都会被发送给 BPDU 类 MAC 地址。这些数据包也可以发送给 Inspur 组播地址，这个地址是专门用来在这个网段出现故障时，接收未阻塞端口通告（BPA）消息的。没有运行 REP 的设备则会丢弃这些数据包。

快速收敛

REP 会以物理链路为单位，而不是基于 VLAN 来运行的。只有一个 hello 消息是每个 VLAN 都需要的，这种做法可以减少协议的负载。我们推荐用户在给定网段的所有交换机上创建相同的 VLAN，并且在 REP trunk 端口上配置相同的许可 VLAN。为了避免由软件中继消息引入的延迟，REP 也允许将一些数据包泛洪到一个常规的组播地址。这些消息会在硬件泛洪层（HFL）中进行处理，并且会被泛洪到整个网络，而不仅仅是在 REP 网段中进行泛洪。那些不属于这个网段的交换机会按照数据流量的方式来处理这些数据包。用户可以给整个域或者给某个网段配置一个管理 VLAN，以控制对这些消息的泛洪。

对于最多 1000 个 MAC 地址，5 个 VLAN 的环境中，估计的收敛恢复时间在 150 毫秒到 500 毫秒之间。对于最多 100 个组，5 个 VLAN 的环境中，组播流量估计的收敛恢复时间在 300

毫秒到 500 毫秒之间。

VLAN 负载分担

在 REP 网段中，有一个边缘端口会充当主用边缘端口，其他边缘端口则会充当辅助边缘端口。主用边缘端口总是会在这个网段中参与 VLAN 负载分担。VLAN 负载分担的实现方式是，在一个配置的替代端口上阻塞一些 VLAN，在主用边缘端口上阻塞其他的 VLAN。在配置 VLAN 负载分担时，用户可以通过下列三种方法之一来设置替代端口：

- 输入接口的端口 ID。要查看网段中一个端口的端口 ID，可以针对该端口输入接口配置命令 **show interface rep detail**；
- 输入网段中端口的邻居偏移编号（neighbor offset number），这个编号标识的是边缘端口的下游邻居端口。邻居偏移编号范围是从-256 到+256，取值 0 是非法的。主用边缘端口的偏移值为 1；大于 1 的正数标识的是主用边缘端口的下游邻居。复数表示的则是辅助边缘端口（偏移值为-1）及其下游邻居。

注释： 用户可以标识从主用（或辅助）边缘端口出发，到端口下游的位置，通过这种方式在主用边缘端口上配置偏移值。用户永远不能将偏移值设置为 1，因为这是主用边缘端口自身的偏移值。

下图显示了一个网段的邻居偏移值，其中 E1 为主用边缘端口，E2 为辅助边缘端口。圆环中的红色数字是从主用边缘端口的偏移值；圆环外的黑色数字是从辅助边缘端口的偏移值。注意，用户可以使用一个正偏移值（从主用边缘端口出发的下游位置）或负偏移值（从辅助边缘端口出发的下游位置）来标识（除主用边缘端口外的）所有端口。如果 E2 成为了主用边缘端口，那么它的偏移值就应该是 1，而 E1 则应该是-1。

- 使用接口配置命令 **rep segment** *segment-id* **preferred** 中的关键字 **preferred**，将之前配置的端口选为替代端口。

图 68：一个网段中的邻居偏移值

E1=Primary edge port	E1=主用边缘端口
E2=Secondary edge port	E2=辅助边缘端口
Offset number from the primary edge port	从主用边缘端口出发的偏移值
Offset number from the secondary edge port (negative numbers)	从辅助边缘端口触发的偏移值（负数）

在 REP 网段完成时，所有 VLAN 都会被阻塞。在配置 VLAN 负载分担时，用户也必须配置下面两种触发方式之一：

- 在拥有主用边缘端口的交换机上，进入特权 EXEC 模式，输入命令 **rep preempt segment** *segment-id* 手动触发 VLAN 负载分担；
- 输入接口配置命令 **rep preempt delay** *seconds* 来配置抢占延迟值。在链路故障并且恢复之后，只要用户配置的抢占时间周期到期，VLAN 负载分担就会开始执行。要注意，如果时间结束之前，另一个端口又发生了故障，延迟计时器会重新启动。

注释： 在配置了 VLAN 负载分担之后，如果用户不进行手动干预，又没有链路故障或者恢复，那么 VLAN 负载分担就不会启动。

在触发了 VLAN 负载分担之后，主用边缘端口会发出一条消息，来通告网段中所有接口关于抢占的信息。当辅助端口接收到了消息时，它会将消息发送到网络，以通告替代端口阻塞消息中指定的那些 VLAN，并且通告主用边缘端口阻塞剩余的 VLAN。

用户也可以在网段中配置一个端口，以阻塞所有的 VLAN。只有主用边缘端口会发起 VLAN 负

载分担，如果网段不是每一段皆为边缘端口，负载分担就无法实现。主用边缘端口决定了本地 VLAN 负载分担的配置。
用户需要重新配置主用边缘端口，以重置配置负载分担。在修改负载分担的配置时，主用边缘端口会等待用户输入命令 **rep preempt segment**，或者在端口故障并恢复后，等待用户配置的抢占延迟周期过去之后，再执行新的配置。如果将一个边缘端口修改为了一个常规的网段端口，当前的 VLAN 负载分担状态并不会变化。配置新的边缘端口可能会导致新的拓扑配置。

与生成树的互动

REP 不会与 STP 或 Flex Link 特性进行互动，但这些技术可以共存。属于一个网段的端口会从生成树控制中移除，网段端口不接受也不发送 STP BPDU。因此，STP 不能在网段中运行。
要从 STP 环形配置迁移到 REP 网段配置中，要首先在网段中的环形拓扑中配置一个端口，然后继续配置相邻的端口，以减少网段的数量。每个网段中总是会包含一个阻塞的端口，因此多个网段也就表示有多个阻塞端口，和更多失去连通性的可能。当网段经过配置，可以从双向通往边缘端口的位置时，用户就可以配置边缘端口了。

REP 端口

REP 网段包含了故障（Failed）端口、开放（Open）端口和替代（Alternate）端口：

- 配置为常规网段端口的端口，在刚刚启动时为故障端口；
- 在确定了邻居的邻接关系之后，端口就会过渡到替代端口状态，这个接口会阻塞所有的 VLAN。被阻塞的端口会进行协商，当网段确定下来时，一个阻塞的端口会继续扮演替代角色，而其他端口则会成为开放端口；
- 当链路上出现故障时，所有端口都会进入故障状态。当替代端口接收到故障通告时，它就会进入开放状态，转发所有 VLAN 的数据。

当一个常规的网段端口转换为了边缘端口，或者当一个边缘端口转换为了常规的网段端口，拓扑未必会发生变更。如果将一个边缘端口转换为常规的网段端口，那么除非用户配置了 VLAN 负载分担，否则设备是不会执行 VLAN 负载分担的。对于 VLAN 负载分担而言，用户必须在网段中配置两个边缘端口。
重新配置为生成树端口的网段端口会依据生成树的配置来重新启动。在默认情况下，有一个指定的阻塞端口。如果用户配置了 PortFast，或者禁用了 STP，那么这个端口就会进入转发状态。

如何配置 REP

网段是一系列彼此相连、且由用户配置了一个网段 ID 的端口。要配置 REP 网段，用户需要配置 REP 管理 VLAN（或者直接使用默认 VLAN 1），然后在接口配置模式中将端口添加到网段中。用户应该在网段中配置两个边缘端口，其中一个为主用边缘端口，另一个默认为辅助边缘端口。一个网段只能有一个主用边缘端口。如果用户将一个网段中的两个端口（例如两个不同交换机上的端口）配置为了主用边缘端口，REP 会选择其中之一来充当这个网段的主用边缘端口。用户也可以配置向哪里发送网段拓扑变更通告（STCN）和 VLAN 负载分担。

默认的 REP 配置

在所有接口上，REP 都是禁用的。在启用之后，这个接口就会成为一个常规的网段端口，除非用户将其配置为了边缘端口。
在启用了 REP 之后，发送网段拓扑变更通告（STCN）的操作是禁用的，所有 VLAN 都是阻塞状态，管理 VLAN 为 VLAN 1，
当用户启用 VLAN 负载分担之后，设备默认操作是等待用户手动执行抢占，而延迟计时器则是被禁用的。如果用户没有配置 VLAN 负载分担，那么在手动抢占之后，默认的操作是在主用边缘端口上阻塞所有 VLAN。

REP 配置指导方针

用户可以按照下面的配置方针来配置 REP：

- 我们推荐用户首先配置一个端口，然后继续配置相邻的端口，以减少网段和阻塞端口的数量；
- 如果一个网段中有两个以上端口出现了故障，而又没有配置外部邻居，那么其中一个端口就会针对这条数据路径进入转发状态，以便在配置期间维系网络的连通性。在命令 **show rep interface** 的输出信息中，这个端口的 Port Role 会显示为“Fail Logical Open”，而其他故障端口的 Port Role 则会显示为“Fail No Exit Neighbor”。当用户配置故障端口的外部邻居时，端口会经历替代端口状态的过渡流程，并且最终进入开放状态，或者停留在替代端口，具体哪种操作方式取决于替代端口的选择机制；
- REP 端口必须是二层 IEEE 802.1Q 或 Trunk 端口；
- 我们推荐用户给网段中的所有 trunk 端口都配置相同的允许 VLAN；
- 在通过 Telnet 连接配置 REP 时，请一定保持警惕。因为 REP 会阻塞所有 VLAN，直到有另一个 REP 接口发送消息来开放这个端口为止。如果用户通过 Telnet 会话从同一个接口访问路由器的话，那么在 Telnet 会话中启用 REP 的操作可能会导致连接断开；
- 用户不能在同一个网段或接口上同时运行 REP 和 STP，也不能同时运行 REP 和 Flex Links；
- 如果用户将 STP 网络连接到一个 REP 网段，一定要确保连接在网段一侧。不在边缘的 STP 连接可能会导致桥接环路，因为 STP 不会在 REP 网段上运行。REP 接口会丢弃所有 STP BPDU；
- 用户必须给网段中的所有 trunk 端口都配置相同的允许 VLAN，否则即属误配；
- 如果一台交换机上的两个端口都启用了 REP，那么这两个端口必须或同为常规的网段端口，或同为边缘端口。REP 端口会遵从下面这些规则：
  - 一台交换机上可以配置的 REP 端口数量是没有限制的。不过，一台交换机上只有两个端口可以属于同一个 REP 网段；
  - 如果在一台交换机上，用户只配置了一个端口，那么这个端口就应该是边缘端口；
  - 如果一台交换机上的两个端口同属一个网段，那么这两个端口必须或同为常规的网段端口，或同为边缘端口，或一个为常规的端口，另一个是边缘非邻居端口。一台交换机上的边缘端口和常规的网段端口不能属于同一个网段；
  - 如果一台交换机上的两个端口同属一个网段，用户将其中一个配置为了边缘端口，将另一个配置为了常规的网段端口（这种做法属于误配），那么交换机会按照常规的网段端口来操作这个边缘端口；
- REP 接口启动时即会处于阻塞状态，并且在确定开放这些接口的状态是安全的操作之前，

这些接口就会一直停留在阻塞状态。用户应该了解这种情况，避免邻接突然断开；

- REP 会在 native VLAN 中以未标记数据帧的形式来发送所有的 LSL PDU。发送给 Inspur 组播地址的 BPA 消息会被发送给管理 VLAN，这个 VLAN 默认为 VLAN 1；
- 用户可以配置 REP 接口在没有从邻居那里接收到 hello 消息多久之后，还会保持 up 状态。用户可以使用接口配置命令 **rep lsl-age-timer** 将这个时间设置为从 120 毫秒到 1000 毫秒之间的值。接下来，LSL hello 计时器就会设置为老化计时器值的时间除以 3。在常规操作中，在对等体交换机上的老化计时器超时，并且查看 hello 消息之前，设备会发送 3 条 LSL hello 消息；
  - EtherChannel port channel 接口不支持用户将 LSL 老化计时器设置为小于 1000 毫秒的值。如果尝试在 port channel 接口上配置小于 1000 毫秒的值，用户就会看到一条错误消息，输入的命令也会被拒绝；
- REP 端口不能配置为下列端口类型：
  - 交换端口分析器（SPAN）目的端口
  - 隧道端口
  - Access 端口
- EtherChannel 上支持 REP，但 EtherChannel 中捆绑的各个端口不支持 REP；
- 每台交换机上最多可以配置 64 个 REP 网段。

配置 REP 管理 VLAN

为了避免由软件因在负载分担期间中继 VLAN 阻塞通告消息而引入的延迟，REP 也允许将一些数据包泛洪到硬件泛洪层（HFL）中的一个常规组播地址。这些消息会被泛洪到整个网络，而不仅仅是在 REP 网段中进行泛洪。用户可以给整个域或者给某个网段配置一个管理 VLAN，以控制对这些消息的泛洪。

在配置 REP 管理 VLAN 时，用户应该遵照下面的指导方针：

- 如果没有配置管理 VLAN，设备默认会使用 VLAN 1；
- 我们可以在交换机上给所有网段配置一个管理 VLAN，也可以给各个网段分别配置管理 VLAN；
- 管理 VLAN 不能是 RSPAN VLAN。

要配置 REP 管理 VLAN，用户应该从特权 EXEC 模式开始，按照下面的步骤进行配置：

总步骤

1. configure terminal

2. rep admin vlan *vlan-id* **segment** *segment-id*

3. end

4. show interface [*interface-id*] **rep detail**

5. copy running-config startup config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**rep admin vlan** *vlan-id*	设置管理 VLAN。管理 VLAN 的取值范围是从 2 到

	segment *segment-id* 示例： `Device(config)#` **rep admin vlan 2 segment 2**	4094，默认值为 VLAN 1。 要给各个网段分别设置管理 VLAN，可以输入全局配置模式命令 **rep admin vlan** *vlan-id* **segment** *segment-id*。 要将管理 VLAN 设置为 1，可以输入全局配置命令 **no rep admin vlan**
步骤 3	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 4	**show interface** [*interface-id*] **rep detail** 示例： `Device#` **show interface gigabitethernet1/1 rep detail**	在其中一个 REP 接口上验证所作的配置
步骤 5	**copy running-config startup-config** 示例： `Device#` **copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

配置 REP 接口

要执行 REP 操作，用户必须在每个网段接口上启用 REP，并且指定网段 ID。这项操作是必须执行的，而且必须在执行其他 REP 配置之前完成。此外，用户还必须在每个网段上配置主用边缘端口和辅助边缘端口。所有其他步骤都是可选的。

用户需要按照下面的步骤在接口上启用和配置 REP。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. switchport mode trunk

5. rep segment *segment-id* [**edge** [**no-neighbor**] [[**primary**]] [**preferred**]

6. rep stcn {**interface** *interface id* | **segment** *id-list* | **stp**}

7. rep block port {**id** *port-id* | *neighbor-offset* | **preferred**} **vlan** {*vlan-list* | **all**}

8. rep preempt delay *seconds*

9. rep lsl-age-timer *value*

10. end

11. show interface [*interface-id*] **rep** [**detail**]

12. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id*	选择接口，进入接口配置模式。这个接口可以是物理二层接口或者 port channel（逻辑接口），port-channel 的取值范围是从 1 到 48
步骤 4	**switchport mode trunk**	将这个接口配置为二层 trunk 端口
步骤 5	**rep segment** *segment-id* [**edge** [**no-neighbor**] [[**primary**]] [**preferred**]	在接口上启用 REP，并标识一个网段号。网段 ID 的取值范围是从 1 到 1024。这条命令可以使用下面这些可选的关键字： **注释**：用户必须配置两个边缘端口，包括给每个网段配置一个主用边缘端口。 （可选）**edge**：将这个端口配置为边缘端口。每个网段只能有两个边缘端口。若输入 **edge** 时不带关键字 **primary**，会将这个端口配置为辅助边缘端口； （可选）**primary**：将这个端口配置为主用边缘端口，用户可以在这个端口上配置 VLAN 负载分担； （可选）**no-neighbor**：将没有外部 REP 邻居的端口配置为一个边缘端口。这个端口会集成所有边缘端口的属性，用户可以对它们执行与边缘端口相同的配置； **注释**：虽然每个网段只能有一个主用边缘端口，如果用户在两台不同的交换机上配置了边缘端口，并且在两台交换机上都配置了 **primary** 这个关键字，那么用户所作的配置是有效的。不过，REP 只会选择其中一个端口作为这个网段的主用边缘端口。用户可以输入特权 EXEC 命令 **show rep topology** 来查看网段的主用边缘端口。 （可选）**preferred**：表示这个端口会优选为替代端口或者优选为执行 VLAN 负载分担的端口 **注释**：将一个端口配置为 preferred，并不能保证这个端口成为替代端口，他只能让这个端口占据一些优势。替代端口往往是之前发生了故障的端口
步骤 6	**rep stcn** {**interface** *interface id* | **segment** *id-list* | **stp**}	（可选）配置边缘端口，让其发送网段拓扑变更通告（STCN）。 • **interface** *interface –id*：指定接收 STCN 的物理接口或 port channel； • **segment** *id-list*：指定接收 STCN 的一个或多个网段。网段的取值范围是从 1 到 1024； • **stp**：向 STP 网络发送 STCN **注释**：在边缘非邻居节点上，如果用户配置了 rep

		stcn stp 以便向 STP 网络发送 STCN，那么生成树模式就必须为 MST
步骤 7	**rep block port** {**id** *port-id* | *neighbor-offset* | **preferred**} **vlan** {*vlan-list* | **all**}	（可选）在主用边缘端口上配置 VLAN 负载分担，用下面三种方式之一来标识 REP 替代端口，并且在替代端口上配置要阻塞的 VLAN。 • **id***port-id*：用端口 ID 标识替代端口。设备会自动给网段中的每个端口创建出端口 ID。用户可以输入特权 EXEC 命令 **show interface** *type number* **rep** [**detail**]来查看接口的端口 ID； • *neighbor_offset*：这个数字的作用是将替代端口标识为从边缘端口出发的下游邻居。这个参数的取值范围是从-256 到 256，负数表示从辅助边缘端口出发的下游邻居。0 是无效取值。输入**-1** 则是将辅助边缘端口标识为替代端口。用户可以参照图 68：一个网段中的邻居偏移值，查看邻居偏移取值的示例。 **注释**：由于用户会在主用边缘端口（偏移值为 1）输入这条命令，因此用户不能输入偏移值 1 来标识替代端口。 • **preferred**：将之前标识为执行 VLAN 负载分担的那个优选替代端口，选为常规的网段端口； • **vlan** *vlan-list*：阻塞一个 VLAN 或者阻塞一个 VLAN 范围； • **vlan all**：阻塞所有 VLAN。 **注释**：用户只能在 REP 主用边缘端口上输入这条命令
步骤 8	**rep preempt delay** *seconds*	（可选）配置抢占时间延迟。 • 如果用户希望在链路失败和恢复之后，VLAN 负载分担能够自动触发，那就应该配置这条命令； • 时间延迟值的取值范围在 15 到 300 秒之间。默认值为用户手动触发抢占，没有时间延迟。 **注释**：用户只能在 REP 主用边缘端口上输入这条命令
步骤 9	**rep lsl-age-timer value**	（可选）配置 REP 接口在没有从邻居那里接收到 hello 之后，依然保持 up 状态的时间（单位为毫秒）。 这个参数的取值范围是 120 到 10000 毫秒，以 40 毫秒为配置的增量。默认值为 5000 毫秒（即 5 秒）。 **注释**：EtherChannel 的 port channel 不支持将 LSL 老化计时器的值设置为少于 1000 毫秒； 链路两端的端口应该配置相同的 LSL 老化值，以避免链路翻动的情况
步　骤	**end**	返回特权 EXEC 模式

10		
步 骤 11	**show interface** [*interface-id*] **rep** [**detail**]	（可选）查看 REP 接口的配置
步 骤 12	**copy running-config startup-config**	（可选）将输入的条目保存到配置文件中

设置手动抢占 VLAN 负载分担

如果用户没有在主用边缘端口上输入接口配置命令 **rep preempt delay***seconds* 来配置抢占时间延迟，那默认需要由用户在这个网段上手动触发 VLAN 负载分担。要确保在手动抢占 VLAN 负载分担之前，所有其他网段都已经完成了配置。在输入命令 **rep preempt delay segment** *segment-id* 后，设备会在执行命令前首先弹出一条确认消息，因为抢占操作可能会导致网络出现中断。

总步骤

1. rep preempt segment *segment-id*

2. show rep topology segment *segment-id*

具体步骤

	命令或操作	目的
步骤 1	**rep preempt segment** *segment-id*	在这个网段手动触发 VLAN 负载分担。 用户在执行这条命令之前，需要首先进行确认
步骤 2	**show rep topology segment** *segment-id*	显示 REP 的拓扑信息

给 REP 配置 SNMP Trap

用户可以通过配置路由器，让它发送 REP 特定的 trap 来向 SNMP（简单网络管理协议）服务器通告链路的操作状态变更，和端口角色的变化。

总步骤

1. configure terminal

2. snmp mib rep trap-rate *value*

3. end

4. show running-config

5. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Switch# configure terminal`	进入全局配置模式
步骤 2	**snmp mib rep trap-rate** *value* 示例： `Switch(config)# snmp mib`	让交换机发送 REP trap，并且设置每秒发送 trap 的数量。 输入每秒发送 trap 的数量。取值范围是从 0 到 1000。默认值为 0（即不施加限制，每次出现 trap

	rep trap-rate 500	即予发送）
步骤 3	**end** 示例： Switch(config)# end	返回特权 EXEC 模式
步骤 4	**show running-config** 示例： Switch# show running-config	（可选）显示运行配置，这些信息可以用来验证 REP trap 的配置
步骤 5	**copy running-config startup-config** 示例： Switch# copy running-config startup-config	（可选）将输入的条目保存到配置文件中

监控 REP

总步骤

1. show interface [*interface-id*] **rep** [**detail**]

2. show rep topology [**segment** *segment-_id*] [**archive**] [**detail**]

具体步骤

	命令或操作	**目的**
步骤 1	**show interface** [*interface-id*] **rep** [**detail**]	显示某个接口或所有接口的 REP 配置与状态。 • （可选）**detail**：显示特定接口的 REP 信息
步骤 2	**show rep topology** [**segment** *segment-_id*] [**archive**] [**detail**]	显示某个网段或所有网段的 REP 拓扑信息，包括这个网段中的主用边缘端口和辅助边缘端口。 • （可选）**archive**：显示最后的稳定拓扑； **注释**：交换机重启之后，存档的拓扑不会保留 • （可选）**detail**：显示具体的存档信息

配置单向链路检测

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com/go/cfn。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置 UDLD 的限制条件

下面是配置单向链路检测（UDLD）的限制条件：

- 支持 UDLD 的端口如果与另一台设备上不支持 UDLD 的端口相连，那么它也无法检测单向链路；
- 在配置模式（普通模式或主动模式）时，要确保链路两端配置的是同一个模式；

注意： 环路防护特性只能部署在点到点链路上。我们推荐链路每一端都有一个直连的设备在运行 STP。

关于 UDLD 的信息

单向链路检测（UDLD）是二层协议，可以让通过光纤或双绞线以太网线缆连接的设备监控线缆的物理配置，并且检测单向链路。所有相连的设备必须都能够支持 UDLD，这样协议才能成功地发现并且禁用单向链路。当 UDLD 检测到一条单向链路时，它会禁用相关端口并且向用户发出告警信息。单向链路可能会导致一系列的问题，包括生成树拓扑环路。

操作模式

UDLD 支持两种操作模式：普通（默认）模式和主动模式。在普通模式中，UDLD 可以检测到光纤连接当中因端口误连接，而产生的单向链路。在主动模式中，UDLD 也可以通过光纤、双绞线上的单向流量，或者光纤链路上端口的误连接，检测到单向链路问题。
在正常和主动模式下，UDLD 可以通过一层机制学习到链路的物理状态。在一层，自动协商机制会处理物理信令和容错检测。UDLD 会执行那些自动协商机制无法执行的任务，譬如检测邻居的身份，和关闭误连接的端口。在启用自动协商和 UDLD 时，一层和二层检测会共同防止出现物理和逻辑单向连接，以及其他协议的故障。
当邻居可以接收到本地设备发送的流量，而邻居发送的流量本地设备却无法接收到时，即表示网络中出现了单向链路。

普通模式

在普通模式下，当光纤端口中的纤维束连接有误，而一层机制又没有检测出这个错误时，UDLD 就可以检测出单向链路。如果端口连接正确，但流量却是单向的，UDLD 就无法检测出

单向链路，因为本该检测出这种情况的一层机制检测不出这种问题。此时，逻辑链路会被视为未确定，UDLD 也不会禁用这个端口。

当 UDLD 工作在普通模式下时，如果纤维对中的一条纤维束连接有误，那么只要自动协商功能正常，这条链路就不会保持在 up 状态，因为一层机制会检测出这条链路上的物理问题。在本例中，UDLD 不会采取任何操作，而逻辑链路也会视为未确定。

主动模式

在主动模式下，UDLD 会使用此前的检测方式来检测单向链路。主动模式下的 UDLD 也可以检测出点到点链路上的单向链路，而这类链路上设备之间是不允许出现故障的。当出现下列这些问题时，UDLD 也可以检测出单向链路：

- 在光纤或双绞线链路上，一个端口无法发送或接收流量；
- 在光纤或双绞线链路上，一个端口关闭，另一个端口打开；
- 线缆中一个纤维束连接错误；

在这些情况下，UDLD 都会禁用受影响的端口。

在一条点到点链路上，UDLD hello 数据包可以视为是心跳信号，它的存在是链路健康状态的佐证。相反，检测不到心跳表示如果无法重新建立双向链路的话，那么这条链路就必须关闭。

如果层一层的角度来看，光纤线缆中的纤维束工作正常，那么主动模式下的 UDLD 会检测到这些纤维束是否连接正确，以及流量是否正在邻居间双向流动。这些校验是不能通过自动协商来完成的，因为自动协商是工作在一层的。

检测单向链路的方法

UDLD 有两种工作方式：

- 邻居数据库维护
- 事件驱动检测与回声

邻居数据库维护

UDLD 会在各个端口上通过周期性发送的 hello 数据包（也称为通告消息或探针）来学习其他 UDLD 邻居，以确保每台设备接收到邻居的通告。

当设备接收到 hello 消息时，它就会将信息缓存起来，直到老化时间（抑制时间或生存时间）超时为止。如果设备在较老的缓存条目超时之前，又接收到了新的 hello 消息，那么设备就会用新的条目替换掉老的条目。

当一个端口被禁用，而 UDLD 又在运行时，那么无论何时用户禁用了端口上的 UDLD，或者 UDLD 重置，UDLD 都会针对那些因配置变更而受到影响的端口，清除缓存的条目。UDLD 会发送至少一条消息来通告邻居，让它们冲刷掉受状态变更响应的那些缓存。这个消息的目的在于确保缓存条目是同步的。

事件驱动检测与回声

UDLD 在检测操作中需要依赖回声机制。只要 UDLD 设备学习到了新的邻居，或者从不同步的邻居那里接收到了一条重新同步缓存的请求，设备就会在自己连接的这一侧重新开启检测窗口，并且发送 echo（回声）消息作出响应。由于这种操作在所有 UDLD 邻居上都是相同的，因此 echo 的发送方也会期待能够接收到对方发来的 echo 消息。

如果直到检测窗口结束，设备都没有接收到响应消息，链路有可能就会关闭，具体操作取决于 UDLD 的模式。当 UDLD 工作在普通模式下时，这条链路会被视为是未确定的，而这条链路可能不会关闭。当 UDLD 工作在主动模式下时，这条链路则会被视为是单向链路，因此端口就会被禁用。

相关主题

UDLD 重置的可选项

如果一个接口因 UDLD 而被禁用，用户可以下面几种选项来重置 UDLD：

- 输入接口配置命令 **udld reset**；
- 在输入接口配置命令 **shutdown** 后，再输入接口配置命令 **no shutdown** 来重新启动禁用的端口；
- 在输入全局配置命令 **no udld** {**aggressive | enable**}之后，再输入全局配置命令 **udld** {**aggressive | enable**}来重新启用禁用的端口；
- 在输入接口配置命令 **no udld port** 之后，再输入接口配置命令 **udld port** [**aggressive**]来重新启用禁用的光纤端口；
- 输入全局配置命令 **errdisable recovery cause udld** 启用计时器，让端口自动从 UDLD error-disabled 状态恢复过来，然后输入全局配置命令 **errdisable recovery interval** *interval* 来设置从 UDLD error-disabled 恢复的时间。

默认的 UDLD 配置

表 68：默认的 UDLD 配置

特性	**默认设置**
UDLD 全局启用状态	全局禁用
光纤媒介的 UDLD 每端口启用状态	在所有以太网光纤端口上禁用
双绞线（铜线）媒介的 UDLD 每端口启用状态	在所有 Ethernet 10/100 和 1000BASE-TX 端口上禁用
UDLD 主动模式	禁用

如何配置 UDLD

在全局启用 UDLD（CLI）

用户可以按照下面的步骤在设备的所有光纤端口上启用主动或正常模式的 UDLD，并且设置消息计时器。

总步骤

1. configure terminal

2. udld {**aggressive** | **enable** | **message time** *message-timer-interval*}

3. end

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**udld** {**aggressive** | **enable** | **message** **time** *message-timer-interval*}	设置 UDLD 的操作模式： • **aggressive**：在所有光纤端口上启用主动模式的 UDLD；

	示例： `Device(config)#` **udld enable message time 10**	• **enable**：在设备的所有光纤端口上启用普通模式的 UDLD。UDLD 默认是禁用的； 个别接口上的配置会覆盖全局配置命令 **udld enable** 所作的设置。 • **message time** *message-timer-interval*：配置处于通告阶段，且为双向通信状态端口，发送两条 UDLD 探针消息之间的时间周期。 **注释**：这条命令只会影响光纤端口。用户可以使用接口配置命令 **udld** 在其他类型的端口上启用 UDLD 使用这条命令的 **no** 形式来禁用 UDLD
步骤 3	**end** 示例： `Device(config)#` **end**	返回特权 EXEC 模式

在接口上启用 UDLD（CLI）

用户可以按照下面的步骤在一个端口上启用主动或正常模式的 UDLD。

总步骤

1. configure terminal

2. interface *interface-id*

3. udld port [**aggressive**]

4. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *interface-id* 示例： `Device(config)#` **interface gigabitethernet 1/0/1**	选择要启用 UDLD 的端口，并进入接口配置模式
步骤 3	**udld port** [**aggressive**] 示例： `Device(config-if)#` **udld port aggressive**	UDLD 默认是禁用的： • **udld port**：在特定端口上启用普通模式的 UDLD； • **udld port aggressive**：（可选）在特定端口上启用主动模式的 UDLD； **注释**：用户可以使用接口配置命令 **no udld port** 禁用特定光纤端口上的 UDLD
步骤 3	**end**	返回特权 EXEC 模式

	示例： `Device(config-if)# end`	

UDLD 的监控与维护

命令	目的
show udld [*interface-id* | **neighbors**]	显示特定端口或所有端口的 UDLD 状态

其他关于 UDLD 的参考资料

相关文档

相关主题	文档名
二层命令参考	《第 2/3 层命令参考手册（Inspur 6650 交换机）》

错误消息解码器

描述	链接
用户如需搜索和解析这个版本的系统错误消息，可以使用错误消息解码器这项工具	http://www.icntnetworks.com/icnt

标准与 RFC

标准/RFC	标题
无	——

技术助手

描述	链接
Inspur 支持（Inspur Support）页面可以为用户提供大量在线资源，其中包括排错的文档和工具，以及对 Inspur 产品与技术中若干问题的解析。 用户如需获取关于所购产品的安全与技术信息，可以选择订阅各类相关服务，譬如产品告警工具（通过最新产品问题信息汇总进行访问）、Inspur 技术服务通讯以及资讯聚合馈送（RSS Feeds）。 在 Inspur 支持页面中访问大多数工具都需要在 icntnetworks.com 上注册一个用户 ID 和密码	http://www.icntnetworks.com/icnt

UDLD 的特性信息

版本	修改
Inspur INOS 11.3.1	引入该特性

第 9 部分 多协议标签交换

Inspur 交换机上的多协议标签交换（MPLS）

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

有关 MPLS 的信息

MPLS 概述

多协议标签交换（MPLS）结合了二层（数据链路层）交换的性能和能力和三层（网络层）路由的可扩展性。MPLS 使用户能够应对网络利用率爆炸性增长的挑战，同时还提供了实现

差异化服务的机会，并不会以牺牲现有的网络基础设施为代价。MPLS 架构非常灵活，可以与任何二层技术结合使用。所有三层协议都支持 MPLS，并且对比于当今网络所提供的扩展功能，MPLS 提供了远超于此的可扩展性。

MPLS 的功能性描述

标签交换是一项高性能的数据包转发技术，它通过数据链路层（第 2 层）交换的性能和流量管理功能，实现了可扩展性和灵活性，通过网络层（第 3 层）路由功能实现了高性能。

标签交换功能

在传统的三层转发机制中（比如数据包在网络中的传输），每台交换机都需要从三层头部中提取与数据包转发相关的全部信息。然后把这些信息当作索引，在路由表中进行查找，以便确定数据包的下一跳。

在大多数一般化的情况下，头部中唯一与数据包转发相关的字段是目的 IP 地址字段，但在有些情况下，头部中的其他字段也可能与转发相关。因此，每台转发数据包的交换机必须独立地进行头部分析。而且，每台交换机还必须执行复杂的表查找工作。

在标签交换环境中，三层头部的分析工作只需要执行一次。然后三层头部会被映射为一个固定长度、非结构化的数值，也称为*标签*。

多个不同的头部可以映射为相同的标签，只要交换机转发这些头部总是使用相同的下一跳。事实上，一个标签代表一个*转发等价类*——这些数据包虽然可能有所区别，但对于转发功能来说，并无法对它们进行区分。

标签的初始选择不必只基于三层数据包头部中的内容；比如后续转发设备也可以基于路由策略来做出转发决定。

分配了标签后，三层数据包前面会添加上一个简短的标签头部。这个头部会作为数据包的一部分，伴随着它在网络中传输。网络转发路径中的每台 MPLS 交换机都会对标签执行交换工作，并通过在 MPLS 转发表中查找数据包头部携带的标签，做出转发决定。因此在数据包穿越网络的过程中，交换机不再需要重新评估数据包头部。并且由于标签是固定长度且非结构化的数值，因此 MPLS 转发表的查找过程既直接又快速。

标签绑定信息的分发

网络中的每台标签交换路由器（LSR）都会独立地在本地决定使用哪个标签值来代表一个转

发等价类。这种关联行为称为标签绑定。每台 LSR 都会告知邻居，自己的标签绑定信息。LSR 是通过使用下列协议，来向邻居交换机告知自己的标签绑定信息的：

- 标签分发协议（LDP）——允许一个 MPLS 网络中的对等体 LSR 之间交换标签绑定信息，以便在这个 MPLS 网络中支持逐跳转发；
- 边界网关协议（BGP）——用来支持 MPLS 虚拟专用网（VPN）。

当 LSR A 向其邻居 LSR B 发送携带标签的数据包时，这个 IP 数据包中携带的标签值正是 SLR B 用来代表这个数据包的转发等价类的标签值。因此，随着 IP 数据包在网络中的传输，标签值也会发生变化。

如何配置 MPLS

这一部分解释了如何实施基本配置，以便为交换机执行 MPLS 交换和转发做好准备。
其他 MPLS 应用的配置任务记录在该应用的特性模块文档中。

配置交换机实现 MPLS 交换

在 Inspur 交换机上实现 MPLS 交换需要启用 Inspur 快速转发（Express Forwarding）特性。
有关 Inspur 快速转发命令的更多信息，请参考 Inspur INOS 交换命令指南。

总步骤

1. enable

2. configure terminal

3. ip cef distributed

4. mpls label range *minimum-value maximum-value*

5. mpls label protocol ldp

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式 • 在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device# configure terminal`	
步骤 3	**ip cef distributed** **示例：** `Device(config)# ip cef distributed`	在交换机上启用 Inspur 快速转发特性
步骤 4	**mpls label range** *minimum-value maximum-value* **示例：** `Device(config)# mpls label range 16 4096`	配置本地使用的标签范围，用于数据包接口上的 MPLS 应用
步骤 5	**mpls label protocol ldp** **示例：** `Device(config)# mpls label protocol ldp`	指定设备使用的标签分发协议

验证 MPLS 交换的配置

要想验证是否已正确配置了 Inspur 快速转发特性，输入命令 **show ip cef summary**，这条命令会输出类似下列这些信息：

总步骤

1. show ip cef summary

具体步骤

show ip cef summary

示例：

```
Switch# show ip cef summary
IPv4 CEF is enabled for distributed and running VRF Default
150 prefixes (149/1 fwd/non-fwd)
Table id 0x0
```

```
Database epoch: 4 (150 entries at this epoch)
Switch#
```

配置交换机实现 MPLS 转发

要想在 Inspur 交换机上实现 MPLS 转发，需要启用 IPv4 数据包转发功能。

总步骤

1. enable

2. configure terminal

3. interface *type slot/subslot/port*

4. mpls ip

5. mpls label protocol ldp

6. end

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式 • 在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *type slot/subslot /port* **示例：** `Device(config)# interface gigabitethernet 1/0/0`	指定千兆以太网接口并进入接口配置模式。指定交换机虚拟接口的示例如下所示 Device(config)# interface vlan 1000
步骤 4	**mpls ip** **示例：** `Device(config-if)# mpls ip`	在路由模式的物理接口（千兆以太网接口）、交换机虚拟接口（SVI）或 Port-Channel 上，为 IPv4 数据包启用 MPLS 转发

步骤 5	**mpls label protocol ldp** **示例：** `Device(config-if)# mpls label protocol ldp`	为接口指定标签分发协议 **注释：** 不能在虚拟路由和转发（VRF）接口上启用 MPLS LDP
步骤 6	**end** **示例：** `Device(config-if)# end`	退出接口配置模式并返回特权 EXEC 模式

验证 MPLS 转发的配置

要想验证是否已正确配置了 MPLS 转发功能，输入命令 **show mpls interfaces detail**，这条命令会输出类似下列这些信息：

总步骤

1. show mpls interfaces detail

2. show running-config interface

具体步骤

步骤1. show mpls interfaces detail

示例：

物理（千兆以太网）接口：

```
Switch# show mpls interfaces detail interface GigabitEthernet 1/0/0
Type Unknown
IP labeling enabled
LSP Tunnel labeling not enabled IP FRR labeling not enabled
BGP labeling not enabled
MPLS not operational
MTU = 1500
```

交换机虚拟接口（SVI）：

```
Switch# show mpls interfaces detail interface Vlan1000
Type Unknown
IPlabeling enabled (ldp) :
```

```
Interface config
LSP Tunnel labeling not enabled IP FRR labeling not enabled
BGP labeling not enabled
MPLS operational
MTU = 1500
```

步骤2. **show running-config interface**

示例：

物理（千兆以太网）接口：

```
Switch# show running-config interface interface GigabitEthernet 1/0/0
Building configuration...
Current configuration : 307 bytes !
interface TenGigabitEthernet1/0/0 no switchport
ip address xx.xx.x.x xxx.xxx.xxx.x mpls ip
mpls label protocol ldp
end
```

交换机虚拟接口（SVI）：

```
Switch# show running-config interface interface Vlan1000
Building configuration...
Current configuration : 187 bytes !
interface Vlan1000
ip address xx.xx.x.x xxx.xxx.xxx.x mpls ip
mpls label protocol ldp
end
```

MPLS 三层 VPN

多协议标签交换（MPLS）虚拟专用网（VPN）由一组通过 MPLS 服务提供商核心网络连接在一起的站点构成。在每个客户站点上，都有一台或多台客户边界（CE）路由器连接着一台或多台服务提供商边界（PE）路由器。

在配置 MPLS 三层 VPN 之前，用户应该已经在网络中实施了 MPLS、标签分发协议（LDP）和 Inspur 快速转发（CEF）。核心网络中的所有路由器（包括 PE 路由器）必须能够支持 CEF

和 MPLS 转发。

MPLS QoS EXP 的分类和标记

通过使用 QoS EXP 匹配特性，用户可以通过修改 IP 数据包中的多协议标签交换（MPLS）实验位（EXP），来分类并标记网络流量。

通过使用 QoS EXP 匹配特性，用户可以通过设置 MPLS 数据包中 MPLS EXP 字段的取值来管理网络流量。通过为 MPLS EXP 字段选择不同的数值，用户可以对数据包进行标记，这样可以为在拥塞期间需要较高优先级的数据包设置优先级。通过设置 MPLS EXP 值，用户可以实现：

- **流量分类：**分类过程会选择处需要标记的流量。分类功能会把流量分隔为多个优先级，或者服务类别。流量分类是基于类的 QoS 部署方案重的重要组成部分；
- **流量限速和标记：**限速特性可以使交换机丢弃那些超出了限速的流量，或者为这些流量标记一个不同的丢弃级别。标记流量是一种区分数据包流的方法。通过使用数据包标记特性，用户能够把自己的网络分隔为多个优先级别或服务类别。

警告

下面是一些警告：

- 只支持统一（Uniform）模式和管道（Pipe）模式；不支持短管道（Short-pipe）模式；
- QoS 组的取值支持范围为 0 至 30（总共 31 个 QoS 组）；
- 只支持在外部标签上使用基于 QoS 策略的 EXP 标记；不支持内部 EXP 标记。

术语表

BGP——外部网关协议。IP 网络中使用的最具有影响力的域间路由协议；

边界网关协议——见 BGP；

FIB——转发信息库。包含了来自于 IP 路由表的转发信息的表格；

转发信息库——见 FIB；

标签——一个短小、固定长度的识别符，告诉交换节点如何转发数据（数据包或网元）；

标签绑定——一个标签与一组数据包之间的关联，这种关联可以通告给邻居，从而建立标签转发路径；

标签分发协议——见 LDP；

标签转发信息库——见 LFIB；

标签压入——在数据包上添加第一个标签的操作；

标签交换路由器——见 LSR；

LDP——标签分发协议。这个协议通过分发标签与网络前缀之间的绑定关系，来支持 MPLS 逐跳转发；

LFIB——标签转发信息库。一种数据结构，其中将目的地和入站标签，与出站接口和标签关联在一起；

MPLS——多协议标签交换。基于标签进行交换的工业标准；

MPLS 逐跳转发——使用 MPLS 转发机制，沿着普通的路由路经转发数据包；

多协议标签交换——见 MPLS；

RIB——路由信息库。一种通用数据库，其中包含一台路由器上运行的所有路由协议；

路由信息库——见 RIB；

虚拟专用网——见 VPN；

VPN——虚拟专用网。一种网络类型，使 IP 流量能够使用隧道技术进行传输，安全地穿越公共 TCP/IP 网络。

配置组播虚拟交换网

配置组播 VPN

组播 VPN（MVPN）特性提供了在三层 VPN 上支持组播的功能。随着企业逐渐扩展其组播应用的范围，服务提供商可以通过自己的多协议标签交换（MPLS）核心网络来提供组播服务。IP 组播通过 MPLS VPN 网络核心实现流媒体视频、语音和数据的传输。

曾经，点到点隧道是唯一一种通过服务提供商网络进行连接的方式。尽管这种通过隧道连接的网络容易伴有扩展性问题，但这却是通过 VPN 传输 IP 组播流量的唯一方式。

由于三层 VPN 只支持单薄流量的连通性，因此 MPLS 和三层 VPN 的结合部署方式能够使服务提供商同时为三层 VPN 客户提供单薄和组播流量的连通性。

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置组播 VPN 的先决条件

用户可以使用“配置基本的 IP 组播”文档中描述的任务，来启用 IP 组播并配置 PIM 接口。

配置组播 VPN 的限制条件

- 设备上为边界网关协议（BGP）对等体配置的更新源接口，必须对于所有 BGP 对等体来说都是相同的，这样才能正确配置默认的组播分发树（MDT）。如果用户为 BGP 对等体使用了环回地址，则必须在环回地址上启用 PIM 稀疏模式；
- MVPN 不支持多个 BGP 对等体更新源；
- 不支持多个 BGP 更新源，配置多个 BGP 更新源会使 MVPN 反向路径转发（RPF）校验失败。MVPN 隧道的源 IP 地址就是 BGP 对等体更新源所使用的最大 IP 地址。如果这个 IP 地址没有在远端服务提供商边界（PE）设备上被用为 BGP 对等体地址，那么 MVPN 将无法正常运行。

配置组播 VPN 的相关信息

组播 VPN 的工作原理

服务提供商使用 MVPN IP 特性能够在 MPLS VPN 环境中配置并支持组播流量。这个特性能够

为每个独立的 VRF 实例提供组播数据包的路由和转发，还能够提供一种机制，在服务提供商骨干网中传输 VPN 组播数据包。

VPN 是通过共享的基础设施（如 ISP）建立的网络连通性。它的功能是像私有网络那样提供相同的策略和性能，同时削减所有者的成本，从而在多方面为运营和基础设施架构节省了成本。

一个企业利用 MVPN，可以通过服务提供商的骨干网无缝连接自己的私有网络。使用 MVPN 来连接企业网络的方式，并不会影响企业网络的管理方式，也不会改变其他的网络连接。

组播 VPN 的优势

- 以可扩展的方式，向多个站点动态发送信息；
- 提供了高速率的信息传输；
- 在共享的基础设施上提供了连通性。

组播 VPN 的路由和转发以及组播域

MVPN 向 VPN 路由和转发表中引入了组播路由信息。当服务提供商边界（PE）设备从客户边缘（CE）路由器那里收到了组播数据或控制数据包，它会根据组播 VPN 路由和转发实例（MVRF）中的信息执行转发操作。MVPN 不使用标签交换。

相互之间可以发送组播流量的一组 MVRF 构成了一个组播域。举例来说，如果一个公司希望向它的所有全球雇员发送特定类型的组播流量，它的组播域会由所有与该公司相关联的 CE 路由器构成。

组播分发树

MVPN 会为每个组播域建立一个静态的默认组播分发树（MDT）。默认 MDT 定义了 PE 路由器使用的路径，PE 路由器使用这些路径向组播域中的每台其他 PE 路由器发送组播数据和控制消息。

如果将特定源组播（SSM）用作核心组播路由协议，那么默认 MDT 和数据 MDT 所使用的组播 IP 地址必须配置在所有 PE 路由器的 SSM 范围中。

MVPN 还支持为高带宽传输动态创建 MDT。数据 MDT 是 Inspur INOS 软件中的一项特有特性。数据 MDT 主要用于传输高带宽源，比如 VPN 中传输的全动态视频，它能够保障 MPLS VPN 核心网中理想的流量转发。用户可以在每台路由器的基础上，或者在每个 VRF 的基础上，

配置流量门限值，根据这个门限值来创建数据 MDT。当组播传输超过了用户定义的门限值，发送方 PE 路由器就会创建数据 MDT，并向默认 MDT 上的所有路由器发送一个 UDP 消息，这个 UDP 消息中包含有关它所创建的数据 MDT 的信息。设备每秒钟都会检查数据 MDT 门限值，确认组播流是否超出了这个值。在 PE 路由器发出了 UDP 消息后，它会在切换前至少等待 3 秒；13 秒是最长的切换时间，3 秒是最短的切换时间。

设备只会为 VRF 组播路由表中的(S, G)组播路由条目创建数据 MDT。无论单独的源数据传输速率是多少，它也不会为(*, G)条目创建数据 MDT。

在下面的案例中，服务提供商的组播客户分别在圣何塞、纽约和达拉斯都有站点。圣何塞正在进行单向的组播展示。服务提供商网络同时支持与这个客户相关联的 3 个站点，除此之外，它还有另一个企业的客户，其站点位于休斯敦。

企业客户的默认 MDT 由服务提供商路由器 P1、P2 和 P3，以及客户所关联的 PE 路由器所构成。PE4 不属于默认 MDT 的一部分，因为它是与另一个客户相关联的。从图片中可以看出，没有数据流沿着默认 MDT 发送，因为圣何塞之外的用户都没有加入组播流。

图 69：默认组播分发树概数

Multicast sender	组播 发送方
Local multicast recipient	本地组播 接收方
Customer 1 San Jose Site	客户 1 圣何塞站点
Customer 1 New York Site	客户 1 纽约站点
MPLS Core	MPLS 核心
PIM (SM/bidir/SSM) in Core	核心网 PIM（SM/bidir/SSM）
Customer 2 Houston Site	客户 2 休斯敦站点
Customer 1 Dallas Site	客户 1 达拉斯站点

纽约站点中的一位雇员加入了组播会话。与纽约站点相关联的 PE 路由器发送出加入请求，这个请求消息沿着为这个客户的组播域创建的默认 MDT 发送。PE1（与组播会话源相关联的 PE 路由器）收到了这个请求。下图中展示了 PE 路由器向 CE 路由器转发请求的操作，这台

CE 路由器是与组播源相关联的 CE 路由器（CE1a）。

图 70：数据 MDT 的初始化

Multicast sender	组播 发送方
Local multicast recipient	本地组播 接收方
1. Remote enterprise client issues join request	1. 远端企业客户端 发出了加入请求
2. PE2 sends join request along default MDT	2. PE2 沿着 默认 MDT 发出了加入请求
Customer 1 San Jose Site	客户 1 圣何塞站点
Customer 1 New York Site	客户 1 纽约站点
3. PE1 receives join request and asks CE1a to begin sending data	3. PE1 收到了 加入请求并要求 CE1a 开始发送数据
MPLS Core	MPLS 核心
Customer 2 Houston Site	客户 2 休斯敦站点
Customer 1 Dallas Site	客户 1 达拉斯站点

CE 路由器（CE1a）开始向其关联的 PE 路由器（PE1）发送组播数据，这些组播数据是沿着默认 MDT 发送的。在开始发送组播数据后，PE1 立即意识到组播数据流量超出了需要创建数据 MDT 的带宽门限值。因此 PE1 创建了数据 MDT，并使用默认 MDT 向所有路由器发送了一个消息，这个消息中包含了有关这个数据 MDT 的信息；3 秒钟过后，PE1 开始使用数据 MDT 为这个特定的组播流发送组播数据。目前只有 PE2 有兴趣接收这个组播源发来的数据，因此只有 PE2 会加入数据 MDT 并从这里接收流量。

PE 路由器会通过默认 MDT 与其他 PE 路由器之间维护 PIM 关系，同时会与其直连的 PE 路由

器之间维护 PIM 关系。

组播隧道接口

MVRF 是基于每个组播域创建的，设备需要创建一个隧道接口，并以这个隧道接口为源发起所有 MVRF 流量。组播隧道接口是 MVRF 用来访问组播域的接口。可以把它当作是连接一个 MVRF 和全局 MVRF 的管道。隧道接口是基于每个 MVRF 创建的。

BGP 中用于组播 VPN 的 MDT 地址家族

用户在命令 **address-family ipv4** 中添加关键字 **mdt**，可以配置 MDT 地址家族会话。MDT 地址家族会话能够使用边界网关协议（BGP）的 MDT 子地址家族识别符（SAFI）更新消息，向 PIM 传递源 PE 地址和 MDT 组地址。

组播 VPN 支持的 BGP 通告方式

在单个自治系统中，如果一个 MVPN 的默认 MDT 使用了 PIM 稀疏模式（PIM-SM），并部署了汇集点（RP），那么 PIM 能够通过组播隧道接口（MTI）建立邻接关系，因为源 PE 和接收方 PE 能够通过 RP 发现彼此。在这种环境中，本地 PE（源 PE）会向 RP 发送注册消息，接着 RP 会建立去往源 PE 的最短路径树（SPT）。远端 PE 作为 MDT 组播组的接收方，会向 RP 发送(*, G)加入消息，并加入这个组的分发树。

但是，如果默认 MDT 组的配置环境是 PIM 特定源组播（PIM-SSM）环境，而不是 PIM-SM 环境的话，接收方 PE 就需要源 PE 和默认 MDT 组的信息。因为它需要使用这些信息向源 PE 发送(S,G)加入消息，来建立从源 PE 到本地的分发树（不需要 RP）。源 PE 地址和默认 MDT 组地址是使用 BGP 发送的。

BGP 扩展团体

当使用了 BGP 扩展团体时，PE 环回接口（源地址）的信息是使用路由区分符（RD）类型 2（为了将其与单播 VPNv4 前缀相区分），作为 VPNv4 前缀进行发送的。MDT 组地址是携带在 BGP 扩展团体中的。通过在 VPNv4 地址中嵌入源地址，以及在扩展团体中携带组地址，同一个 MVRF 实例中的 PE 路由器之间能够建立 SSM 树。

注释： 在引入 MDT SAFI 之前，BGP 扩展团体属性作为一种临时的解决方案，在 IETF 完成

标准化之前，用于通告源 PE 的 IP 地址和默认 MDT 组地址。但是 MVPN 环境中的 BGP 扩展团体属性也有其局限性：不能用在多个 AS 之间的环境中（因为这个属性是非传递的），并且它使用 RD 类型 2（这不是一个广泛支持的标准）。

如何配置组播 VPN

配置数据组播组

对于一台 PE 设备中的一个 VRF 实例中的一个 VPN 来说，一个数据 MDT 组中可以最多包含 256 个组播组。用来创建 MDT 组的组播组地址是从配置的 IP 地址池中动态选择的。

总步骤

1. enable

2. configure terminal

3. vrf definition *vrf-name*

4. rd *route-distinguisher*

5. route-target both *ASN:nn or IP-address:nn*

6. address family ipv4 unicast *value*

7. mdt default *group-address*

8. mdt data *group number*

9. mdt data threshold *kbps*

10. mdt log-reuse

11. end

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式 • 在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 3	**vrf definition** *vrf-name* **示例：** `Device(config)# vrf definition vrf1`	进入 VRF 配置模式，通过指定 VRF 名称来定义 VPN 路由实例
步骤 4	**rd** *route-distinguisher* **示例：** `Device(config-vrf)# rd 1:1`	为一个 VRF 创建路由和转发表 • 变量 *route-distinguisher* 指定了一个 8 字节的数值，将其添加在 IPv4 前缀上就创建了 VPN IPv4 前缀。用户可以使用下列格式输入 *route-distinguisher*： • 16 比特的自治系统号（ASN）：用户的 32 比特号码。比如 101:3 • 32 比特的 IP 地址：用户的 16 比特号码。比如 192.168.122.15:1
步骤 5	**route-target both** *ASN:nn or IP-address:nn* **示例：** `Device(config-vrf)# route-target both 1:1`	为一个 VRF 创建路由目标扩展团体。关键字 **both** 为目标 VPN 扩展团体指定了导入和导出路由信息
步骤 6	**address family ipv4 unicast** *value* **示例：** `Device(config-vrf)# address family ipv4 unicast`	进入 VRF 地址家族配置模式并为 VRF 指定一个地址家族。 • 关键字 **ipv4** 为 VRF 指定了 IPv4 地址家族
步骤 7	**mdt default** *group-address* **示例：** `Device(config-vrf-af)# mdt default 226.10.10.10`	为一个 VRF 的数据 MDT 组配置组播组地址。 • 输入这条命令后设备会创建隧道接口 • 在同一个 VRF 中的所有 PE 上，默认 MDT 组地址的配置必须相同

步骤 8	**mdt data** *group number* **示例：** `Device(config-vrf-af)# mdt data 232.0.1.0 0.0.0.31`	指定数据 MDT 池中使用的地址范围
步骤 9	**mdt data threshold** *kbps* **示例：** `Device(config-vrf-af)# mdt data threshold 50`	以 *kbps* 为单位指定门限值。取值范围为 1 至 4294967
步骤 10	**mdt log-reuse** **示例：** `Device(config-vrf-af)# mdt log-reuse`	（可选）启用数据 MDT 重新使用的记录行为，当一个数据 MDT 再次使用时，设备会生成一个系统日志消息
步骤 11	**end** **示例：** `Device(config-if)# end`	退出接口配置模式并返回特权 EXEC 模式

为 VRF 配置模式 MDT 组

执行这个任务来为 VRF 配置默认 MDT 组。

属于相同 VPN 的所有设备上，必须配置相同的默认 MDT 组。源 IP 地址就是用来发起 BGP 会话的地址。

总步骤

1. enable

2. configure terminal

3. ipmulticast-routing

4. ipmulticast-routing vrf *vrf-name*

5. vrf definition *vrf-name*

6. rd *route-distinguisher*

7. route-target both *ASN:nn or IP-address:nn*

8. address family ipv4 unicast *value*

9. mdt default *group-address*

10. end

11. configure terminal

12. ippim vrf *vrf-name* **rp-address** *value*

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式 • 在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip multicast-routing** **示例：** `Device(config)# ip multicast-routing`	启用组播路由
步骤 4	**ip multicast-routingvrf** *vrf-name* **示例：** `Device(config)# ip multicast-routing vrf vrf1`	配置支持的 MVPN VRF 实例
步骤 5	**vrf definition** *vrf-name* **示例：** `Device(config)# vrf definition vrf1`	进入 VRF 配置模式，并通过指定 VRf 名称来定义 VPN 路由实例
步骤 6	**rd** *route-distinguisher*	为一个 VRF 创建路由和转发表 • 变量 *route-distinguisher* 指定了一

	示例： `Device(config-vrf)# rd 1:1`	个 8 字节的数值，将其添加在 IPv4 前缀上就创建了 VPN IPv4 前缀。用户可以使用下列格式输入 *route-distinguisher*： • 16 比特的自治系统号（ASN）：用户的 32 比特号码。比如 101:3 • 32 比特的 IP 地址：用户的 16 比特号码。比如 192.168.122.15:1
步骤 7	**route-target both** *ASN:nn or IP-address:nn* **示例：** `Device(config-vrf)# route-target both 1:1`	为一个 VRF 创建路由目标扩展团体。关键字 **both** 为目标 VPN 扩展团体指定了导入和导出路由信息
步骤 8	**address family ipv4 unicast** *value* **示例：** `Device(config-vrf)# address family ipv4 unicast`	进入 VRF 地址家族配置模式并为 VRF 指定一个地址家族。 • 关键字 **ipv4** 为 VRF 指定了 IPv4 地址家族
步骤 9	**mdt default** *group-address* **示例：** `Device(config-vrf-af)# mdt default 226.10.10.10`	为一个 VRF 的数据 MDT 组配置组播组地址。 • 输入这条命令后设备会创建隧道接口 • 在同一个 VRF 中的所有 PE 上，默认 MDT 组地址的配置必须相同
步骤 10	**end** **示例：** `Device(config-vrf-af)# end`	退出接口配置模式并返回特权 EXEC 模式
步骤 11	**configure terminal** **示例：**	进入全局配置模式

	Device# configure terminal	
步骤 12	**ip pim vrf** *vrf-name* **rp-address** *value* **示例：** Device(config)# ip pim vrf vrf1 rp-address 1.1.1.1	进入 RP 配置模式

在 BGP 中为组播 VPN 配置 MDT 地址家族

执行这个任务在 PE 设备上配置 MDT 地址家族会话，以此为 MVPN 建立 MDT 对等体会话。

在开始前

在 MVPN 能够通过 MDT 地址家族建立对等体关系前，用户必须在 BGP 网络中启用 MPLS 和 Inspur 快速转发（CEF），以及在向 CE 设备提供 VPN 服务的 PE 设备上配置多协议 BGP。

注释： 不支持下列策略配置参数：

- 路由起源属性
- 网络层连通性信息（NLRI）前缀过滤（前缀列表、分发列表）
- 扩展团体属性（路由目标和源站点）

总步骤

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4 mdt**
5. **neighbor** *neighbor-address* **activate**
6. **neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**]
7. **exit**
8. **address-family vpnv4**
9. **neighbor** *neighbor-address* **activate**
10. **neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**]
11. **end**

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式

	示例： `Device> enable`	• 在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**router bgp** *as-number* **示例：** `Device(config)# router bgp 65535`	进入路由器配置模式并创建 BGP 路由进程
步骤 4	**address-family ipv4 mdt** **示例：** `Device(config-router)# address-family ipv4 mdt`	进入地址家族配置模式并创建一个 IP MDT 地址家族会话
步骤 5	**neighbor** *neighbor-address* **activate** **示例：** `Device(config-router-af)# neighbor 192.168.1.1 activate`	为指定邻居启用 MDT 地址家族
步骤 6	**neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**] **示例：** `Device(config-router-af)# neighbor 192.168.1.1 send-community extended`	与指定邻居之间启用团体属性和（或）扩展团体属性交换行为
步骤 7	**exit**	退出地址家族配置模式并返回路由器

	示例： `Device(config-router-af)# exit`	配置模式
步骤 8	**address family vpnv4** **示例：** `Device(config-router)# address-family vpnv4`	进入地址家族配置模式并创建一个 VPNv4 地址家族会话
步骤 9	**neighbor** *neighbor-address* **activate** **示例：** `Device(config-router-af)# neighbor 192.168.1.1 activate`	为指定邻居启用 VPNv4 地址家族
步骤 10	**neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**] **示例：** `Device(config-router-af)# neighbor 192.168.1.1 send-community extended`	与指定邻居之间启用团体属性和（或）扩展团体属性交换行为
步骤 11	**end** **示例：** `Device(config-router-af)# end`	退出接口配置模式并返回特权 EXEC 模式

验证 MDT 默认组的信息

总步骤

1. enable

2. show ip pim [**vrf** *vrf-name*] **mdt bgp**

3. show ip pim [**vrf** *vrf-name*] **mdt send**

4. show ip pim [**vrf** *vrf-name*] **mdt history interval** *minutes*

具体步骤

步骤1. enable

示例：

```
Device> enable
```

进入特权 EXEC 模式

- 在提示时输入密码

步骤2. show ip pim [**vrf** *vrf-name*] **mdt bgp**

示例：

```
Device# show ip pim mdt bgp
MDT-default group232.2.1.4
rid:1.1.1.1 next_hop:1.1.1.1
```

显示 MDT 默认组的 BGP RD 通告信息。

步骤3. show ippim [**vrf** *vrf-name*] **mdt send**

示例：

```
Device# show ip pim mdt send
MDT-data send list forVRF:vpn8
(source, group)  MDT-data group  ref_count
(10.100.8.10,  225.1.8.1)  232.2.8.0  1
(10.100.8.10,  225.1.8.2) 232.2.8.1 1
(10.100.8.10,  225.1.8.3) 232.2.8.2 1
(10.100.8.10,  225.1.8.4) 232.2.8.3 1
(10.100.8.10,  225.1.8.5) 232.2.8.4 1
(10.100.8.10,  225.1.8.6) 232.2.8.5 1
(10.100.8.10,  225.1.8.7) 232.2.8.6 1
(10.100.8.10,  225.1.8.8) 232.2.8.7 1
(10.100.8.10,  225.1.8.9) 232.2.8.8 1
(10.100.8.10,  225.1.8.10) 232.2.8.9 1
```

显示 MDT 数据组的详细信息，其中包括指定设备发出过的 MDT 通告。

步骤4. show ip pim [**vrf** *vrf-name*] **mdt history interval** *minutes*

示例：

```
Device# show ip pim vrf vrf1 mdt history interval 20
MDT-data send history for VRF- vrf1 for the past 20 minutes
MDT-data group  Number of reuse
10.9.9.8  3
10.9.9.9  2
```

显示在刚经过的时长内，被重新使用的数据 MDT。

组播 VPN 的配置案例

案例：配置 MVPN 和 SSM

在下列案例中，骨干网中配置了 PIM-SSM。因此默认 MDT 和数据 MDT 组配置在 SSM 的 IP 地址范围中。在 VPN 内部配置了 PIM-SM，并且只接受自动 RP 通告。

```
ip vrf vrf1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 mdt default 232.0.0.1
 mdt data 232.0.1.00.0.0.255 threshold 500 list101
!
ip pim ssm default
ip pim vrf vrf1 accept-rp auto-rp
```

案例：为组播路由启用 VPN

在下列案例中，为组播路由启用了名为 vrf1 的 VPN 路由实例：

```
ip multicast-routing vrf1
```

案例：为数据 MDT 组配置组播组地址范围

在下列案例中，VPN 路由实例被指定给名为 blue 的 VRF。VPN VRF 的 MDT 默认组是 239.1.1.1，MDT 组的组播组地址范围是 239.1.2.0，反掩码为 0.0.0.3：

```
ip vrf blue
```

```
rd 55:1111
route-target both 55:1111
mdt default 239.1.1.1
mdt data 239.1.2.0 0.0.0.3
end
```

案例：限制组播路由的数量

在下列案例中，能够放入组播路由表中的组播路由数量被设置为 200,000，会触发告警信息的组播路由数量门限值被设置为 20,000：

```
!
ip multicast-routing
ip multicast-routing vrf inspur
ip multicast cache-headers
ip multicast route-limit 200000 20000
ip multicast vrf inspur route-limit 200000
20000 no mpls traffic-eng auto-bw timers
frequency 0
!
```

配置组播 VPN 的其他参考资料

技术助手

描述	链接
Inspur 支持和文档网站提供了可供下载的在线资源，其中包括文档、软件和工具。用户可以使用这些资源来安装和配置软件，以及排查和解决 Inspur 产品和技术相关的技术问题。如需在 Inspur 支持和文档网站中访问更多资源，需要 Inspur.com 的用户 ID 和密码	http://www.icntnetworks.com

第 10 部分 网络管理

配置 Inspur INOS 配置引擎

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

实施配置引擎配置的先决条件

- 获得用户连接的配置引擎实例名称；
- 由于 CNS 同时使用事件总线和配置服务器来向设备提供配置，因此用户必须为每台需要配置的设备同时定义配置 ID 和设备 ID；
- 所有配置了全局配置命令 **cns config partial** 的设备都必须访问事件总线。设备上生成的

设备 ID 必须与 Inspur 配置引擎中为该设备定义的设备 ID 相同。用户必须知道自己连接的事件总线的主机名。

实施配置引擎配置的限制条件

- 在配置服务器的单个实例范围中，两台需要配置的设备不能拥有相同的配置 ID；
- 在事件总线的单个实例范围中，两台需要配置的设备不能拥有相同的设备 ID。

实施配置引擎配置的相关信息

Inspur 配置引擎软件

Inspur 配置引擎（Configuration Engine）是网络管理工具软件，它作为配置服务，能够自动部署和管理网络设备和服务。每个 Inspur 配置引擎可以管理一组 Inspur 设备（设备和路由器）及其提供的服务，同时还能够保存它们的配置，并在需要时提供这些配置。Inspur 配置引擎能够自动初始化配置和配置更新，因为它能够收集与设备相关的配置变更，将这些信息发送给设备，执行配置变更并记录变更结果。

Inspur 配置引擎支持单机模式和服务器模式，并且包含下列 Inspur 网络服务（CNS）组成部分：

- 配置服务：
  - Web 服务器
  - 文件管理器
  - Namespace 映射服务器
- 事件服务（事件网关）
- 数据服务目录（数据模型和模式）

在单机模式中，Inspur 配置引擎能够支持嵌入式目录服务。在这个模式中，不需要外部目录或其他数据存储。在服务器模式中，Inspur 配置引擎能够支持使用用户自定义的外部目录。

图 71：Inspur 配置引擎的架构概述

Service provider network	服务提供商网络
Configuration	配置引擎

engine	
Data service directory	数据服务 目录
Configuration server	配置服务器
Event service	事件服务
Web-based user interface	基于网页的 用户界面
Order entry configuration management	订单输入 配置管理

配置服务

配置服务（Configuration Service）是 Inspur 配置引擎中的核心组成部分。它由配置服务器（Configuration Server）构成，配置服务器会与设备上的 Inspur INOS CNS 代理协同工作。配置服务负责把设备和服务配置送达给设备，通过逻辑组完成初始配置和大量重配置工作。设备在网络中首次启动时，会从配置服务那里收到初始配置。

配置服务会使用 CNS 事件服务（Event Service）来发送和接收配置变更事件，以及发送成功和失败通知。

配置服务器是一台网页服务器，它所使用的配置模板和与设备相关的配置信息都保存在内置目录（单机模式）或远端目录（服务器模式）中。

配置模板是包含有静态配置信息的文本文件，文本格式与 CLI 命令相同。在模板中，变量是通过使用轻量目录访问协议（LDAP）URL 指定的，LDAP URL 调用了目录中存储的与设备相关的配置信息。

Inspur INOS 代理可以对收到的配置文件执行语法检查，并通过发布事件来展示语法检查是否成功。配置代理既可以立即应用配置，也可以等待从配置服务器那里收到同步事件后再应用配置。

事件服务

Inspur 配置引擎使用事件服务（Event Service）来接收和生成配置事件。事件服务由事件代理和事件网关构成。事件代理位于设备上，能够简化设备之间的通信；事件网关位于 Inspur 配置引擎上。

事件服务是一种高性能的发布和订阅通信方法。事件服务使用基于主题的寻址方式，向目的

地发送消息。基于主题的寻址方式为消息及其目的地定义了简单且统一的命名空间。

命名空间映射器

Inspur 配置引擎中包含命名空间映射器（NSM），它为设备的管理逻辑组提供了查找服务，能够基于应用、设备或组 ID，以及事件进行查找。

Inspur INOS 设备只能够识别与 Inspur INOS 软件中配置的事件主题名称相同的事件主题名称；比如 inspur.cns.config.load。用户可以使用命名空间映射服务，通过任意命名规范来指定事件。当用户使用主题名称来发布数据存储时，NSM 会将用户定义的事件主题名称符串更改为 Inspur INOS 已知的字符串。

对于订阅者来说，通过唯一的设备 ID 和事件，命名空间映射服务会返回一组可供订阅的事件。类似的，对于发布者来说，通过唯一的组 ID、设备 ID，以及事件，映射服务会返回一组可供发布的事件。

Inspur 网络服务 ID 和设备主机名

Inspur 配置引擎认为每台需要配置的设备上都关联了唯一的标识符。这个唯一的标识符可以是多个同义词，每个同义词在特定的命名空间中是唯一的。事件服务使用命名空间内容来进行基于主题的消息寻址。

Inspur 配置引擎中有两个命名空间，一个用于事件总线，另一个用于配置服务器。在配置服务器的命名空间范围中，术语*配置 ID（ConfigID）*表示设备的唯一标识符。在事件总线的命名空间范围中，术语*设备 ID（DeviceID）*表示设备的 CNS 唯一标识符。

配置 ID

每台需要配置的设备都有一个唯一的配置 ID，这是它在 Inspur 配置引擎目录中访问相应的一组设备 CLI 属性所需的钥匙。设备上定义的配置 ID 必须与 Inspur 配置引擎中，为相应的设备所定义的配置 ID 相同。

配置 ID 在启动时即固定下来，在设备重新启动前无法修改，即使用户重新配置了设备的主机名，配置 ID 也不会受到影响。

设备 ID

事件总线上的每台需要配置的设备都有一个唯一的设备 ID，它类似于设备的源地址，这样设备就可以被定义为总线上的具体目的地了。

最初的设备 ID 是由设备的 Inspur INOS 主机名定义的。但设备 ID 是可变的，它的用法与和设备相邻的事件网关相关。

事件总线上的逻辑 Inspur INOS 端点是内嵌在事件网关中的，事件网关会以设备代理的角色发挥功能。对于事件总线来说，事件网关会代表设备及其相应的设备 ID。

设备会在自己成功连接到事件网关后，立即向事件网关告知自己的主机名。在每次连接建立时，事件网关都会把设备 ID 值和 Inspur INOS 主机名组合在一起。在与设备的连接终结前，事件网关都会保留这个设备 ID 值。

主机名和设备 ID

在设备连接到事件网关时，设备 ID 就固定了，哪怕用户重新配置了设备的主机名，设备 ID 也不会受到影响。

当用户更改设备上的设备主机名时，唯一一种能够刷新设备 ID 的做法是中断设备与事件网关之间的连接。“相关主题”中给出了刷新设备 ID 的指导。

在重新建立连接时，设备会向事件网关发送自己重新配置后的主机名。事件网关则会使用这个新的值来重新定义设备 ID。

注意： 在使用 Inspur 配置引擎的用户界面时，用户必须首先在设备 ID 字段中设置主机名值，而设备会在*之后*，而不是*之前*，获得主机名值；并且用户必须重新初始化 Inspur INOS CNS 代理的配置。否则后续的一部分配置命令操作将会出现问题。

主机名、设备 ID 和配置 ID

在单机模式中，当用户为一台设备设置了主机名值之后，配置服务器会在向改主机名发送事件时，将这个主机名作为设备 ID 使用。如果用户没有设置主机名，配置服务器会把事件发送到设备的 cn=<*value*>。

在服务器模式中不使用主机名。在这个模式中，总是使用唯一的设备 ID 属性，来发送总线上的事件。如果用户没有设置这个属性，则无法对设备进行更新。

这些属性以及其他相关的属性（标记数值对）是在 Inspur 配置引擎上运行 **Setup** 的过程中进

行设置的。

Inspur INOS CNS 代理

设备通过使用 CNS 事件代理特性，能够发布和订阅事件总线上的事件，并与 Inspur INOS CNS 代理协同工作。这些代理是内嵌在设备的 Inspur INOS 软件中的，使设备能够实现连接和自动配置。

初始配置

当设备第一次启动时，它会向网络中发送广播的动态主机配置协议（DHCP）请求，来尝试获得 IP 地址。假设在这个子网上没有部署 DHCP 服务器，分布层设备会充当 DHCP 中继代理，把请求转发给 DHCP 服务器。在收到请求后，DHCP 服务器会为这台新设备分配 IP 地址，在发送给 DHCP 中继代理的单播应答消息中，还包括简单文件传输协议（TFTP）服务器 IP 地址、获得启动配置文件的路径，以及默认网关 IP 地址。DHCP 中继代理会把应答转发给设备。

设备会自动（默认）在接口 VLAN 1 上配置服务器分配的 IP 地址，并从 TFTP 服务器那里下载启动配置文件。在成功下载了启动配置文件后，设备会在自己的运行配置中加载这个文件。

Inspur INOS CNS 代理会使用适当的配置 ID 和事件 ID 来初始化与配置引擎之间的连接。配置引擎会把这个配置 ID 映射到一个模板中，然后把完整的配置文件下载到设备中。

下图展示出一个网络配置示例，描绘了使用基于 DHCP 的自动配置功能，获取初始的启动配置文件的环境。

图 72：初始配置

Configuration Engine	配置引擎
TFTP server	TFTP 服务器
DHCP server	DHCP 服务器
Distribution layer	分布层
DHCP relay agent default gateway	DHCP 中继代理 默认网关
Access layer	接入层

switches	交换机

增量（部分）配置

在网络运行起来后，用户可以使用 Inspur INOS CNS 代理来添加新的服务。它可以向设备发送增量（部分）配置。事件网关可以把实际的配置作为事件负载进行发送（推送操作），或者也可以由单个事件触发设备发起拉取操作。

设备可以在应用配置前，检查配置语法是否正确。如果语法正确，设备会应用这个增量配置，并向配置服务器发布一个事件，表示自己已成功应用配置。如果设备没能应用增量配置，它会通过发布一个事件来展示错误状态。当设备应用了增量配置后，它可以把配置写入非易失性随机访问存储器（NVRAM）中，或者等待相关信令，在收到后需事件时再进行保存。

同步配置

当设备收到一个配置时，它可以根据是否收到写入信令（Write-Signal）事件，来决定是否推迟应用这个配置。写入信令事件会告诉设备不要把这个更新的配置写入它的 NVRAM 中。设备会把这个更新的配置作为自己的运行配置。这样做能够保证在将配置保存到 NVRAM 中（以便下次重启后使用）之前，设备的配置就能够同步其他网络活动。

自动 CNS 配置

要想为设备启用自动 CNS 配置，用户必须首先达成这部分中列出的先决条件。在达成了这些条件后，再给设备加电。在看到 **setup** 提示时什么都不要操作；设备会以初始化配置启动。在将完整的配置文件加载到设备上之后，用户无需再进行任何其他的操作。

有关初始化配置的更多信息，用户可以参考“相关主题”。

表 69：启用自动配置的先决条件

设备	**必需配置**
接入层设备	出厂默认（没有配置文件）
分布层设备	• IP Helper 地址 • 启用 DHCP 中继代理 [2] • IP 路由（如需作为默认网关使用）
DHCP 服务器	• IP 地址分配

	• TFTP 服务器 IP 地址 • TFTP 服务器上的启动配置文件路径 • 默认网关 IP 地址
TFTP 服务器	• 启动配置文件，其中包含 CNS 配置命令，并通过这些命令使设备能够与配置引擎进行通信 • 确定需要进行配置的设备要使用设备 MAC 地址或序列号（代替默认主机名）来生成配置 ID 和事件 ID • 配置 CNS 事件代理把配置文件推送给设备
CNS 配置引擎	为每种类型的设备创建一个或多个模板，把设备的配置 ID 映射到模板

[2] 只有当 DHCP 服务器与客户端不属于同一个子网时，才需要使用 DHCP 代理。

如何实施配置引擎的配置

启用 CNS 事件代理

注释： 在启用 CNS 配置代理之前，用户必须先在设备上启用 CNS 事件代理。

用户可以按照以下步骤，在设备上启用 CNS 事件代理。

总步骤

1. enable

2. configure terminal

3. cns event {*hostname* | *ip-address*} [*port-number*] [[**keepalive** *seconds retry-count*] [**failover-time** *seconds*] [**reconnect-time** *time*] | **backup**]

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的

步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**cns event** {*hostname* | *ip-address*} [*port-number*] [[**keepalive** *seconds retry-count*] [**failover-time** *seconds*] [**reconnect-time** *time*] | **backup**] **示例：** `Device(config)# cns event 10.180.1.27 keepalive 120 10`	启用事件代理并输入网关参数。 • 在{*hostname* | *ip-address*}部分输入事件网关的主机名或 IP 地址 • （可选）在 *port-number* 部分输入事件网关的端口号。默认端口号是 11011 • （可选）在 **keepalive** *seconds* 部分输入让设备发送存活消息的时间间隔。在 *retry-count* 部分输入让设备重复发送存活消息的次数，在此之后连接终结。这两个参数的默认值都是 0 • （可选）在 **failover-time** *seconds* 部分输入在设备重新连接事件网关前，让设备等待的最大时间间隔 • （可选）输入 **backup** 表示这是备用网关（如果忽略这个关键字，表示这是主用网关） **注释：** 尽管在命令行的帮助信息中可以看到 **encrypt** 和 **clock-timeout** *time* 关键字，但设备实际并不支持
步骤 4	**end** **示例：**	返回特权 EXEC 模式

	Device(config)# **end**	
步骤 5	**show running-config** **示例：** Device# **show running-config**	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

接下来做什么？

要想检查有关事件代理的信息，用户需要在特权 EXEC 模式中使用命令 **show cns event connections**。

要想禁用 CNS 事件代理，用户需要在全局配置模式中使用命令 **no cns event** {*ip-address* | *hostname*}。

启用 Inspur INOS CNS 代理

用户可以按照以下步骤，在设备上启用 Inspur INOS CNS 代理。

在开始前

在启用这个代理前，用户必须在设备上启用 CNS 事件代理。

总步骤

1. enable

2. configure terminal

3. cns config initial {*hostname* | *ip-address*} [*port-number*]

4. cns config partial {*hostname* | *ip-address*} [*port-number*]

5. end

6. show running-config

7. copy running-config startup-config

8. 开启设备上的 INOS CNS 代理

具体步骤

	命令或操作	**目的**

步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**cns config initial** {*hostname* | *ip-address*} [*port-number*] **示例：** `Device(config)# cns config initial 10.180.1.27 10`	启用 Inspur INOS CNS 代理并输入配置服务器参数。 • 在{*hostname* | *ip-address*}部分输入配置服务器的主机名或 IP 地址 • （可选）在 *port-number* 部分输入配置服务器的端口号 这条命令负责启用 Inspur INOS CNS 代理，并在设备上初始化其初始配置
步骤 4	**cns config partial** {*hostname* | *ip-address*} [*port-number*] **示例：** `Device(config)# cns config partial 10.180.1.27 10`	启用 Inspur INOS CNS 代理并输入配置服务器参数。 • 在{*hostname* | *ip-address*}部分输入配置服务器的主机名或 IP 地址 • （可选）在 *port-number* 部分输入配置服务器的端口号 这条命令负责启用 Inspur INOS CNS 代理，并在设备上初始化部分配置
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息

步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中
步骤 8	在设备上开启 Inspur INOS CNS 代理	

接下来做什么？

用户现在可以使用 Inspur 配置引擎，从远端向设备发送增量更新。

为 INOS CNS 代理启用初始配置

用户可以按照以下步骤，在设备上启用 CNS 配置代理并初始化其初始配置。

总步骤

1. enable

2. configure terminal

3. cns template connect *name*

4. cli *config-text*

5. 重复步骤 3 和步骤 4，配置其他 CNS 连接模板

6. exit

7. cns connect *name* [**retries** *number*] [**retry-interval** *seconds*] [**sleep** *seconds*] [**timeout** *seconds*]

8. discover {**controller** *controller-type* | **dlci** [**subinterface** *subinterface-number*] | **interface** [*interface-type*] | **line** *line-type*}

9. template name [*...name*]

10. 重复步骤 8 和步骤 9，在 CNS 连接配置文件中，指定更多的接口参数和 CNS 连接模板

11. exit

12. hostname *name*

13. iproute *network-number*

14. cns id *interface num* {**dns-reverse** | **ipaddress** | **mac-address**} [**event**] [**image**]

15. cns id {**hardware-serial** | **hostname** | **string** *string* | **udi**} [**event**] [**image**]

16. cns config initial {*hostname* | *ip-address*} [*port-number*] [**event**] [**no-persist**] [**page** *page*] [**source** *ip-address*] [**syntax-check**]

17. end

18. show running-config

19. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**cns template connect** *name* **示例：** `Device(config)# cns template connect template-dhcp`	进入 CNS 模板连接配置模式并指定 CNS 连接模板的名称
步骤 4	**cli** *config-text* **示例：** `Device(config-tmpl-conn)# cli ip address dhcp`	在 CNS 连接模板中输入一条命令。重复这个步骤，在模板中输入每条命令
步骤 5	重复步骤 3 和步骤 4，来配置其他 CNS 连接模板	
步骤 6	**exit** **示例：** `Device(config-tmpl-conn)# exit`	返回全局配置模式
步骤 7	**cns connect** *name* [**retries** *number*] [**retry-interval** *seconds*] [**sleep** *seconds*] [**timeout** *seconds*] **示例：**	进入 CNS 连接配置模式，指定 CNS 连接配置文件的名称，并定义配置文件中的参数。设备会使用 CNS 连接配置文件来连接配置引擎。 • 在 *name* 部分输入 CNS 连接配置

	Device(config)# **cns connect dhcp**	文件的名称 • （可选）在 **retries** *number* 部分输入连接重试次数。取值范围是 1 至 30。默认值为 3 • （可选）在 **retry-interval** *seconds* 部分输入下一次尝试连接配置引擎的时间间隔。取值范围是 1 至 40 秒。默认值为 10 秒 • （可选）在 **sleep** *seconds* 部分输入首次尝试连接之前的总时间。取值范围是 0 至 250 秒。默认值为 0 秒 • （可选）在 **timeout** *seconds* 部分输入尝试连接结束时的总时间。取值范围是 10 至 2000 秒。默认值为 120
步骤 8	**discover** {**controller** *controller-type* | **dlci** [**subinterface** *subinterface-number*] | **interface** [*interface-type*] | **line** *line-type*} **示例：** Device(config-cns-conn)# **discover interface gigabitethernet**	在 CNS 连接配置文件中指定接口参数。 • 在 **controller** *controller-type* 部分输入控制器类型 • 在 **dlci** 部分输入启用的数据链路连接识别符（DLCI） （可选）在 **subinterface** *subinterface-number* 部分指定点到点子接口编号，用其来搜索启用的 DLCI • 在 **interface** [*interface-type*]部分输入接口类型 • 在 **line** *line-type* 部分输入线路类型
步骤 9	**template** *name* [… *name*]	在 CNS 连接配置文件中指定一系列要被应用到设备配置的模板名称。用

	示例： `Device(config-cns-conn)#` **`template template-dhcp`**	户可以指定多个模板
步骤 10	重复步骤 8 和步骤 9，在 CNS 连接配置文件中指定更多接口参数和 CNS 连接模板	
步骤 11	**exit** **示例：** `Device(config-tmpl-conn)#` **`exit`**	返回全局配置模式
步骤 12	**hostname** *name* **示例：** `Device(config)#` **`hostname device1`**	输入设备的主机名
步骤 13	**ip route** *network-number* **示例：** `RemoteDevice(config)` **`ip route 172.28.129.22 255.255.255.255 11.11.11.1`**	（可选）配置去往配置引擎的静态路由，在 *network-number* 部分设置配置引擎的 IP 地址
步骤 14	**cns id** *interface num* {**dns-reverse** | **ipaddress** | **mac-address**} [**event**] [**image**] **示例：** `RemoteDevice(config)#` **`cns id GigabitEthernet1/0/1 ipaddress`**	（可选）设置配置引擎使用的唯一事件 ID 或配置 ID。如果用户输入这条命令，就不要输入命令 **cns id** {**hardware-serial** | **hostname** | **string** *string* | **udi**} [**event**] [**image**]。 • 在 *interface num* 部分输入接口类型。比如 ethernet、group-async、loopback 或 virtual-template。这个参数指定了设备应该使用哪个接口的 IP 地址或 MAC 地址来定义为一 ID • 在 {**dns-reverse** | **ipaddress** |

		mac-address} 部分输入 **dns-reverse** 表示提取主机名并将其作为唯一 ID，输入 **ipaddress** 表示使用 IP 地址，输入 **mac-address** 表示使用 MAC 地址作为唯一 ID • （可选）输入 **event** 来设置事件 ID 使用的 ID 值，以此来标识设备 • （可选）输入 **image** 来设置镜像 ID 使用的 ID 值，以此来标识设备 **注释：** 如果用户同时忽略了关键字 **event** 和 **image**，系统就会使用镜像 ID 来标识设备
步骤 15	**cns id** {**hardware-serial** | **hostname** | **string** *string* | **udi**} [**event**] [**image**] **示例：** RemoteDevice(config)# **cns id hostname**	（可选）设置配置引擎使用的唯一事件 ID 或配置 ID。如果用户输入这条命令，就不要输入命令 **cns id** *interface num* {**dns-reverse** | **ipaddress** | **mac-address**} [**event**] [**image**]。 • 在{**hardware-serial** | **hostname** | **string** *string* | **udi**}部分输入 **hardware-serial** 表示把设备序列号设置为唯一 ID，输入 **hostname**（默认）表示将设备主机名作为唯一 ID，输入 **string** *string* 表示使用自定义字符串作为唯一 ID，或者输入 **udi** 表示设置唯一设备识别符（UDI）作为唯一 ID
步骤 16	**cns config initial** {*hostname* | *ip-address*} [*port-number*] [**event**] [**no-persist**] [**page** *page*] [**source** *ip-address*] [**syntax-check**]	启用 Inspur INOS 代理并初始化一个初始配置。 • 在{*hostname* | *ip-address*}部分输

	示例： `RemoteDevice(config)#` **`cns config initial 10.1.1.1 no-persist`**	入配置服务器的主机名或 IP 地址 • （可选）在 *port-number* 部分输入配置服务器的端口号。默认端口号为 80 • （可选）当配置完成时，为配置成功、配置失败或告警消息启用 **event** • （可选）使用 **no-persist** 可以实现自动把配置写入 NVRAM，效果与使用全局配置命令 **cns config initial** 相同。如果用户没有输入关键字 **no-persist**，也可以配置 **cns config initial**，把配置自动写入 NVRAM • （可选）在 **page** *page* 部分输入初始配置的 Web 页面。默认值为/Config/config/asp • （可选）将 **source** *ip-address* 作为源 IP 地址使用 • （可选）输入 **syntax-check** 后启用语法检查功能 **注释：** 尽管在命令行的帮助信息中可以看到 **encrypt**、**status** *url* 和 **inventory** 关键字，但设备实际并不支持
步骤 17	**end** **示例：** `Device(config)#` **`end`**	返回特权 EXEC 模式
步骤 18	**show running-config**	检查用户输入的信息

	示例： `Device# show running-config`	
步骤 19	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

接下来做什么？

要想验与证配置代理相关的信息，用户需要在特权 EXEC 模式中使用命令 **show cns config connections**。

要想禁用 CNS Inspur INOS 代理，用户需要使用全局配置命令 **no cns config initial** {*ip-address* | *hostname*}。

刷新设备 ID

用户可以按照以下步骤，当设备上的主机名变更时，刷新设备 ID。

总步骤

1. enable

2. show cns config connections

3. 确保 CNS 事件代理已经正确连接到事件网关

4. show cns event connections

5. 记录步骤 4 中有关当前连接的相关信息。用户会在接下来的步骤中用到 IP 地址和端口号

6. configure terminal

7. no cns event *ip-address port-number*

8. cns event *ip-address port-number*

9. end

10. 通过命令 **show cns event connections** 的输出信息，用户需要确保已经重新建立了设备与事件代理之间的连接

11. show running-config

12. copy running-config startup-config

具体步骤

	命令或操作	**目的**

步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**show cns config connections** **示例：** `Device# show cns config connections`	查看 CNS 事件代理是否正在连接网关、已连接或处于活跃状态，并且查看事件代理使用的网关 IP 地址及其端口号
步骤 3	确保 CNS 事件代理已经正确连接到事件网关	查看命令 **show cns config connections** 中的下列输出信息： • 连接是活跃的 • 连接使用的是当前配置的设备主机名。通过接下来的步骤，设备 ID 将会刷新为使用新的主机名
步骤 4	**show cns event connections** **示例：** `Device# show cns event connections`	查看用户设备的事件连接信息
步骤 5	记录步骤 4 中有关当前连接的相关信息。用户会在接下来的步骤中用到 IP 地址和端口号	
步骤 6	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 7	**no cns event** *ip-address port-number* **示例：** `Device(config)# no cns event`	指定用户在步骤 5 中记录的 IP 地址和端口号。 这条命令会断开设备和事件网关之间的连接。必须先断开连接，之后再

	172.28.129.22 102	重新建立连接，这样才能刷新连接使用的设备 ID
步骤 8	**cns event** *ip-address port-number* **示例：** Device(config)# **cns event 172.28.129.22 2012**	指定用户在步骤 5 中记录的 IP 地址和端口号。 这条命令会重新建立设备和事件网关之间的连接。
步骤 9	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 10	通过命令 **show cns event connections** 的输出信息，用户需要确保已经重新建立了设备与事件代理之间的连接	
步骤 11	**show running-config** **示例：** Device# **show running-config**	检查用户输入的信息
步骤 12	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

为 Inspur INOS CNS 代理启用部分配置

用户可以按照以下步骤，在设备上启用 Inspur INOS CNS 代理，并初始化一个部分配置。

总步骤

1. enable

2. configure terminal

3. cns config partial {*ip-address* | *hostname*} [*port-number*] [**source** *ip-address*]

4. end

5. show running-config

6. copy running-config startup-config

具体配置

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**cns config partial** {*ip-address* | *hostname*} [*port-number*] [**source** *ip-address*] **示例：** `Device(config)# cns config partial 172.28.129.22 2013`	启用配置代理并初始化部分配置。 • 在{*ip-address* | *hostname*}部分输入配置服务器的 IP 地址或主机名 • （可选）在 *port-number* 部分输入配置服务器的端口号。默认端口号为 80 • （可选）输入作为源 IP 地址使用的 **source** *ip-address* **注释：** 尽管在命令行的帮助信息中可以看到关键字 **encrypt**，但设备实际并不支持
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

	示例： `Device# copy running-config startup-config`	

接下来做什么？

要想检查与配置代理相关的信息，用户需要在特权 EXEC 模式中使用命令 **show cns config stats** 或 **show cns config outstanding**。

要想禁用 Inspur INOS 代理，用户需要使用全局配置命令 **no cns config partial** {*ip-address* | *hostname*}。要想取消部分配置，用户需要使用全局配置命令 **cns config cancel**。

监控 CNS 的配置

表 70：与 CNS 相关的 show 命令

命令	目的
show cns config connections `Device# show cns config connections`	显示 CNS Inspur INOS CNS 代理连接的状态
show cns config connections `Device# show cns config connections`	显示有关增量（部分）CNS 配置的信息，其中这些配置已经开始并且还未结束
show cns config stats `Device# show cns config stats`	显示有关 Inspur INOS CNS 代理的信息
show cns event connections `Device# show cns event connections`	显示 CNS 事件代理连接的状态信息
show cns event gateway `Device# show cns event gateway`	显示用户设备的事件代理信息
show cns event stats `Device# show cns event stats`	显示 CNS 事件代理的统计状态信息
show cns event subject `Device# show cns event subject`	显示应用订阅的事件代理项目列表

其他参考资料

相关文档

相关主题	文档名称
配置引擎的初始设置	*Inspur Configuration Engine Installation and Setup Guide, 1.5 for Linux* http://www.icntnetworks.com

错误消息解码器

描述	链接
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

标准和 RFC

标准/RFC	标题
无	-

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	http://www.icntnetworks.com

配置 Cisco 发现协议

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

有关 CDP 的信息

CDP 概述

CDP 是运行在二层（数据链路层）的设备发现协议，Inspur 的所有设备（路由器、网桥、接入服务器、控制器和交换机）都支持 CDP，网络管理应用可以通过 CDP 发现邻居 Inspur 设备。通过使用 CDP，网络管理应用可以学到（运行较低层透明协议的）邻居设备的设备类型和简单网络管理协议（SNMP）代理地址。应用能够通过这个特性向邻居设备发送 SNMP 查询消息。

CDP 能够在所有支持子网接入协议（SNAP）的媒介上运行。由于 CDP 只运行在数据链路层，因此两个运行不同网络层协议的系统也可以学习到对方的信息。

每台配置了 CDP 的设备都会周期性向一个组播地址发送消息，这个通告中至少包含一个它可以接收 SNMP 消息的地址。通告中还包含存活时间或保持时间信息，这个时间间隔表示接收方设备会在这么长时间后丢弃它所收到的 CDP 信息。每台设备还会监听其他设备发送的消息，以此学习邻居设备的信息。

对于设备来说，网络助手（Network Assistant）工具可以通过使用 CDP，以图形的方式展示网络结构。设备可以使用 CDP 找到集群候选者，并维护集群成员和其他设备的信息，

默认的 CDP 配置

下面这个表格中展示了默认的 CDP 配置。

特性	**默认设置**
CDP 全局状态	已启用
CDP 接口状态	已启用
CDP 计时器（数据包更新频率）	60 秒
CDP 保持时间（超时丢弃）	180 秒
CDP 版本 2 通告	已启用

如何配置 CDP

配置 CDP 特征

用户可以配置下列 CDP 特征：

- CDP 更新的频率
- 在丢弃前，维护信息的时长
- 是否发送版本 2 通告

注释： 步骤 3 至步骤 5 都是可选配置，在配置时可以打乱顺序。

用户可以按照以下步骤配置 CDP 特征。

总步骤

1. enable

2. configure terminal

3. cdp timer *seconds*

4. cdp holdtime *seconds*

5. cdp advertise-v2

6. end

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**cdp timer** *seconds* **示例：** `Device(config)# cdp timer 20`	（可选）以秒为单位配置 CDP 更新的传输频率。 取值范围是 5 至 254；默认值为 60 秒
步骤 4	**cdp holdtime** *seconds* **示例：** `Device(config)# cdp holdtime 60`	（可选）设置一个时间值，接收方设备在这段时间内应该保留本端设备发送的信息，超时后丢弃信息。 取值范围是 10 至 255 秒；默认值为 180 秒
步骤 5	**cdp advertise-v2** **示例：** `Device(config)# cdp advertise-v2`	（可选）配置 CDP 来发送版本 2 通告。这是默认状态
步骤 6	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 7	**show running-config** **示例：**	检查用户输入的信息

	Device# **show running-config**	
步骤 8	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

接下来做什么？

用户可以在 CDP 命令前添加关键字 **no**，使配置恢复默认值。

禁用 CDP

CDP 默认是启用的。

注释： 设备集群和其他 Inspur 设备（如 Inspur IP 电话）会有规律地交换 CDP 消息。禁用 CDP 会中断集群发现和设备连接。

用户可以按照以下步骤禁用 CDP 设备发现功能。

总步骤

1. enable

2. configure terminal

3. no cdp run

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** Device> **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式

步骤 3	**no cdp run** **示例：** `Device(config)# no cdp run`	禁用 CDP
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

接下来做什么？

要想使用 CDP，用户必须再次启用它。

启用 CDP

CDP 默认就是启用的。

注释： 设备集群和其他 Inspur 设备（如 Inspur IP 电话）会有规律地交换 CDP 消息。禁用 CDP 会中断集群发现和设备连接。

当 CDP 功能被禁用时，用户可以按照以下步骤启用 CDP 设备发现功能。

在开始前

CDP 必须是禁用状态，否则无法启用。

总步骤

1. enable

2. configure terminal

3. cdp run

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**cdp run** **示例：** `Device(config)# cdp run`	启用 CDP
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

接下来做什么？

用户可以使用命令 **show run all** 来确认 CDP 已被启用。如果用户只使用了命令 **show run**，可

能无法看到 CDP 的启用状态。

在接口上禁用 CDP

在所有支持 CDP 的接口上，CDP 默认都是启用的，接口能够发送和接收 CDP 信息。

注释： 设备集群和其他 Inspur 设备（如 Inspur IP 电话）会有规律地交换 CDP 消息。禁用 CDP 会中断集群发现和设备连接。

用户可以按照以下步骤，在端口上禁用 CDP 设备发现功能。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. no cdp enable

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet1/0/1`	指定用户希望禁用 CDP 的接口，并进入该接口的配置模式
步骤 4	**no cdp enable**	在步骤 3 指定的接口上禁用 CDP

	示例： `Device(config-if)# no cdp enable`	
步骤 5	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

在接口上启用 CDP

在所有支持 CDP 的接口上，CDP 默认都是启用的，接口能够发送和接收 CDP 信息。

注释： 设备集群和其他 Inspur 设备（如 Inspur IP 电话）会有规律地交换 CDP 消息。禁用 CDP 会中断集群发现和设备连接。

如果 CDP 已被禁用，用户可以按照以下步骤，在端口上启用 CDP 设备发现功能。

在开始前

CDP 在该接口上的状态必须为禁用，否则无法启用它。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. cdp enable

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet1/0/1`	指定用户希望禁用 CDP 的接口，并进入该接口的配置模式
步骤 4	**cdp enable** **示例：** `Device(config-if)# cdp enable`	在已禁用 CDP 的接口上启用 CDP
步骤 5	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：**	（可选）把输入的命令保存到配置文件中

	Device# **copy running-config startup-config**	

监控和维护 CDP

表 71：显示 CDP 信息的命令

命令	描述
clear cdp counters	把流量计数器重置为 0
clear cdp table	删除 CDP 表中有关邻居的信息
show cdp	显示全局信息，比如发送数据包的传输频率和保持时间
show cdp entry *entry-name* [**version**] [**protocol**]	显示有关指定邻居的信息。 用户可以通过输入星号（*）查看所有邻居，或者也可以输入邻居名称来查看相应邻居的信息。 用户还可以限制显示内容，比如只查看指定邻居上启用的协议，或者只查看设备上运行的软件版本
show cdp interface [*interface-id*]	显示启用了 CDP 的接口信息。 用户可以只查看某一个接口的信息
show cdp neighbors [*interface-id*] [**detail**]	显示有关邻居的信息，其中包括设备类型、接口类型和编号、保持时间的设置、能力、平台，以及端口 ID。 用户可以只查看某个接口上的邻居信息，或者丰富显示内容，查看详细信息
show cdp traffic	显示 CDP 计数器，其中包括发送和接收的数据包数量，以及校验和的错误数量

其他参考资料

相关文档

相关主题	文档名称
系统管理命令	*Network Management Command Reference, Inspur INOS*

错误消息解码器

描述	链接
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

标准和 RFC

标准/RFC	标题
无	-

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	http://www.icntnetworks.com

配置简单网络管理协议

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

部署 SNMP 的先决条件

支持的 SNMP 版本

本软件版本支持下列版本的 SNMP：

- SNMPv1——简单网络管理协议，它是完整的 Internet 标准，定义在 RFC 1157 文档中；
- SNMPv2C 使用基于团体（Community）字符串的管理架构，代替了 SNMPv2Classic 中基于派别（Party）的管理和安全架构，同时保留了 SNMPv2Classic 中的批量检索并增强了错误处理功能。它拥有以下特性：
  - SNMPv2——简单网络管理协议版本 2，它是 Internet 标准草案，定义在 RFC 1902 至 1907 文档中；
  - SNMPv2C——使用基于团体字符串的管理架构，它是实验性 Internet 协议，定义在 RFC 1901 文档中。
- SNMPv3——SNMP 版本 3，它是具有互操作性的标准协议，定义在 RFC 2273 至 2275 文档中。SNMPv3 提供了安全接入设备的方法，它能够在网络中对数据包执行认证和加密，并包含以下安全特性：

- 消息完整性——确保数据包没有在传输过程中遭到篡改；
- 认证——确定消息来自于合法的源；
- 加密——对消息的内容提供保护，防止未经授权的源对其进行读取。

注释： 要想选择加密功能，用户需要输入关键字 **priv**。

SNMPv1 和 SNMPv2C 都使用基于团体的方式来提供安全保护。管理器团体能够访问代理的 MIB，用户需要使用 IP 地址访问控制列表和密码来对管理器团体进行定义。

SNMPv2C 中包含批量检索功能，还向管理站提供了更详细的错误消息报告。批量检索功能能够在多种表格和大量信息中进行检索，把检索所需的往返数量降到最低。SNMPv2C 增强了错误处理功能，其中包括丰富的错误代码，能够区分不同类型的错误条件；在 SNMPv1 中，这些条件都会报告为一个单独的错误代码。SNMPv2C 中的错误返回代码报告了错误类型。

SNMPv3 同时提供了安全模型和安全级别。安全模型是为用户以及用户所属的组设置的指认证策略。安全级别是指一个安全模型所属的安全级别。通过把安全级别和安全模型结合在一起，就能够在处理 SNMP 数据包时，决定使用哪种安全方式了。各种安全模型包括 SNMPv1、SNMPv2C 和 SNMPv3。

下面这个表格中展示了这些安全模型的特征，并对比了各种安全模型和安全级别的组合方式。

表 72：SNMP 安全模型和安全级别

模型	级别	认证	加密	结果
SNMPv1	无认证无加密	团体字符串	无	使用团体字符串来进行认证
SNMPv2C	无认证无加密	团体字符串	无	使用团体字符串来进行认证
SNMPv3	无认证无加密	用户名	无	使用用户名来进行认证
SNMPv3	有认证无加密	消息摘要 5（MD5）或安全散列算法（SHA）	无	基于 HMAC-MD5 或 HMAC-SHA 算法来提供认证
SNMPv3	有认证有加密	MD5 或 SHA	数据加密标准（DES）或高级加密标准（AES）	基于 HMAC-MD5 或 HMAC-SHA 算法来提供认证。用户能够使用下列加密算法，定

				义自己的安全模型（USM）： • DES 56 比特加密，基于 CBC-DES（DES-56）标准的认证 • 3DES 168 比特加密 • AES 128 比特、192 比特或 256 比特加密

用户必须配置 SNMP 代理，才能使用管理站支持的 SNMP 版本。由于一个代理能够与多个管理器进行通信，因此用户可以配置 SNMP 代理分别使用 SNMPv1、SNMPv2C 或 SNMPv3 与管理器进行通信。

部署 SNMP 的限制条件

版本限制

- SNMPv1 不支持通知 Inform 消息。

有关 SNMP 的信息

SNMP 概述

SNMP 是一项应用层协议，它为管理器和代理之间的通信提供了一种消息格式。SNMP 系统由 SNMP 管理器、SNMP 代理和管理信息库（MIB）构成。SNMP 管理器可以是网络管理系统（NMS）的一部分，比如 Inspur Prime Infrastructure 就是一种 NMS。代理和 MIB 位于设备

上。要想在设备上配置 SNMP，用户需要定义管理器和代理之间的关系。

SNMP 代理中包含 MIB 变量，SNMP 管理器可以请求或更改其中的变量值。管理器可以从代理那里获得一个值，也可以向代理中存入一个值。代理可以从 MIB 中收集数据，MIB 中保存了有关设备参数和网络数据的信息。代理可以对管理器发来的获取或设置数据的请求作出响应。

代理可以向管理器发送未经请求的 Trap 消息。Trap 消息是用来警示 SNMP 管理器的，它说明了网络中的某种状况。Trap 可以通告错误的用户认证、重新启动、链路状态（Up 或 Down）、MAC 地址追踪、TCP 连接关闭、邻居连接断开，或其他重要的事件。

SNMP 管理器功能

SNMP 管理器会使用 MIB 中的信息来执行一些操作，下面这个表格中展示了这些操作：

表 73：SNMP 的操作

操作	**描述**
get-request	检索指定变量的值
get-next-request	从一个表中检索指定变量的值 [3]
get-bulk-request[4]	检索大数据块（比如一个表中的多个行），否则需要传输多个小数据块
get-response	NMS 发送的对于 get-request、get-next-request 和 set-request 请求的响应
set-request	设置指定变量的值
trap	SNMP 代理向 SNMP 管理器发送的未经请求的消息，SNMP 代理会在发生特定事件时进行发送

[3] 在使用这个操作时，SNMP 管理器无需指导具体的变量名称。设备会按顺序在表中查找所需变量。

[4] get-bulk 命令只适用于 SNMPv2 及其后续版本。

SNMP 代理功能

SNMP 代理能够对 SNMP 管理器发出请求作出下列应答：

- 获得（Get）一个 MIB 变量——SNMP 代理使用这个功能对 NMS 发出的请求作应答。SNMP 代理会检索 NMS 请求的 MIB 变量值，然后把这个值发送给 NMS；
- 设置（Set）一个 MIB 变量——SNMP 代理使用这个功能对 NMS 发出的消息作应答。SNMP 代理会把相应的 MIB 变量值设置为 NMS 要求的值。

SNMP 代理还能够发送未经请求的 Trap 消息，以此向 NMS 通知代理上正在发生的重要事件。发送 Trap 的条件包括但不限于以下这些：端口或模块状态改变（Up 或 Down）、生成树拓扑发生变化，以及认证失败。

SNMP 团体字符串

SNMP 使用团体字符串作为内嵌密码，对去往 MIB 对象和功能的访问行为进行认证。为了让 NMS 能够访问设备，NMS 上定义的团体字符串必须与设备上定义的三个团体字符串之一相匹配。

团体字符串可以包含下列属性之一：

- 只读（RO）——为授权的管理站提供 MIB 中所有对象（除团体字符串之外）的读取权限，但不允许写入访问；
- 读写（RW）——为授权的管理站提供 MIB 中所有对象的读写访问权限，但不允许访问团体字符串；
- 在创建集群（Cluster）时，命令设备负责管理成员设备与 SNMP 应用之间的消息交换。网络助手（Network Assistant）软件会把成员设备编号（@esN，其中 N 就是设备编号）附加到命令设备上第一个配置的 RW 和 RO 团体字符串上，并把它们传输给成员设备。

图 73：SNMP 网络

SNMP Manager	SNMP 管理器
Network device	网络设备
SNMP Agent	SNMP 代理

SNMP 通知

SNMP 允许设备在特定事件发生时，向 SNMP 管理器发送通知。设备可以通过 Trap 或 Inform 请求的形式发送 SNMP 通知。在命令语法中，除非命令提供了具体的 traps 或 informs 关键字，否则关键字 traps 表示 Trap 或 Inform，或者同时表示两者。用户可以使用命令 **snmp-server host**，来指定以 Trap 或 Inform 的形式发送 SNMP 通知。

注释： SNMPv1 不支持 Inform 消息。

Trap 消息是不可靠的，因为接收方在收到 Trap 消息后，并不会发送确认消息；发送方无法确认对方是否收到了它发送的 Trap 消息。当 SNMP 管理器收到 Inform 请求时，它会通过 SNMP 响应协议数据单元（PDU）来对这个消息进行确认。如果发送方没有收到响应消息，它就会再次发送这个 Inform 请求。由于 Inform 是可以重新发送的，因此它比 Trap 消息更有可能成

功到达目的地。

正因为Inform消息比Trap消息更加可靠，使用Inform消息也消耗了更多的设备和网络资源。设备在发送 Trap 消息后会立即丢弃这个消息，与此不同的是，设备在发送 Inform 请求后，会将其保存在内存中，直到它收到了有关这个 Inform 请求的响应消息，或者直到请求超时。对于 Trap 消息，设备只会发送一次；但对于 Inform 消息，设备可能会发送或尝试发送多次。尝试的次数越多，对网络中带来的流量和负载也就越大。因此用户在选择 Trap 和 Inform 时，需要在可靠性和资源消耗上进行权衡。如果需要保障 SNMP 管理器能够收到每个通知，就使用 Inform 请求。如果需要着重考量网络中的流量或设备中的内存，而且通知消息也并不是必需的，就使用 Trap。

默认 SNMP 配置

特性	**默认设置**
SNMP 代理	禁用 [5]
SNMP Trap 接收方	无配置
SNMP Trap 消息	未启用，除了 TCP 连接（TTY 线路）的 Trap 消息
SNMP 版本	若没有使用 version 关键字，则默认为版本 1
SNMPv3 认证	若没有输入关键字，则默认为 **noauth**（无认证无加密）安全级别
SNMP 通知类型	若没有指定类型，则默认发送所有通知

[5] 这是设备启动时的默认状态，并且在启动配置中也不包含任何全局配置命令 **snmp-server**。

SNMP 配置指导

如果设备刚刚启动，并且设备启动配置文件中至少有一条全局配置命令 **snmp-server**，设备上就启用了 SNMP 代理。

SNMP *组（Group）*是用来把 SNMP 用户与 SNMP 视图映射在一起的表。SNMP *用户（User）*是 SNMP 组中的成员。SNMP *主机（Host）*是 SNMP Trap 操作的接收方。SNMP *引擎 ID（Engine ID）*是本地或远端 SNMP 引擎的名称。

在配置 SNMP 时，用户需要遵从以下三条指导方针：

- 在配置 SNMP 组时，不要指定通知视图（Notify View）。全局配置命令 **snmp-server host** 会为用户自动生成通知视图，并且把它添加到与用户相关联的组中。在修改组的通知视

图时，会影响与这个组相关联的所有用户；

- 要想配置远端用户，需要指定这个用户所在设备的远端 SNMP 代理所使用的 IP 地址或端口号；
- 在为指定代理配置远端用户时，需要在全局配置命令 **snmp-server engineID** 中，通过 **remote** 选项配置 SNMP 引擎 ID。远端代理的 SNMP 引擎 ID 和用户密码会被拿来计算认证和加密摘要。如果事先没有配置远端引擎 ID，则配置命令无法成功实施；
- 在配置 SNMP Inform 通知时，用户需要首先为 SNMP 数据库中的远端代理配置 SNMP 引擎 ID，然后才能向这个它发送代理请求或 Inform 消息；
- 如果本地用户没有与远端主机进行关联，设备就不会为 **auth**（有认证无加密）和 **priv**（有认证有加密）认证级别发送 Inform 消息；
- 改变 SNMP 引擎 ID 的值会带来严重后果。（在命令行中输入的）用户的密码会基于密码和本地引擎 ID，转换为 MD5 或 SHA 安全摘要。之后这个命令行密码会被丢弃，这是 RFC 2274 文档中的要求。由于有了这个要求，当引擎 ID 的值发生改变时，SNMPv3 用户的安全摘要就会变得不可用，用户需要通过全局配置命令 **snmp-server user** *username* 重新配置 SNMPv3 用户。当引擎 ID 发生变化时，也由于同样的限制因素，用户需要重新配置团体字符串。

如何配置 SNMP

禁用 SNMP 代理

用户可以使用全局配置命令 **no snmp-server**，禁用设备上运行的所有版本的 SNMP 代理（版本 1、版本 2C 和版本 3）。用户可以通过在设置 SNMP 代理时配置的第一条全局配置命令 **snmp-server**，重新启用所有类型的 SNMP 代理功能。Inspur INOS 命令中并没有为启用 SNMP 功能设置单独的命令。

用户可以按照以下步骤禁用 SNMP 代理。

在开始前

SNMP 代理特性当前必须是启用的，用户才能将其禁用。设备中输入的第一条全局配置命令 **snmp-server** 就会启用 SNMP 代理。

总步骤

1. enable

2. configure terminal

3. no snmp-server

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**no snmp-server** **示例：** `Device(config)# no snmp-server`	禁用 SNMP 代理功能
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config`	（可选）把输入的命令保存到配置文件中

	startup-config	

配置团体字符串

用户需要使用团体字符串来定义 SNMP 管理器与代理之间的关系。团体字符串的工作类似于密码，它能够允许管理器访问设备上的代理。用户可以（可选）将下列特征中的一个或多个与字符串结合使用：

- 匹配 SNMP 管理器 IP 地址的访问列表，用户允许这些管理器通过团体字符串访问代理；
- MIB 视图，其中定义了指定团体能够访问的所有 MIB 对象中的一部分；
- 指定团体对于 MIB 对象的读写权限或只读权限。

用户可以按照以下步骤，在设备上配置团体字符串。

总步骤

1. enable

2. configure terminal

3. snmp-server community *string* [**view** *view-name*] [**ro** | **rw**] [*access-list-number*]

4. access-list *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*access-list-number*]	配置团体字符串。 **注释：** @符号是用来界定环境信息的。不要在配置这条命令时，在 SNMP

	示例： `Device(config)#` **`snmp-server community comaccess ro 4`**	团体字符串中使用@符号。 • 在 *string* 部分指定一个字符串，当作密码，允许访问 SNMP 协议。用户可以配置一个或多个任意长度的团体字符串 • （可选）在 **view** 部分指定这个团体能够访问的视图记录 • （可选）如果用户希望授权管理站能够检索 MIB 对象，就配置只读（**ro**）权限；如果用户希望授权管理站能够检索和修改 MIB 对象，就配置读写（**rw**）权限。默认情况下，团体字符串会放行去往所有对象的只读访问 • （可选）在 *access-list-number* 部分指定编号的标准 IP 访问列表，取值范围是 1 至 99 和 1300 至 1999
步骤 4	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] **示例：** `Device(config)#` **`access-list 4 deny any`**	（可选）如果用户在步骤 3 中指定了标准 IP 访问列表，就需要创建一个列表，用户可以根据需要多次重复配置这条命令。 • 在 *access-list-number* 部分输入步骤 3 中指定的访问列表编号 • 关键字 **deny** 会在条件匹配时拒绝访问。关键字 **permit** 会在条件匹配时放行访问 • 在 *source* 部分输入 SNMP 管理器的 IP 地址，也就是用户希望能够通过团体字符串访问代理的管理器 • （可选）在 *source-wildcard* 部分以点分十进制格式，输入与源相匹配的通配符比特。把希望在对比中忽

		略的二进制位设置为 1 要记住，所有访问列表的末尾都有隐含拒绝所有数据包的语句
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

接下来做什么？

要想禁止一个 SNMP 团体的访问权限，用户可以把这个团体的团体字符串设置为空字符（也就是配置团体字符串时不输入任何值）。

要想删除某个团体字符串，可以使用全局配置命令 **no snmp-server**。

用户可以为本地或远端设备上的 SNMP 服务器引擎指定一个名称（引擎 ID）。用户可以通过配置 SNMP 服务器组，把 SNMP 用户映射到 SNMP 视图，也可以把新用户添加到 SNMP 组中。

配置 SNMP 组和用户

用户可以为本地或远端设备上的 SNMP 服务器引擎指定一个名称（引擎 ID）。用户可以通过配置 SNMP 服务器组，把 SNMP 用户映射到 SNMP 视图，也可以把新用户添加到 SNMP 组中。

用户可以按照以下步骤，在设备上配置 SNMP 组和用户。

总步骤

1. enable

2. configure terminal

3. snmp-server engineID {**local** *engineid-string* | **remote** *ip-address* [**udp-port** *port-number*] *engineid-string*}

4. snmp-server group *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}}[**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*]

5. snmp-server user *username group-name* {**remote** *host* [**udp-port** *port*]} {**v1** [**access** *access-list*] | **v2c** [**access** *access-list*] | **v3** [**encrypted**] [**access** *access-list*] [**auth** {**md5** | **sha**} *auth-password*] } [**priv** {**des** | **3des** | **aes** {**128** | **192** | **256**}} *priv-password*]

6. end

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**snmp-server engineID** {**local** *engineid-string* | **remote** *ip-address* [**udp-port** *port-number*] *engineid-string*} **示例：** `Device(config)# snmp-server engineID local 1234`	为本地或远端的 SNMP 副本配置名称。 • 在 *engineid-string* 部分为 SNMP 副本指定最长 24 字符的 ID 字符串。如果引擎 ID 是以 0 结尾的，用户无需指定完整的 24 字符。用户可以只指定到 0 之前的字符。示例中的配置指定的引擎 ID 是 123400000000000000000000 • 如果用户选择了关键字 **remote**，需要在 *ip-address* 部分指定包含远端 SNMP 副本的设备，还可以（可选）设置远端设备上使用的用户数据报协议（UDP）端口。端口默认为 62

步骤 4	**snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}}[**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*] **示例：** `Device(config)#` **`snmp-server group public v2c access lmnop`**	在远端设备上配置新的 SNMP 组。 在 *group-name* 部分指定租的名称。 指定以下安全模型之一： • **v1** 是可用安全级别中提供最低安全保障的级别 • **v2** 是倒数第 2 低的安全模型。它允许传输 Inform 消息，并且是正常宽度的两倍 • **v3** 最安全的选择，要求用户选择以下安全级别之一： **auth**——启用消息摘要 5（MD5）和安全散列算法（SHA）进行数据包认证 **noauth**——启用无认证无加密安全级别。如果用户没有指定关键字，这就是默认设置 **priv**——启用数据加密标准（DES）进行数据包加密（也成为隐私） （可选）在 **read** *readview* 部分输入一个字符串（不超过 64 字符），指定具体视图的名称，在这个视图中，只能读取代理中的内容。 （可选）在 **write** *writeview* 部分输入一个字符串（不超过 64 字符），指定具体视图名称，在这个视图中，能够写入数据和配置代理中的内容。 （可选）在 **notify** *notifyview* 部分输入一个字符串（不超过 64 字符），指定具体视图名称，在这个视图中设置 Notify、Inform 或 Trap 消息。 （可选）在 **access** *access-list* 部分输入一个字符串（不超过 64 字符），指定访

		问列表的名称
步骤 5	**snmp-server user** *username group-name* {**remote** *host* [**udp-port** *port*]} {**v1** [**access** *access-list*] | **v2c** [**access** *access-list*] | **v3** [**encrypted**] [**access** *access-list*] [**auth** {**md5** | **sha**} *auth-password*] } [**priv** {**des** | **3des** | **aes** {**128** | **192** | **256**}} *priv-password*] **示例：** `Device(config)# snmp-server user Pat public v2c`	为 SNMP 组添加一个新用户。 在 *username* 部分输入主机上用户的名称，这台主机连接着代理。 在 *group-name* 部分输入组的名称，也就是希望用户关联的组。 使用关键字 **remote** 指定远端 SNMP 实体，也就是用户所属的 SNMP 实体，指定这个实体的主机名或 IP 地址,（可选）以及 UDP 端口号。默认端口号为 62。 输入 SNMP 版本号（**v1**、**v2c** 或 **v3**）。 在输入 **v3** 后，用户还可以配置以下选项： • **encrypted** 指定密码的加密格式。这个关键字只有当用户配置了 **v3** 时才可以使用 • **auth** 是认证级别设置会话，可以是 HMAC-MD5-96（**md5**）认证级别，也可以是 HMAC-SHA-96（**sha**）认证级别；用户同时还要配置密码字符串 *auth-password*（不超过 64 字符） 在输入 **v3** 后，用户还可以配置隐私（**priv**）加密算法和密码字符串 *priv-password*，使用下列关键字（不超过 64 字符）： • **priv** 指定用户自定义安全模型（USM） • **des** 指定使用 56 比特 DES 算法 • **3des** 指定使用 168 比特 DES 算法 • **aes** 指定使用 DES 算法。用户必须在 128 比特、192 比特或 256 比特

		算法之中选择其一 （可选）在 **access** *access-list* 部分输入一个字符串（不超过 64 字符），指定访问列表的名称
步骤 6	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 7	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 8	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 SNMP 通知

Trap 管理器是一台管理站，负责接收和处理 Trap 消息。Trap 是设备在发生了特定事件后生成的系统告警。默认情况下没有定义 Trap 管理器，设备也不会发送 Trap 消息。运行这个版本 Inspur INOS 的设备可以拥有多个 Trap 管理器（数量不限）。

注释： 有多条命令都在命令语法中使用了关键字 **traps**。除非命令中提供了具体选择：Trap 或 Inform，否则关键字 **traps** 就代表 Trap、Inform 或两者。用户可以使用全局配置命令 **snmp-server host**，来指定发送 SNMP 通知的形式是 Trap 还是 Inform。

用户可以使用全局配置命令 **snmp-server enable traps**，与全局配置命令 **snmp-server host** 相结合，来指定主机接收下表中的通知类型。用户可以启用或禁用这些 Trap 消息，并配置一个 Trap 管理器来接收这些 Trap 消息。

注释： 命令 **snmp-server enable traps** 不支持为设备本地认证生成 Trap 消息。

表 74：设备通知类型

通知类型关键字	描述

bridge	生成 STP 桥接 MIB Trap 消息
cluster	当集群配置发生变化时，生成 Trap 消息
config	当 SNMP 配置发生变化时，生成 Trap 消息
copy-config	当 SNMP 副本配置发生变化时，生成 Trap 消息
cpu threshold	允许与 CPU 相关的 Trap 消息
entity	当 SNMP 实体发生变化时，生成 Trap 消息
envmon	生成环境监控 Trap 消息。用户可以启用以下任意或所有环境 Trap 消息：风扇、关机、状态、电源、温度
flash	生成 SNMP FLASH 通知。在设备栈中，用户可以（可选）为 Flash 的插入和移除状态生成通知，当设备栈中的设备被移除或插入（物理移除、断电或重启）时就会生成 Trap 消息
fru-ctrl	生成实体的现场可更换单元（FRU）控制 Trap 消息。在设备栈中，这个 Trap 消息表示设备栈中的设备被插入或移除
hsrp	为热备份路由器协议（HSRP）的变化生成 Trap 消息
ipmulticast	为 IP 组播路由的变化生成 Trap 消息
mac-notification	为 MAC 地址通知生成 Trap 消息
ospf	为最短路径优先（OSPF）的变化生成 Trap 消息。用户可以启用下列任意或全部 Trap 消息：Inspur 指定、错误、链路状态通告、速率限制、重传和状态变化
pim	为协议无关组播（PIM）的变化生成 Trap 消息。用户可以启用下列任意或全部 Trap 消息：不合法的 PIM 消息、邻居变化和汇集点（RP）映射的变化
port-security	生成 SNMP 端口安全 Trap 消息。用户可以以秒为单位设置 Trap 最大传输速率。取值范围是 0 至 1000；默认值为 0，表示没有限速。 **注释：** 在使用通知类型 **port-security** 配置 Trap 消息时，用户需要首先配置端口安全特性，然后配置端口安全 Trap 速率： **1. snmp-server enable traps port-security** **2. snmp-server enable traps port-security trap-rate** *rate*
snmp	为 SNMP 类型的通知生成 Trap 消息，其中包括认证、冷启动、热启动、链路 Up 或链路 Down
storm-control	为 SNMP 风暴控制生成 Trap 消息。用户还可以以分钟为单位设置 Trap 最大传输速率。取值范围是 0 至 1000；默认值为 0（没有限制；每当条

	件满足时都发送 Trap 消息）
stpx	生成 SNMP STP 扩展 MIB Trap 消息
syslog	生成 SNMP 系统日志 Trap 消息
tty	为 TCP 连接生成 Trap 消息。默认这个 Trap 消息是启用的
vlan-mumbership	为 SNMP VLAN 成员关系变化生成 Trap 消息
vlancreate	生成 SNMP VLAN 创建 Trap 消息
vlandelete	生成 SNMP VLAN 删除 Trap 消息
vtp	为 VLAN 干道协议（VTP）的变化生成 Trap 消息

用户可以按照以下步骤，配置设备向主机发送 Trap 或 Inform 消息。

总步骤

1. enable

2. configure terminal

3. snmp-server engineID remote *ip-address engineid-string*

4. snmp-server user *username group-name* {**remote** *host* [**udp-port** *port*]} {**v1** [**access** *access-list*] | **v2c** [**access** *access-list*] | **v3** [**encrypted**] [**access** *access-list*] [**auth** {**md5** | **sha**} *auth-password*] }

5. snmp-server group *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*]

6. snmp-server host *host-addr* [**informs** | **traps**] [**version** {**1** | **2c** | **3** {**auth** | **noauth** | **priv**}}] *community-string* [*notification-type*]

7. snmp-server enable traps *notification-types*

8. snmp-server trap-source *interface-id*

9. snmp-server queue-length *length*

10. snmp-server trap-timeout *seconds*

11. end

12. show running-config

13. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码

步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**snmp-server engineID remote** *ip-address engineid-string* **示例：** `Device(config)# snmp-server engineID remote 192.180.1.27 00000063000100a1c0b4011b`	为远端主机指定引擎 ID。
步骤 4	**snmp-server user** *username group-name* {**remote** *host* [**udp-port** *port*]} {**v1** [**access** *access-list*] | **v2c** [**access** *access-list*] | **v3** [**encrypted**] [**access** *access-list*] [**auth** {**md5** | **sha**} *auth-password*] } **示例：** `Device(config)# snmp-server user Pat public v2c`	配置 SNMP 用户，与步骤 3 中创建的远端主机相关联。 **注释：** 用户不能先为远端用户配置一个地址，然后再为远端主机配置引擎 ID。否则用户会看到错误消息，并且命令也不会执行
步骤 5	**snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}}[**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*] **示例：** `Device(config)# snmp-server group public v2c access lmnop`	配置 SNMP 组。
步骤 6	**snmp-server host** *host-addr* [**informs** | **traps**] [**version** {**1** | **2c** | **3** {**auth** | **noauth** | **priv**}}] *community-string*	指定接收 SNMP Trap 消息的主机。 在 *host-addr* 部分指定主机（目标接收方）名称或 IP 地址。

	[*notification-type*] **示例：** `Device(config)#` **`snmp-server host 203.0.113. comaccess snmp`**	（可选）指定 **traps**（默认），向主机发送 SNMP Trap 消息 （可选）指定 **informs**，向主机发送 SNMP Inform 消息 （可选）指定 SNMP 版本 **version**（**1**、**2c** 或 **3**）。SNMPv1 不支持发送 Inform 消息 （可选）在配置版本 3 时，用户可以选择安全级别：**auth**、**noauth** 或 **priv** **注释：** 只有当设备装安装了加密软件版本时才能使用关键字 **priv** 在配置 **version 1** 或 **version 2** 时，用户可以在 *community-string* 部分指定作为密码使用的团体字符串，与通知消息一起发送。在配置 **version 3** 时，用户可以输入 SNMPv3 用户名。 @符号是用来界定环境信息的。不要在配置这条命令时，在 SNMP 团体字符串中使用@符号。 （可选）在 *notification-type* 部分使用上表中列出的关键字。如果没有指定具体类型，表示发送所有类型的通知
步骤 7	**snmp-server enable traps** *notification-types* **示例：** `Device(config)#` **`snmp-server enable traps snmp`**	使设备发送 Trap 或 Inform 消息，并指定发送的通知类型。具体的通知类型见上表，或者使用命令 **snmp-server enable traps ?**。 要想启用多种类型的 Trap 消息，用户必须为每种 Trap 类型单独输入一条 **snmp-server enable traps** 命令。 **注释：** 在使用通知类型 **port-security** 配置 Trap 消息时，用户要首先配置端口安全 Trap，然后配置端口安全 Trap

		速率： **1. snmp-server enable traps port-security** **2 snmp-server enable traps port-security trap-rate** *rate*
步骤 8	**snmp-server trap-source** *interface-id* **示例：** `Device(config)# snmp-server trap-source GigabitEthernet1/0/1`	（可选）指定源接口，这个接口为 Trap 消息提供源 IP 地址。这条命令也设置了 Inform 消息的源 IP 地址
步骤 9	**snmp-server queue-length** *length* **示例：** `Device(config)# snmp-server queue-length`	（可选）为每个 Trap 主机建立消息队列长度。取值范围是 1 至 1000；默认值为 10
步骤 10	**snmp-server trap-timeout** *seconds* **示例：** `Device(config)# snmp-server trap-timeout 60`	（可选）定义重新发送 Trap 消息的时间间隔。取值范围是 1 至 1000；默认值为 30 秒
步骤 11	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 12	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 13	**copy running-config startup-config** **示例：**	（可选）把输入的命令保存到配置文件中

	Device# **copy running-config startup-config**	

接下来做什么？

命令 **snmp-server host** 指定了由哪台主机负责接收通知消息。命令 **snmp-server enable traps** 在全局（为 Trap 和 Inform）为指定通知类型启用了通知特性。要想让主机收到 Inform 消息，用户必须为主机配置命令 **snmp-server host informs**，然后使用命令 **snmp-server enable traps** 在全局启用 Inform 消息。

要想让某台主机不再接收 Trap 消息，用户需要使用全局配置命令 **no snmp-server host** *host*。命令 **no snmp-server host** 中带有关键字 **no**，会为主机禁用 Trap 消息，但 Inform 消息不受影响。要向禁用 Inform 消息，用户需要使用全局配置命令 **no snmp-server host informs**。要想禁用某种 Trap 类型，用户需要使用全局配置命令 **no snmp-server enable traps** *notification-types*。

设置代理联系和位置信息

用户可以按照以下步骤，来设置 SNMP 代理的系统联系和位置信息，用户通过配置文件能够访问这些描述信息。

总步骤

1. enable

2. configure terminal

3. snmp-server contact *text*

4. snmp-server location *text*

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** Device> **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device# configure terminal`	
步骤 3	**snmp-server contact** *text* 示例： `Device(config)# snmp-server contact Dial System Operator at beeper 21555`	设置系统联系信息
步骤 4	**snmp-server location** *text* 示例： `Device(config)# snmp-server location Building 3/Room 222`	设置系统位置信息
步骤 5	**end** 示例： `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** 示例： `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

通过 SNMP 限制 TFTP 服务器的使用

用户可以按照以下步骤，通过 SNMP 调用访问列表，限制使用 TFTP 服务器进行保存和加载配置文件的行为。

总步骤

1. enable

2. configure terminal

3. snmp-server tftp-server-list *access-list-number*

4. access-list *access-list-number* {**deny** | **permit**} **source** [*source-wildcard*]

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**snmp-server tftp-server-list** *access-list-number* **示例：** `Device(config)# snmp-server tftp-server-list 44`	通过 SNMP 调用访问列表，来限制使用 TFTP 进行配置文件复制工作。 在 *access-list* 部分输入标准 IP 访问李彪的编号，取值范围是 1 至 99 和 1300 至 1999
步骤 4	**access-list** *access-list-number* {**deny** | **permit**} **source** [*source-wildcard*] **示例：** `Device(config)# access-list 44 permit 10.1.1.2`	创建标准访问列表，用户可以按照需要多次重复配置这条命令。 在 *access-list-number* 部分输入步骤 3 中指定的访问列表编号。 关键字 **deny** 会在条件匹配时拒绝访问。关键字 **permit** 会在条件匹配时放行访问。 在 *source* 部分输入能够访问设备的 TFTP 服务器的 IP 地址。

		（可选）在 *source-wildcard* 部分以点分十进制格式，输入与源相匹配的通配符比特。把希望在对比中忽略的二进制位设置为 1。 要记住，所有访问列表的末尾都有隐含拒绝所有数据包的语句
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

为 SNMP 配置 Trap 标记

总步骤

1. configure terminal

2. trapflags ap { **interfaceup** | **register**}

3. trapflags client {**dot11** | **excluded**}

4. trapflags dot11-security {**ids-sig-attack** | **wep-decrypt-error**}

5. trapflags mesh

6. trapflags rogueap

7. trapflags rrm-params {**channels** | **tx-power**}

8. trapflags rrm-profile {**coverage** | **interference** | **load** | **noise**}

9. end

具体配置

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**trapflags ap** {**interfaceup** | **register**} **示例：** `Device(config)# trapflags ap interfaceup`	允许发送与 AP 相关的 Trap 消息。在这条命令前添加关键字 **no** 来禁用 Trap 标记。 • **interfaceup**——当 Inspur AP 接口（A 或 B）变为 Up 状态时发送 Trap 消息 • **register**——当 Inspur AP 上注册了 Inspur 设备时发送 Trap 消息
步骤 3	**trapflags client** {**dot11** | **excluded**} **示例：** `Device(config)# trapflags client excluded`	允许发送与客户端相关的 802.11 Trap 消息。在这条命令前添加关键字 **no** 来禁用 Trap 标记。 • **dot11**——为客户端启用 802.11 Trap 消息 • **excluded**——为客户端启用拒绝 Trap 消息
步骤 4	**trapflags dot11-security** {**ids-sig-attack** | **wep-decrypt-error**} **提示：** `Device(config)# trapflags dot11-security wep-decrypt-error`	启用与 802.11 安全相关的 Trap 消息。在这条命令前添加关键字 **no** 来禁用 Trap 标记。 • **ids-sig-attack**——启用 IDS 签名攻击 Trap 消息 • **wep-decrypt-error**——为客户端启用 WEP 解密错误 Trap 消息
步骤 5	**trapflags mesh** **示例：** `Device(config)# trapflags mesh`	为 Mesh 启用 Trap 消息。在这条命令前添加关键字 **no** 来禁用 Trap 标记。

步骤 6	**trapflags rogueap** **示例：** `Device(config)# trapflags rogueap`	为 Rogue AP 检测启用 Trap 消息。在这条命令前添加关键字 **no** 来禁用 Trap 标记。
步骤 7	**trapflags rrm-params {channels | tx-power}** **示例：** `Device(config)# trapflags rrm-params tx-power`	启用与更新相关的 RRM 参数 Trap 消息。在这条命令前添加关键字 **no** 来禁用 Trap 标记。 • **channels**——当 RF 管理器自动为 Inspur AP 接口更改新到号码时发送 Trap 消息 • **tx-power**——当时 RF 管理器自动为 Inspur AP 接口更改发送功率时发送 Trap 消息
步骤 8	**trapflags rrm-profile {coverage | interference | load | noise}** **注释：** `Device(config)# trapflags rrm-profile interference`	启用与 RRM 配置文件相关的 Trap 消息。在这条命令前添加关键字 **no** 来禁用 Trap 标记。 • **coverage**——当 RF 管理器中维护的范围配置文件失效时发送 Trap 消息 • **interference**——当 FR 管理器中维护的干扰配置文件失效时发送 Trap 消息 • **load**——当 FR 管理器中维护的加载配置文件失效时发送 Trap 消息 • **noise**——当 FR 管理器中维护的噪声配置文件失效时发送 Trap 消息
步骤 9	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式

监控 SNMP 状态

要想查看 SNMP 的输入和输出状态统计信息，其中包括用户输入的错误团体字符串次数、错误和被请求过的变量，用户需要使用特权 EXEC 命令 **show snmp**。用户也可以使用下面这个表格中列出的其他特权 EXEC 命令来查看 SNMP 信息。

表 75：显示 SNMP 信息的命令

命令	**目的**
show snmp	显示 SNMP 的统计状态信息
	显示设备上配置的本地 SNMP 引擎和所有远端引擎信息
show snmp group	显示网络中每个 SNMP 组的信息
show snmp pending	显示暂缓处理的 SNMP 请求信息
show snmp sessions	显示当前的 SNMP 会话信息
show snmp user	显示 SNMP 用户表中每个 SNMP 用户名称的信息。 **注释：** 用户必须使用这条命令来查看 **auth | noauth | priv** 模式的 SNMPv3 配置信息。命令 **show running-config** 中不会显示这些信息

SNMP 示例

下面这个示例中展示了如何启用所有版本的 SNMP。这个配置允许任意 SNMP 管理器使用团体字符串 *public*，以只读权限访问所有对象。这个配置并不会使设备发送任何 Trap 消息。

```
Device(config)# snmp-server community public
```

下面这个示例展示了如何通过配置，允许任意 SNMP 管理器使用团体字符串 *public*，以只读权限访问所有对象。设备还会使用 SNMPv1 向主机 192.180.1.111 和 192.180.1.33 发送 VTP Trap 消息，使用 SNMPv2C 向主机 192.180.1.27 发送 VTP Trap 消息。团体字符串 *public* 也随 Trap 消息发送。

```
Device(config)# snmp-server community public
Device(config)# snmp-server enable traps vtp
```

```
Device(config)# snmp-server host 192.180.1.27 version 2c public
Device(config)# snmp-server host 192.180.1.111 version 1 public
Device(config)# snmp-server host 192.180.1.33 public
```

下面这个示例展示了如何通过配置，允许访问列表 4 中指定的成员使用团体字符串 *comaccess*，以只读的权限访问所有对象。其他 SNMP 管理器不能访问任何对象。设备会使用团体字符串 *public*，以 SNMPv2C 向主机 *icntnetworks.com* 发送 SNMP 认证失败 Trap 消息。

```
Device(config)# snmp-server community comaccess ro 4
Device(config)# snmp-server enable traps snmp authentication
Device(config)# snmp-server host icntnetworks.com version 2c public
```

下面这个示例展示了如何通过配置，使设备向主机 *icntnetworks.com* 发送实体（Entity）MIB Trap 消息。团体字符串是 *restricted*。第一条命令会使设备发送实体（Entity）MIB Trap 消息，以及之前启用的 Trap 消息。第二条命令指定了这些 Trap 消息的目的地，如果没有指定的话，会使用以前为主机 *icntnetworks.com* 设置的 **snmp-server host** 命令。

```
Device(config)# snmp-server enable traps entity
Device(config)# snmp-server host icntnetworks.com restricted entity
```

下面这个示例展示了如何通过配置，让设备使用团体字符串 *public* 向主机 *myhost.icntnetworks.com* 发送所有 Trap 消息：

```
Device(config)# snmp-server enable traps
Device(config)# snmp-server host myhost.icntnetworks.com public
```

下面这个示例展示了如何通过配置，把用户与远端主机相关联，并在用户进入全局配置模式时，让设备发送 **auth**（有认证无加密）认证级别的 Inform 消息：

```
Device(config)# snmp-server engineID remote 192.180.1.27
00000063000100a1c0b4011b
Device(config)# snmp-server groupauth group v3 auth
Device(config)# snmp-server user authuser authgroup remote
192.180.1.27 v3 auth md5 mypassword
Device(config)# snmp-server user authuser authgroup v3 auth md5
mypassword
Device(config)# snmp-server host 192.180.1.27 informs version 3 auth
authuser config
Device(config)# snmp-server enable traps
Device(config)# snmp-server inform retries 0
```

其他参考资料

相关文档

相关主题	文档名称
SNMP 命令	*Network Management Command Reference, Inspur INOS*

错误消息解码器

描述	链接
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

标准和 RFC

标准/RFC	标题
无	-

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	http://www.icntnetworks.com

简单网络管理协议的特性历史与信息

版本	变更
Inspur INOS 11.3.1	引入该特性

配置服务等级协定

本章描述了如何在交换机上使用 Inspur INOS IP 服务等级协定（SLA）。除非另行说明，否则术语*交换机*在这里表示单台交换机或交换机堆栈。

查询特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

SLA 的限制条件

这部分列出了 SLA 的限制条件。

IP SLA 网络性能测量的限制条件如下所示：

- 设备不支持使用网守注册延迟探针评估的 VoIP 服务等级；
- 只有 Inspur INOS 设备可以作为目的 IP SLA 响应方的源；
- 用户不能在非 Inspur 设备上配置 IP SLA 响应方，Inspur INOS IP SLA 只能向那些设备的原生服务发送探针数据包。

SLA 的相关信息

Inspur INOS IP 服务等级协定（SLA）

Inspur INOS IP SLA 会向网络中发送数据，以此来评估多个网络站点或多条网络路径的性能。它会模拟网络数据和 IP 服务，并实时收集网络性能信息。Inspur INOS IP SLA 可以在 Inspur INOS 设备之间生成并分析流量，也可以从 Inspur INOS 设备向远端 IP 设备（比如网络应用服务器）发送并分析流量。用户通过使用各种 Inspur INOS IP SLA 探针提供的评估结果，可以进行排错、问题分析，以及设计网络拓扑。

根据具体的 Inspur INOS IP SLA 探针，Inspur 设备中的各种网络性能状态统计信息都可以实现监控，并且这些状态统计信息会同时保存在命令行界面（CLI）和简单网络管理协议（SNMP）MIB 中。IP SLA 数据包拥有可配置的 IP 地址和应用层选项，比如源和目的 IP 地址、用户数据报协议（UDP）/TCP 端口号、服务类型（ToS）字节（其中包括差分服务代码点[DSCP]和 IP 前缀比特）、虚拟专用网（VPN）路由/转发实例（VRF），以及 URL 网页地址。

由于 Inspur IP SLA 与二层传输无关，因此用户可以在相互分离的网络上配置端到端探针，以便更好地反映出度量结果，也就是终端用户最有可能经历的环境。IP SLA 会收集并分析下列性能度量值：

- 延迟（往返和单向）
- 抖动（有方向性）
- 丢包（有方向性）
- 数据包序列（数据包排序）
- 路径（逐跳）
- 连通性（有方向性）
- 服务器或网站下载时间

由于 Inspur INOS IP SLA 可以由 SNMP 进行访问，因此性能监控应用也能使用它，比如 Inspur Prime 互联网络性能监控器（IPM）和其他第三方 Inspur 合作伙伴的性能管理产品。

使用 IP SLA 可以获得以下好处：

- 服务等级协定监控、评估和验证；
- 网络性能监控：
  - 评估网络中的抖动、延迟或丢包
  - 可持续性、可靠性和可预测性评估
- IP 服务网络健康评估特性能够验证现有的 QoS 是否能够满足新的 IP 服务；
- 边界到边界网络可用性监控特性能够提供主动的网络资源验证和连通性测试（举例来说，从远端站点查看用来存储业务重要数据的 NFS 服务器的网络可用性）；
- 通过提供持续且可靠的评估结果，可以立即指出问题并节省排错时间，以此实现网络探针排错；
- 多协议标签交换（MPLS）性能监控和网络验证（如果设备支持 MPLS 的话）。

使用 Inspur INOS IP SLA 评估网络性能

用户可以使用 IP SLA 来监控网络中各个区域之间的性能——核心层、分布层和边界——而无需部署物理探针。它会通过生成的流量来测量两台网络通信设备之间的网络性能。

下图展示了当源色湖北向目的设备发送了它所生成的数据包后，IP SLA 是如何展开工作的。在目的设备收到数据包后，根据 IP SLA 探针的类型，它会向源设备反悔带有时间戳的信息，以便计算性能度量值。IP SLA 探针可以使用某种具体协议（比如 UDP）来评估网络中源设备与目的设备之间的网络性能。

图 74：Inspur INOS IP SLA 探针

Any IP device	任意 IP 设备
IP SLA measurement and IP SLA responder to IP SLA responder（共 2 处）	IP SLA 评估以及 IP SLA 响应方到 IP SLA 响应方
IP SLA responder	IP SLA 响应方
IP network	IP 网络
IP SLA source	IP SLA 源
Performance management application	性能管理应用

IP SLA 响应方和 IP SLA 控制协议

IP SLA 响应方是内嵌在目的 Inspur 设备中的一个组成部分，它使系统能够参与并响应 IP SLA 请求数据包。响应方能够提供精确的评估，无需部署专用的探针。响应方使用 Inspur INOS IP SLA 控制协议，使其能够知道应该监听哪个端口以及使用哪个端口进行响应。

注释： IP SLA 响应方可以是 Inspur INOS 二层（可配置响应方的）设备。响应方无需支持完整的 IP SLA 功能。

下图展示了 IP 网络中能够使用 Inspur INOS IP SLA 响应方的位置。响应方会在指定端口上监听由 IP SLA 探针发来的控制协议消息。根据收到的控制消息，它能够在指定时间段内启用指定的 UDP 或 TCP 端口。在这段时间内，响应方会接受请求并对其进行响应。在它对 IP SLA 数据包做出响应后，或者当指定时间超时后，它会禁用相应的端口。为了增加安全性，设备可以为控制消息实施 MD5 认证

图 75：Inspur INOS IP SLA 探针

Any IP device	任意 IP 设备
IP SLA measurement and IP SLA responder to IP SLA responder（共 2 处）	IP SLA 评估以及 IP SLA 响应方到 IP SLA 响应方
IP SLA responder	IP SLA 响应方
IP network	IP 网络
IP SLA source	IP SLA 源
Performance management application	性能管理应用

用户无需在目的设备上为所有 IP SLA 探针启用响应方。举例来说，对于目的路由器上已经提供的服务就无需启用响应方（比如 Telnet 或 HTTP）。

IP SLA 响应时间的计算

交换机、控制器和路由器可以通过使用其他高优先级进程，花费几十毫秒来处理入站数据包。这个延迟会影响响应时间，因为测试数据包的响应可能会在等待处理时进入队列。在这种情况中，响应时间可能无法精确反映出真实的网络延迟。IP SLA 会把源设备和目标设备上（如果使用了响应方）的这些处理延迟最小化，以此来确定真实的往返时间。IP SLA 测试数据包

会使用时间戳来把处理延迟最小化。

当启用了 IP SLA 响应方后，它使目标设备在数据包到达接口时，在中断级（Interrupt Level）设置时间戳，并在测试数据包离开时也设置时间戳，以此消除处理时间。这个时间戳精确到亚毫秒级。

下图展示了响应方的工作。其中共有四个时间戳用来计算往返时间。在目标路由器上，当启用了响应方功能后，用时间戳 3（TS3）减去时间戳 2（TS2）就能够得到花费在处理测试数据包上的时间，也就是图中 Δ 表示的内容。最后 IP SLA 会从整体往返时间中减去这个 Δ 值。源路由器上的 IP SLA 也会执行相同的行为，也就是会在中断级设置入站时间戳 4（TS4），来达到更高的精确度。

图 76：Inspur INOS IP SLA 响应方时间戳

Source router	源路由器
Target router	目标路由器
Responder	响应方
RTT (Round-trip time) = T4 (Time stamp 4) – T1 (Time stamp 1) - Δ	RTT（往返时间）= T4（时间戳 4）– T1（时间戳 1）– Δ

在目标设备上设置两个时间戳带来的另一个好处是：能够追踪单向延迟、抖动和单向丢包。由于很多网络行为都不是同步发生的，因此很有必要记录这些统计数据。然而，在进行单向延迟评估时，用户必须同时在源路由器和目标路由器上配置网络时间协议（NTP），这样源和目标才能与相同的时钟源进行同步。单向抖动的评估不需要同步时钟。

IP SLA 探针计划

当用户在配置 IP SLA 探针时，必须对探针开始捕获状态统计信息的时间，以及收集错误信息的时间有所计划。用户可以设置让探针立即开始工作，也可以指定具体的月、日、小时。用户可以使用 *pending（待定）*选项让探针稍后开始工作。待定选项是探针的一种内部状态，可以通过 SNMP 进行查看。当一个探针的响应（门限值）行为有待触发时，探针的状态也是待定。用户可以一次性规划一个 IP SLA 探针的工作时间，也可以同时规划一组探针的工作时间。

用户可以在 Inspur INOS CLI 或 INSPUR RTTMON-MIB 中使用一条命令来规划多个 IP SLA 探针的工作时间。通过规划让探针平均地分多次运行，可以让用户控制 IP SLA 监控流量的总量。这种分布 IP SLA 探针的工作方式有助于最小化 CPU 利用率，从而提高网络可扩展性。

更多有关 IP SLA 多探针计划功能的详细信息，用户可以参考 *Inspur INOS IP SLA 配置指南（Inspur INOS IP SLAs Configuration Guide）*中“IP SLA——多探针计划”一章。

IP SLA 探针门限值监控

为了成功地进行服务等级协定监控，必须有某种机制能够及时向用户通知网络中可能发生的违规行为。IP SLA 可以发送 SNM Trap 消息，下列这些事件可以作为触发机制：

- 连接断开
- 超时
- 往返时间门限值
- 平均抖动门限值
- 单向丢包
- 单向抖动
- 单向平均意见得分（MOS）
- 单向延迟

一个 IP SLA 门限值检测到的违规行为同时也会触发另一个 IP SLA 探针进行深入分析。比如提高测试频率，或者开始使用 Internet 控制消息协议（ICMP）路径应答或 ICMP 路径抖动探针来进行排错。

ICMP Echo

ICMP Echo（应答）探针能够评估 Inspur 设备与其他 IP 设备之间端到端的响应时间。响应时间是由测量源设备向目的设备发出 ICMP Echo 请求消息，再到源设备收到 ICMP Echo Reply 之间所花费的时间计算出来的。很多客户都会使用 IP SLA 中基于 ICMP 的探针，进行内部 Ping 测试，或者基于 Ping 的专用探针来评估响应时间。IP SLA ICMP Echo 探针符合 ICMP Ping 测试的定义，这两种方式都可以得到相同的响应时间。

UDP 抖动

抖动与数据包之间延迟的变化是同义术语。当源设备向目的设备以 10 毫秒为间隔连续发送多个数据包时，目的设备应该每隔 10 毫秒收到一个数据包（如果网络行为完全正常的话）。但如果网络中存在延迟（比如队列延迟、通过替代路径到达目的地等），数据包的到达时间间隔可能会小于或大于 10 毫秒。正抖动值表示数据包的到达时间间隔大于 10 毫秒。负抖动值表示数据包的到达时间间隔小于 10 毫秒。如果数据包的到达时间间隔是 12 毫秒，正抖动就是 2 毫秒；如果数据包的到达时间间隔是 8 毫秒，负抖动就是 2 毫秒。对于延迟敏感的网络，正抖动是不可容忍的，抖动值为 0 是理想情况。

除了监控抖动外，IP SLA UDP 抖动探针还能当作多目的数据收集探针。由 IP SLA 生成的数据

包中会携带序列号信息，以及从源到目的的时间戳，其中还包含数据包的发送和接收数据。根据这些数据，UDP 抖动探针可以评估下列度量值：

- 有方向性的抖动（源到目的，以及目的到源）
- 有方向性的丢包
- 有方向性的延迟（单向延迟）
- 往返延迟（平均往返时间）

由于发送和接收数据的路径可能有多条(不对称路径),用户可以使用有方向性的数据测试，更有效地识别网络中拥塞的位置，或者网络中发生的其他问题。

UDP 抖动探针能够产生合成（模拟）UDP 流量，并发送大量 UDP 数据包，并且每个数据包都有指定的大小，发送指定的毫秒数，从源路由器到目标路由器以固定的频率发送。默认情况下，UDP 抖动探针会发送 10 个数据包-数据帧，每个负载大小为 10 字节，每 10 毫秒生成一个，每 60 秒重复进行测试。用户可以对这些参数一一进行配置，精确模拟网络中的 IP 服务。

为了提供精确的单向延迟评估结果，源设备和目标设备之间需要进行时间同步（比如 NTP 提供的服务）。评估单向抖动和丢包不必需进行时间同步。如果源设备和目标设备之间的时间没有同步，UDP 抖动探针在反回单向延迟和丢包的评估数据时，单向延迟的评估结果为 0。

如何配置 IP SLA 探针

这部分中不包含所有可用的探针配置信息，*Inspur INOS IP SLA 配置指南（Inspur INOS IP SLAs Configuration Guide）*中包含更多详细配置信息。这部分中会包含多个探针配置示例，其中包括配置响应方、配置 UDP 抖动探针（需要配置响应方），以及配置 ICMP Echo 探针（不需要配置响应方）。有关配置其他探针的详细信息，用户可以参考 *Inspur INOS IP SLA 配置指南（Inspur INOS IP SLAs Configuration Guide）*。

默认配置

设备中没有配置 IP SLA 探针。

配置指导

有关 IP SLA 命令的信息,用户可以查看 *Inspur INOS IP SLA 命令参考,版本 12.4T(Inspur IP SLAs*

Command Reference, Release 12.4T）命令参考手册。

有关描述和配置步骤的细节信息，用户可以查看 *Inspur INOS IP SLA 配置指南，版本 12.4TL*（*Inspur INOS IP SLAs Configuration Guide, Release 12.4TL*）。

并不是参考指南中的所有 IP SLA 命令或探针设备都能够支持。设备支持的 IP 服务等级分析功能包括使用 UDP 抖动、UDP Echo、HTTP、TCP 连接、ICMP Echo、ICMP 路径 Echo、ICMP 路径抖动、TFP、DNS 和 DHCP，以及多种探针计划和主动门限值监控。设备不支持使用网守注册延迟探针评估的 VoIP 服务等级。

在配置任意 IP SLA 应用之前，用户可以使用特权 EXEC 命令 **show ip sla application**，来确认软件版本所支持的探针类型。这条命令的输出示例如下所示：

```
Device# show ip slaapplication
IP Service Level Agreements
Version: Round Trip Time MIB 2.2.0, Infrastructure Engine-III
Supported OperationTypes:
icmpEcho, path-echo, path-jitter, udpEcho,tcpConnect, http
dns, udpJitter, dhcp,ftp, udpApp, wspApp
Supported Features:
IPSLAs Event Publisher
IP SLAs low memory water mark: 33299323
Estimated system maxnumber of entries: 24389
Estimated number ofconfigurable operations:24389
Number of Entriesconfigured : 0
Number of active Entries: 0
Number of pending Entries: 0
Number of inactiveEntries : 0
Time of last change inwhole IP SLAs: *13:04:37.668 UTC Wed Dec 19 2012
```

配置 IP SLA 响应方

只有运行 Inspur INOS 软件的设备上能够使用 IP SLA 响应方功能，其中包括不支持完整 IP SLA 功能的一些二层设备。

用户可以按照以下步骤，在目标设备（探针目标）上配置 IP SLA 响应方。

总步骤

1. enable

2. configure terminal

3. ip sla responder {**tcp-connect** | **udp-echo**} **ipaddress** *ip-address* **port** *port-number*

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip sla responder** {**tcp-connect** | **udp-echo**} **ipaddress** *ip-address* **port** *port-number* **示例：** `Device(config)# ip sla responder udp-echo 172.29.139.134 5000`	把设备配置为 IP SLA 响应方。 关键字的解释如下所示： • **tcp-connect——**为 TCP 连接探针启用响应方 • **udp-echo**——为用户数据报协议（UDP）Echo 或抖动探针启用响应方 • **ipaddress** *ip-address*——输入目的 IP 地址 • **port** *port-number*——输入目的端口号 **注释：** IP 地址和端口号必须与源设备上为 IP SLA 探针配置的 IP 地址和端口号相同
步骤 4	**end** **示例：**	返回特权 EXEC 模式

	`Device(config)# end`	
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

实施 IP SLA 网络性能评估

用户可以按照以下步骤，在设备上实施 IP SLA 网络性能评估。

在开始前

用户可以使用特权 EXEC 命令 **show ip sla application**，来确认软件版本所支持的探针类型。

总步骤

1. enable

2. configure terminal

3. ip sla *operation-number*

4. udp-jitter {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interval-packet-interval*]

5. frequency *seconds*

6. threshold *milliseconds*

7. exit

8. ipsla schedule *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*] [**ageout** *seconds*] [**recurring**]

9. end

10. show running-config

11. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip sla** *operation-number* **示例：** `Device(config)# ip sla 10`	创建 IP SLA 探针，并进入 IP SLA 配置模式
步骤 4	**udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*] **示例：** `Device(config-ip-sla)# udp-jitter 172.29.139.134 5000`	配置 IP SLA 探针，用户可以任意选择探针类型（示例中使用了 UDP 抖动探针），并进入这个探针的配置模式（示例中进入了 UDP 抖动配置模式）。 • *destination-ip-address* | *destination-hostname*——指定目的 IP 地址或主机名 • *destination-port*——指定目的端口，取值范围是 1 至 65535 • （可选）**source-ip** {*ip-address* | *hostname*}——指定源 IP 地址或主机名。如果用户没有指定源 IP 地址或主机名，IP SLA 会选择距离目的地最近的 IP 地址 • （可选）**source-port** *port-number*——指定源端口号，取值范围是 1 至 65535。当用户没有指定源端口号时，IP SLA 会选择一个可用端口

		• （可选）**control**——启用或禁用设备向 IP SLA 响应方发送 IP SLA 控制消息的功能。默认情况下，设备会向目的设备发送 IP SLA 控制消息，以便与 IP SLA 响应方建立连接 • （可选）**num-packets** *number-of-packets*——输入需要生成的数据包数量。取值范围是 1 至 6000；默认值为 10 • （可选）**interval** *interpacket-interval*——以毫秒为单位输入发送数据包的时间间隔。取值范围是 1 至 6000；默认值为 20 毫秒
步骤 5	**frequency** *seconds* **示例：** `Device(config-ip-sla-jitter)#` **`frequency 45`**	（可选）为 SLA 探针配置选项信息。示例中指定了 IP SLA 探针重复执行的速率。取值范围是 1 至 604800；默认值为 60 秒
步骤 6	**threshold** *milliseconds* **示例：** `Device(config-ip-sla-jitter)#` **`threshold 200`**	（可选）配置门限值条件。示例中把指定 IP SLA 探针的门限值设置为 200。取值范围是 1 至 60000 毫秒
步骤 7	**exit** **示例：** `Device(config-ip-sla-jitter)#` **`exit`**	退出 SLA 探针配置模式（也就是示例中的 UDP 抖动配置模式），并返回全局配置模式
步骤 8	**ipsla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm*	为某个 IP SLA 探针配置计划参数。 • *operation-number*——输入 RTR 条

	[*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*] [**ageout** *seconds*] [**recurring**] **示例：** `Device(config)# ip sla schedule 10 start-time now life forever`	目数量 • （可选）**life**——设置探针永远运行（**forever**）或以秒 *seconds* 为单位指定时长。取值范围是 0 至 2147483647；默认值为 3600 秒（1 小时） • （可选）**start-time**——输入这个探针开始收集信息的时间： 要想在指定时间开始，输入小时、分钟、秒（以 24 小时格式），以及月份日期。如果没有输入月份，默认为当前月。 输入 **pending** 表示不收集信息，直到设置了开始时间。 输入 **now** 表示马上开始运行探针。 输入 **after** *hh:mm:ss* 表示在指定时间过后开始执行探针。 • （可选）**ageout** *seconds*——以秒为单位输入当探针并没有收集到信息使，把探针保存在内存中的时间。取值范围是 0 至 2073600 秒，默认值为 0 秒（永不超时） • （可选）**recurring**——设置让探针每天自动运行
步骤 9	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 10	**show running-config** **示例：**	检查用户输入的信息

	Device# **show running-config**	
步骤 11	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

UDP 抖动配置

以下示例展示了如何配置 UDP 抖动 IP SLA 探针：

```
Device(config)# ip sla 10
Device(config-ip-sla)# udp-jitter 172.29.139.134 5000
Device(config-ip-sla-jitter)# frequency 30
Device(config-ip-sla-jitter)# exit
Device(config)# ip sla schedule 5 start-time now life forever
Device(config)# end
Device# show ip sla configuration 10
IP SLAs, InfrastructureEngine-II.
Entry number: 10
Owner:
Tag:
Type of operation toperform: udp-jitter
Target address/Sourceaddress: 1.1.1.1/0.0.0.0
Target port/Source port: 2/0
Request size (ARR dataportion): 32
Operation timeout(milliseconds): 5000
Packet Interval (milliseconds)/Number of packets: 20/10
Type Of Service parameters: 0x0
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
Operation frequency(seconds): 30
Next Scheduled StartTime: Pending trigger
Group Scheduled : FALSE
```

```
Randomly Scheduled: FALSE
Life (seconds): 3600
Entry Ageout (seconds):never
Recurring (StartingEveryday): FALSE
Status of entry (SNMPRowStatus): notInService
Threshold (milliseconds): 5000
Distribution Statistics:
Number of statistic hours kept:2
Number of statisticdistribution buckets kept: 1
Statistic distributioninterval (milliseconds): 20
Enhanced History:
```

使用 UDP 抖动探针分析 IP 服务等级

用户可以按照以下步骤，在源设备上配置 UDP 抖动探针。

在开始前

用户必须在目标设备（探针目标）上启用 IP SLA 响应方功能，以便在源设备上配置 UDP 抖动探针。

总步骤

1. enable

2. configure terminal

3. ip sla *operation-number*

4. udp-jitter {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*]

5. frequency *seconds*

6. exit

7. sla schedule *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*] [**ageout** *seconds*] [**recurring**]

8. end

9. show running-config

10. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip sla** *operation-number* **示例：** `Device(config)# ip sla 10`	创建 IP SLA 探针，并进入 IP SLA 配置模式
步骤 4	**udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*] **示例：** `Device(config-ip-sla)# udp-jitter 172.29.139.134 5000`	配置 IP SLA 探针，用户可以任意选择探针类型（示例中使用了 UDP 抖动探针），并进入这个探针的配置模式（示例中进入了 UDP 抖动配置模式）。 • *destination-ip-address* | *destination-hostname*——指定目的 IP 地址或主机名 • *destination-port*——指定目的端口，取值范围是 1 至 65535 • （可选）**source-ip** {*ip-address* | *hostname*}——指定源 IP 地址或主机名。如果用户没有指定源 IP 地址或主机名，IP SLA 会选择距离目的地最近的 IP 地址 • （可选）**source-port** *port-number*——指定源端口号，取值范围是 1 至 65535。当用户没有指定源端口号时，IP SLA 会选择一个可用端口

		• （可选）**control**——启用或禁用设备向 IP SLA 响应方发送 IP SLA 控制消息的功能。默认情况下，设备会向目的设备发送 IP SLA 控制消息，以便与 IP SLA 响应方建立连接 • （可选）**num-packets** *number-of-packets*——输入需要生成的数据包数量。取值范围是 1 至 6000；默认值为 10 • （可选）**interval** *interpacket-interval*——以毫秒为单位输入发送数据包的时间间隔。取值范围是 1 至 6000；默认值为 20 毫秒
步骤 5	**frequency** *seconds* **示例：** `Device(config-ip-sla-jitter)# frequency 45`	（可选）为 SLA 探针配置选项信息。示例中指定了 IP SLA 探针重复执行的速率。取值范围是 1 至 604800；默认值为 60 秒
步骤 6	**exit** **示例：** `Device(config-ip-sla-jitter)# exit`	退出 UDP 抖动配置模式，并返回全局配置模式
步骤 7	**ipsla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*] [**ageout** *seconds*] [**recurring**] **示例：**	为某个 IP SLA 探针配置计划参数。 • *operation-number*——输入 RTR 条目数量 • （可选）**life**——设置探针永远运行（**forever**）或以秒 *seconds* 为单位指定时长。取值范围是 0 至 2147483647；默认值为 3600 秒（1

	Device(config)# **ip sla schedule 10 start-time now life forever**	小时） • （可选）**start-time**——输入这个探针开始收集信息的时间： 要想在指定时间开始，输入小时、分钟、秒（以 24 小时格式），以及月份日期。如果没有输入月份，默认为当前月。 输入 **pending** 表示不收集信息，直到设置了开始时间。 输入 **now** 表示马上开始运行探针。 输入 **after** *hh:mm:ss* 表示在指定时间过后开始执行探针。 • （可选）**ageout** *seconds*——以秒为单位输入当探针并没有收集到信息使，把探针保存在内存中的时间。取值范围是 0 至 2073600 秒，默认值为 0 秒（永不超时） • （可选）**recurring**——设置让探针每天自动运行
步骤 8	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 9	**show running-config** **示例：** Device# **show running-config**	检查用户输入的信息
步骤 10	**copy running-config startup-config** **示例：** Device# **copy running-config**	（可选）把输入的命令保存到配置文件中

	startup-config	

配置 UDP 抖动 IP SLA 探针

以下示例展示了如何配置 UDP 抖动 IP SLA 探针：

```
Device(config)# ip sla 10
Device(config-ip-sla)# udp-jitter 172.29.139.134 5000
Device(config-ip-sla-jitter)# frequency 30
Device(config-ip-sla-jitter)# exit
Device(config)# ip sla schedule 5 start-time now life forever
Device(config)# end
Device# show ip sla configuration 10
IP SLAs, InfrastructureEngine-II.
Entry number: 10
Owner:
Tag:
Type of operation toperform: udp-jitter
Target address/Sourceaddress: 1.1.1.1/0.0.0.0
Target port/Source port: 2/0
Request size (ARR dataportion): 32
Operation timeout(milliseconds): 5000
Packet Interval (milliseconds)/Number of packets: 20/10
Type Of Service parameters: 0x0
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
Operation frequency(seconds): 30
Next Scheduled StartTime: Pending trigger
Group Scheduled : FALSE
Randomly Scheduled: FALSE
Life (seconds): 3600
Entry Ageout (seconds):never
Recurring (StartingEveryday): FALSE
Status of entry (SNMPRowStatus): notInService
```

```
Threshold (milliseconds): 5000
Distribution Statistics:
Number of statistic hours kept:2
Number of statisticdistribution buckets kept: 1
Statistic distributioninterval (milliseconds): 20
Enhanced History:
```

使用 ICMP Echo 探针分析 IP 服务等级

用户可以按照以下步骤，在源设备上配置 ICMP Echo 探针：

在开始前

这个探针不需要启用 IP SLA 响应方。

总步骤

1. enable

2. configure terminal

3. ip sla *operation-number*

4. icmp-echo {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*} | **source-interface** *interface-id*]

5. frequency *seconds*

6. exit

7. sla schedule *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*] [**ageout** *seconds*] [**recurring**]

8. end

9. show running-config

10. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device# configure terminal`	
步骤 3	**ip sla** *operation-number* **示例：** `Device(config)# ip sla 10`	创建 IP SLA 探针，并进入 IP SLA 配置模式
步骤 4	**icmp-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*} | **source-interface** *interface-id*] **示例：** `Device(config-ip-sla)# icmp-echo 172.29.139.134`	配置 ICMP Echo 探针，并进入 ICMP Echo 配置模式。 • *destination-ip-address* | *destination-hostname*——指定目的 IP 地址或主机名 • （可选）**source-ip** {*ip-address* | *hostname*}——指定源 IP 地址或主机名。如果用户没有指定源 IP 地址或主机名，IP SLA 会选择距离目的地最近的 IP 地址 • （可选）**source-interface** *interface-id*——为探针指定源接口
步骤 5	**frequency** *seconds* **示例：** `Device(config-ip-sla-echo)# frequency 45`	（可选）为 SLA 探针配置选项信息。示例中指定了 IP SLA 探针重复执行的速率。取值范围是 1 至 604800；默认值为 60 秒
步骤 6	**exit** **示例：** `Device(config-ip-sla-echo)# exit`	退出 UDP 抖动配置模式，并返回全局配置模式
步骤 7	**ipsla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm*	为某个 IP SLA 探针配置计划参数。 • *operation-number*——输入 RTR 条

	[*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*] [**ageout** *seconds*] [**recurring**] **示例：** `Device(config)# ip sla schedule 10 start-time now life forever`	目数量 • （可选）**life**——设置探针永远运行（**forever**）或以秒 *seconds* 为单位指定时长。取值范围是 0 至 2147483647；默认值为 3600 秒（1 小时） • （可选）**start-time**——输入这个探针开始收集信息的时间： 要想在指定时间开始，输入小时、分钟、秒（以 24 小时格式），以及月份日期。如果没有输入月份，默认为当前月。 输入 **pending** 表示不收集信息，直到设置了开始时间。 输入 **now** 表示马上开始运行探针。 输入 **after** *hh:mm:ss* 表示在指定时间过后开始执行探针。 • （可选）**ageout** *seconds*——以秒为单位输入当探针并没有收集到信息使，把探针保存在内存中的时间。取值范围是 0 至 2073600 秒，默认值为 0 秒（永不超时） • （可选）**recurring**——设置让探针每天自动运行
步骤 8	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 9	**show running-config** **示例：**	检查用户输入的信息

	Device# **show running-config**	
步骤 10	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

配置 ICMP Echo IP SLA 探针

以下示例展示了如何配置 ICMP Echo IP SLA 探针：

```
Device(config)# ip sla 12
Device(config-ip-sla)# icmp-echo 172.29.139.134
Device(config-ip-sla-jitter)# frequency 30
Device(config-ip-sla-jitter)# exit
Device(config)# ip sla schedule 5 start-time now life forever
Device(config)# end
Device# show ip sla configuration 12
Entry number: 12
Owner:
Tag:
Type of operation toperform: echo
Target address: 2.2.2.2
Source address: 0.0.0.0
Request size (ARR dataportion): 28
Operation timeout(milliseconds): 5000
Type Of Service parameters: 0x0
Verify data: No
Vrf Name:
Schedule:
Operation frequency(seconds): 60
Next Scheduled StartTime: Pending trigger
Group Scheduled : FALSE
Randomly Scheduled: FALSE
Life (seconds): 3600
Entry Ageout (seconds):never
```

```
Recurring (StartingEveryday): FALSE
Status of entry (SNMPRowStatus): notInService
Threshold (milliseconds): 5000
Distribution Statistics:
Number of statistic hours kept:2
Number of statisticdistribution buckets kept: 1
Statistic distributioninterval (milliseconds): 20
History Statistics:
Number of history Liveskept: 0
Number of history Buckets kept: 15
History Filter Type: None
Enhanced History:
```

监控 IP SLA 探针

下面这个表格中描述了用来查看 IP SLA 探针配置和结果的命令。

表 76：监控 IP SLA 探针

命令	目的
show ip sla application	显示有关 INOS IP SLA 的全局信息
show ip sla authentication	显示 IP SLA 认证信息
show ip sla configuration [*entry-number*]	显示配置值，其中包括所有 IP SLA 探针或指定探针的所有默认值
show ip sla enhanced-history {**collection-statistics** | **distribution-statistics**} [*entry-number*]	以收集的历史桶或分布统计信息形式，为所有 IP SLA 探针或指定探针显示高级历史统计状态信息
show ip sla ethernet-monitor configuration [*entry-number*]	显示 IP SLA 自动以太网配置
show ip sla group schedule [*schedule-entry-number*]	显示 IP SLA 组计划配置及其详细信息
show ip sla history [*entry-number* | **full** | **tabular**]	显示为所有 IP SLA 探针收集的历史信息

show ip sla mpls-lsp-monitor {**collection-statistics** | **configuration** | **ldp operational-state** | **scan-queue** | **summary** [*entry-number*] | **neighbors**}	显示 MPLS 标签交换路径（LSP）健康监控器探针
show ip sla reaction-configuration [*entry-number*]	显示为所有 IP SLA 探针或指定探针设置的正门限值监控设置
show ip sla reaction-trigger [*entry-number*]	显示为所有 IP SLA 探针或指定探针设置的响应触发器信息
show ip sla responder	显示有关 IP SLA 响应方的信息
show ip sla statistics [*entry-number* | **aggregated** | **details**]	显示当前或汇集的探针状态和统计状态信息

监控 IP SLA 探针示例

以下示例展示了所有 IP SLA 应用的相关信息：

```
Device# show ip slaapplication
IP Service Level Agreements
Version: Round Trip Time MIB 2.2.0, Infrastructure Engine-III
Supported OperationTypes:
icmpEcho, path-echo, path-jitter, udpEcho,tcpConnect, http
dns, udpJitter, dhcp,ftp, udpApp, wspApp
Supported Features:
IPSLAs Event Publisher
IP SLAs low memory water mark: 33299323
Estimated system maxnumber of entries: 24389
Estimated number ofconfigurable operations:24389
Number of Entriesconfigured  : 0
Number  of  active  Entries: 0
Number  of  pending  Entries: 0
Number of inactiveEntries  : 0
Time of last change inwhole IP SLAs: *13:04:37.668 UTC Wed Dec 19 2012
```

以下示例展示了所有 IP SLA 分布统计信息：

```
Device# show ip slaenhanced-history distribution-statistics
Point by point EnhancedHistory
Entry  = Entry Number
Int  = Aggregation Interval
BucI  = Bucket Index
StartT  = Aggregation Start Time
Pth  = Path index
Hop  = Hop in path index
Comps  = Operations completed
OvrTh  = Operations completedover thresholds
SumCmp  = Sum of RTT (milliseconds)
SumCmp2L = Sum of RTTsquared low 32 bits (milliseconds)
SumCmp2H = Sum of RTTsquared high 32 bits(milliseconds)
TMax  = RTT maximum (milliseconds)
TMin  = RTT minimum (milliseconds)
Entry Int BucI StartT  Pth Hop Comps OvrThSumCmp  SumCmp2L  SumCmp2H
T
Max  TMin
```

其他参考资料

相关文档

相关主题	文档名称
Inspur Medianet Metadata Guide	*http://wwwicntnetworks.com*
Inspur Media Services Proxy Configuration Guide	*http://www.icntnetworks.com*
Inspur Mediatrace and Inspur Performance Monitor Configuration Guide	*http://www.icntnetworks.com*

错误消息解码器

描述	链接

为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

标准和 RFC

标准/RFC	标题
无	-

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	http://www.icntnetworks.com

配置本地策略

查询特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本

的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置本地策略的限制条件

- 设备上支持的 policy-map 属性包括 QoS、VLAN、会话超时和 ACL；
- HTTP 分析描述特性会把 Apple iPhone 6s 分类为“工作站”。

有关配置本地策略的信息

本地策略可以基于 HTTP 和 DHCP 对设备进行分析描述，以此识别网络中的终端设备。用户可以配置基于设备的策略，并在网络中强制使用基于用户的策略或基于设备的策略。

本地策略能够对移动设备进行分析描述，并把相应设备划分到指定 VLAN 中。本地策略还能够分配 ACL 和 QoS，或者配置会话超时时间。

用户可以使用两种相互独立的组成部分来配置本地策略：

- 以服务模版的形式定义策略属性，以此定义客户端加入网络的方式并一个用策略匹配规则；
- 在策略中应用匹配规则。

用户可以使用下列策略匹配属性来配置本地策略：

- 设备——定义设备类型。Windows 计算机、智能手机、Apple 设备（比如 iPad 和 iPhone）；
- 用户名——定义用户的用户名；
- 用户角色——定义用户类型或用户所属的用户组，比如 student（学生）或 employee（雇员）；
- MAC——定义端点的 MAC 地址；
- MAC OUI——定义 MAC 地址的 OUI。

一旦设备匹配了基于端点设定的参数，就会有策略添加到设备上。策略会在以下会话属性方面，对加入网络的移动设备进行限制：

- VLAN
- QoS
- ACL
- 会话超时时间

用户可以配置这些策略并对端点应用指定策略。设备会使用这些属性并预先定义一些类别配置文件，来识别设备。

替换默认的分析描述文本文件

如果有新设备没有被分类，用户可以把设备的 MAC 地址提供给 Inspur 支持小组（Support Team）。Inspur 支持小组会提供一个包含有该 MAC 地址的新文件 **dc_default_profile.txt**。用户需要用 **dc_default_profile.txt** 文件替换之前的文件。用户可以按照以下步骤，替换 **dc_default_profile.txt** 文件：

1. 输入以下命令来停止设备分类器：

   device**(config)# no device classifier**

2. 输入以下命令来复制文件：

   device# **device classifier profile location** *filepath*

3. 输入以下命令来启用设备分类器：

   device**(config)# device classifier**

在 Trunk 端口上禁用会话监控器

在上行 Trunk 端口上，用户不应该创建任何会话监控功能。默认情况下，会话监控功能是启用的。用户应该禁用会话监控功能。

1. 输入以下命令进入全局配置模式：

   device# **configure terminal**

2. 输入以下命令进入接口配置模式：

   device**(config)# interface** *interface-id*

3. 输入以下命令禁用会话监控功能：

   device**(config-if)# no access-session monitor**

如何配置本地策略

配置本地策略（CLI）

用户可以按照以下步骤配置本地策略：

1. 创建 service-template（服务模版）
2. 创建 interface-template（接口模版）
3. 创建 parameter-map（参数映射）
4. 创建 policy-map（策略映射）

创建 interface-template（接口模版；CLI）

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**template** *interface-template-name* **示例：** `Device(config)# template inspur-phone-template`	进入接口模版配置模式
步骤 3	**switchport mode access** **示例：** `Device(config-template)# switchport mode access`	把接口设置为非 Trunk 未打标的单 VLAN 以太网接口。Access 端口只能够承载一个 VLAN 的流量。默认情况下，Access 端口负责承载 VLAN1 的流量
步骤 4	**switchport voice vlan** *vlan_id* **示例：** `Device(config-template)# switchport voice vlan 20`	设置通过指定 VLAN 传输所有语音流量。用户可以指定的 VLAN 编号为 1 至 4094

步骤 5	**end** **示例：** `Device(config-template)# end`	返回特权 EXEC 模式。或者用户也可以使用 **Ctrl-Z** 退出全局配置模式

创建 parameter-map（参数映射；CLI）

建议优先使用 parameter-map，而不是 class-map。

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**parameter-map type subscriber attribute-to-service** *parameter-map-name* **示例：** `Device(config)# parameter-map type subscriber attribute-to-service Aironet-Policy-para`	指定 parameter-map（参数映射）的类型和名称
步骤 3	*map-index* **map** {**device-type** | **mac-address** | **oui** | **user-role** | **username**} {**eq** | **not-eq** | **regex** *filter-name*} **示例：** `Device(config-parameter-map-filter)# 10 map device-type eq "WindowsXP-Workstation"`	指定 parameter-map（参数映射）的属性过滤条件
步骤 4	**interface-template** *interface-template-name* **示例：** `Device(config-parameter-map-filter-submode)# interface-template inspur-phone-template`	进入 interface-template（接口模版）配置模式
步骤 5	**end** **示例：**	返回特权 EXEC 模式。或者用户也可以使用 **Ctrl-Z** 退出全局配置模

	Device(config-parameter-map-filter-submode)# **end**	式

创建 class-map（类映射；CLI）

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式
步骤 2	**class-map type control subscriber** *class-map-na*me {**match-all** | **match-any** | **match-firs**t} **示例：** Device(config)# **class-map type control subscriber CLASS_AC_1 match-all**	指定 class-map（类映射）的类型和名称
步骤 3	**match** {**device-type** | **mac-address** | **oui** | **username** | **userrole**} *filter-type-name* **示例：** Device(config-class-map)# **match device-type Inspur-IP-Phone-7961**	指定 class-map（类映射）的属性过滤条件
步骤 4	**end** **示例：** Device(config-class-map)# **end**	返回特权 EXEC 模式。或者用户也可以使用 **Ctrl-Z** 退出全局配置模式

创建 policy-map（策略映射，CLI）

具体步骤

	命令或操作	目的
步	**configure terminal**	进入全局配置模式

骤 1	**示例：** `Device# configure terminal`	
步骤 2	**policy-map type control subscriber** *policy-map-name* **示例：** `Device(config)# policy-map type control subscriber Aironet-Policy`	指定 policy-map（策略映射）的类型
步骤 3	**event identity-update** {**match-all** | **match-first**} **示例：** `Device(config-policy-map)# event identity-update match-all`	指定 policy-map（策略映射）的匹配条件
步骤 4	*class_number* **class** {*class_map_name* | **always**} {**do-all** | **do-until-failure** | **do-until-success**} **示例：** `Device(config-class-control-policymap)# 1 class local_policy1_class do-until-success`	配置本地分析描述 policy-map（策略映射）的编号，并指定如何实施行为。class-map（类映射）配置模式中包含以下配置选项： • **always**——无需进行任何匹配直接执行 • **do-all**——执行所有行为 • **do-until-failure**——执行所有行为，直到匹配失败为止。这是默认设置 • **do-until-success**——执行所有行为，直到匹配成功为止
步骤 5	*action-index* **map attribute-to-service table** *parameter-map-name* **示例：**	指定需要使用的 parameter-map（参数映射）表

	Device(config-policy-map)# **10 map attribute-to-service table Aironet-Policy-para**	
步骤 6	**end** **示例：** Device(config-class-map)# **end**	返回特权 EXEC 模式。或者用户也可以使用 **Ctrl-Z** 退出全局配置模式

监控本地策略

用户可以使用下面这个表格中展示的命令来对设备上配置的本地策略进行监控。

表 77：监控本地策略的命令

命令	目的
show access-session	显示访问会话的汇总信息，其中包括每个客户端或 MAC 地址的授权状态、方式和域
show access-session cache	显示客户端最新的分类情况
show device classifier attached detail	显示基于参数（比如 Mac、DHCP 或 HTTP）的客户端最新分类情况
show access-session mac *mac-address* **details**	显示为客户端应用的策略、service-template（服务模版）和属性。 **注释：** 如果在命令 **show access-session detail** 的输出信息中没有显示会话超时详情，用户应该在客户端访问会话中启用客户端分析描述功能和会话超时功能，然后执行命令 **show access-session mac** *mac-address* **details** 来查看会话超时详情
show access-session mac *mac-address* **policy**	显示为客户端应用的策略、service-template（服务模版）和属性。 除此之外，用户还可以查看 **Resultant Policy**，其中会显示以下信息： • 当会话拥有本地配置的属性时，显示最

	终应用在会话上的属性 • 从服务器应用的属性

示例：本地策略的配置

注释： 在每条配置命令最后，用户可以输入 Ctrl-Z 来执行该命令并继续配置下一行命令。

以下示例展示了如何创建 interface-template（接口模版）：

```
Device# configure terminal
Device(config)# template inspur-phone-template
Device(config-template)# switchport mode access
Device(config-template)# switchport voice vlan 20
Device(config-template)# end
```

以下示例展示了人如何创建 parameter-map（参数映射）：

```
Device# configure terminal
Device(config)# parameter-map type subscriberattribute-to-service param-wired
Device(config-parameter-map-filter)# 10 map device-type regex Inspur-IP-Phone
Device(config-parameter-map-filter-submode)# 10 interface-template inspur-phone-template
Device(config-parameter-map)# end
```

以下示例展示了如何创建 policy-map（策略映射）：

```
Device(config)# policy-map type control subscriber apple-tsim
Device(config-policy-map)# event identity-update match-all
Device(config-policy-map)# 1 class always do-until-failure
Device(config-policy-map)# 1 map attribute-to-service table apple-tsim-param
Device(config-policy-map)# end
```

其他参考资料

相关文档

相关主题	**文档名称**
安全命令	*Security Command Reference Guide, Inspur INOS*

标准和 RFC

标准/RFC	**标题**
无	-

技术助手

描述	**链接**
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	http://www.icntnetworks.com

实施本地策略配置的特性历史

版本	**特性信息**
Inspur INOS 11.3.1	引入该特性

配置 SPAN 和 RSPAN

查询特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

实施 SPAN 和 RSPAN 的先决条件

SPAN

- 用户可以使用关键字 **filter vlan**，把 SPAN 流量限制在指定 VLAN 中。如果监控的是 Trunk 端口，只有通过这个关键字指定的 VLAN 流量才会被监控。默认情况下 Trunk 端口上的所有 VLAN 都会受到监控。

RSPAN

- 建议用户先配置 RSPAN VLAN，再配置 RSPAN 源或目的会话。

实施 SPAN 和 RSPAN 的限制条件

SPAN

实施 SPAN 有以下限制条件：

- 在每台设备上，用户可以配置 66 个会话。最多可以配置 8 个源会话，其他会话可以配置为 RSPAN 目的会话。源会话既可以是本地 SPAN 会话，也可以是 RSPAN 源会话；
- 对于 SPAN 源来说，用户可以在一个会话中监控单个端口或 VLAN 的流量，也可以监控一系列端口/VLAN 或一个端口/VLAN 范围的流量。用户不能在一个 SPAN 会话中同时监控源端口和源 VLAN 的流量；
- 目的端口不能再充当源端口；源端口也不能再充当目的端口；
- 用户可以使用相同的目的端口配置两个 SPAN 会话；
- 当用户把一个设备端口配置为 SPAN 目的端口后，这个端口就不再是普通的设备端口了；只有受监控的流量会通过这个 SPAN 目的端口；
- 输入 SPAN 配置命令并不会删除之前配置的 SPAN 参数。用户必须使用全局配置模式的命令 **no monitor session** {*session_number* | **all** | **local** | **remote**}，才能删除配置的 SPAN 参数；
- 对于本地 SPAN 来说，如果用户设置了关键字 **encapsulation replicate**，那么从 SPAN 目的端口离开的出站数据包会携带原始的封装头部——未打标、ISL 或 IEEE 802.1Q。如果没有指定这个关键字，数据包会以端口本地的形式进行发送；
- 用户可以把禁用（Disabled）状态的端口配置为源端口或目的端口，但只有当目的端口和至少一个源端口/VLAN 启用后，SPAN 功能才会生效；
- 用户不能在单个 SPAN 会话中同时监控源 VLAN 和过滤 VLAN。

在一个 SPAN 会话中监控的流量具有以下限制条件：

- 源可以是端口或 VLAN，但不能在同一个会话中同时监控源端口和源 VLAN；
- 当用户启用了出向 SPAN 会话时，Wireshark 无法捕获出向数据包；
- 用户可以在同一台设备或同一个设备堆栈中，同时运行本地 SPAN 和 RSPAN 源会话。设备或设备堆栈一共支持 66 个源和 RSPAN 目的会话；
- 用户可以在两个不同的 SPAN 或 RSPAN 源会话中，配置不同的或重叠的 SPAN 源端口和源 VLAN。交换端口和路由端口都可以被配置为 SPAN 源和目的；
- 用户在一个 SPAN 会话中可以设置多个目的端口，但在一个设备堆栈中最多有 64 个目

的端口；

- SPAN 会话不会影响设备的正常操作。但超额预订的 SPAN 目的可能会导致尾部丢弃或数据包丢失，比如通过 10 Mbit/s 端口监控 100 Mbit/s 端口；
- 当启用了 SPAN 或 RSPAN 后，每个受监控的数据包都会被发送两次，一次作为普通流量发送，一次作为监控数据包发送。监控大量端口或 VLAN 会在不知不觉中生成大量网络流量；
- 用户可以在禁用（Disabled）状态的端口上配置 SPAN 会话，但只有当目的端口和至少一个源端口/VLAN 启用后，这个 SPAN 会话才会生效；
- 设备不支持在单个会话中结合本地 SPAN 和 RSPAN：
  - RSPAN 源会话中不能有本地目的端口；
  - RSPAN 目的会话中不能有本地源端口；
  - 使用了相同 RSPAN VLAN 的 RSPAN 目的会话和 RSPAN 源会话不能运行在同一台设备或同一个设备堆栈中。

RSPAN

实施 RSPAN 有以下限制条件：

- RSPAN 不支持 BPDU 数据包监控或其他二层设备协议；
- RSPAN VLAN 只能配置在 Trunk 端口上，不能配置在 Access 端口上。为了避免在 RSPAN VLAN 中生成不必要的流量，用户要确保所有参与设备都支持 VLAN remote-vlan（远端 VLAN）特性；
- 当源 Trunk 端口上有活跃的 RSPAN VLAN 时，RSPAN VLAN 是作为源，包含在基于端口的 RSPAN 会话中的。RSPAN VLAN 也可以作为 SPAN 会话中的源。但由于设备不监控 SPAN 流量，因此它也不能把任何 RSPAN VLAN 中的出向 SPAN 数据包，识别为这台设备上 RSPAN 源会话的目的；
- 如果用户启用了 VTP 和 VTP 修剪（Pruning）特性，Trunk 中的 RSPAN 流量会被修剪掉，以防止 VLAN ID 小于 1005 的无用 RSPAN 流量穿越网络；
- 要想使用 RSPAN，交换机必须运行 LAN Base 镜像；

有关 SPAN 和 RSPAN 的信息

SPAN 和 RSPAN

用户可以通过使用 SPAN 或 RSPAN 来分析穿越端口或 VLAN 的网络流量，SPAN 或 RSPAN 会把这些网络流量发送到设备上的另一个端口，或者发送到另一台设备上的端口，这个端口上连接着网络分析设备，或者其他监控或安全设备。SPAN 会复制（镜像）源端口或源 VLAN 上收到或发出（或者双向）的流量，并把这些流量发送到目的端口进行分析。SPAN 不会影响源端口或源 VLAN 上的网络流量交换。用户必须把目的端口专用于 SPAN 特性。除了 SPAN 或 RSPAN 会话要求的流量外，目的端口不会接收或转发其他任何流量。

只有进入或离开源端口的流量，或者进入或离开源 VLAN 的流量，才能使用 SPAN 进行监控；被路由到源 VLAN 的流量不会被监控。举例来说，如果用户监控了入站流量，那些从其他 VLAN 通过路由进入源 VLAN 的流量并不会受到监控；但源 VLAN 上收到的流量，以及从源 VLAN 被路由到其他 VLAN 的流量会受到监控。

用户可以使用 SPAN 或 RSPAN 目的端口，从网络安全设备向网络中注入流量。举例来说，如果用户在目的端口上连接了 Inspur 入侵检测系统（IDS）传感器应用，那么 IDS 设备可以通过发送 TCP 重置数据包，来中断嫌疑攻击者的 TCP 会话。

本地 SPAN

本地 SPAN 支持在一台设备中部署 SPAN 会话；也就是所有源端口或源 VLAN，以及目的端口都在同一台设备或设备堆栈中。本地 SPAN 会从属于任意 VLAN 中的一个或多个源端口，或者从一个或多个 VLAN 复制流量，并将其发送到目的端口进行分析。

端口 5（源端口）的所有流量都会被镜像发送到端口 10（目的端口）。连接在端口 10 上的网络分析设备能够收到端口 5 的所有网络流量，而它却无需在物理上与端口 5 相连。

图 77：单台设备上实施本地 SPAN 配置的示例

Port 5 traffic mirrored on Port 10	端口 5 的流量被 镜像到端口 10
Network analyzer	网络分析设备

下图为设备堆栈中实施本地 SPAN 的示例，其中源和目的端口位于不同的堆栈成员上。

图 78：在设备堆栈中实施本地 SPAN 配置的示例

Switch stack	交换机堆栈
Switch 1	交换机 1
Port 4 on switch 1 in the stack mirrored on port 15 on switch 2	堆栈中交换机 1 上的端口 4 把 流量镜像到交换机 2 上的端口 15
Stackwise Plus port connections	智能堆叠 端口连接
Switch 2	交换机 2
Switch 3	交换机 3
Network analyzer	网络分析设备

远端 SPAN

RSPAN 能够支持源端口、源 VLAN 和目的端口分别位于不同的设备上（不同的设备堆栈上），用户可以在网络中的多台设备上启用远端监控。

下图中展示出设备 A 和设备 B 上配置了源端口。用户指定的 RSPAN VLAN 用来承载每个 RSPAN 会话的流量，这个 VLAN 专门用来承载所有参与设备的 RSPAN 会话流量。从源端口或源 VLAN 来的 RSPAN 流量会被复制到 RSPAN VLAN 中，并通过包含有 RSPAN VLAN 的 Trunk 端口把这些流量转发到监控这个 RSPAN VLAN 的目的会话。每个 RSPAN 源设备必须以端口或 VLAN 作为 RSPAN 源。目的总是物理端口，比如图中设备 C 上的目的端口。

图 79：RSPAN 配置的示例

RSPAN destination ports	RSPAN 目的端口
Switch C	交换机 C
RSPAN destination session	RSPAN 目的会话
Intermediate switches must support RSPAN VLAN	中间的交换机 必须支持 RSPAN VLAN
Switch A	交换机 A
Switch B	交换机 B

RSPAN source session A	RSPAN 源会话 A
RSPAN source ports	RSPAN 源端口
RSPAN source session B	RSPAN 源会话 B
RSPAN source ports	RSPAN 源端口

SPAN 和 RSPAN 的概念和术语

- SPAN 会话
- 受监控流量
- 源端口
- 源 VLAN
- VLAN 过滤
- 目的端口
- RSPAN VLAN

SPAN 会话

用户使用（本地或远端）SPAN 会话能够监控一个或多个端口、一个或多个 VLAN 的流量，并把受监控流量发送到一个或多个目的端口。

本地 SPAN 会话就是一个目的端口和源端口/源 VLAN 的关联组合，它们都位于同一台网络设备上。本地 SPAN 并没有相互分离的源和目的会话。本地 SPAN 会话会根据用户的指定，把一组入向和出向数据包集合在一起，并把它们汇集到一个 SPAN 数据流中，这个数据流最终会被转发到目的端口。

RSPAN 由至少一个 RSPAN 源会话、一个 RSPAN VLAN，以及至少一个 RSPAN 目的会话构成。用户可以分别在不同的网络设备上配置 RSPAN 源会话和 RSPAN 目的会话。为了在设备上配置 RSPAN 源会话，用户会把一组源端口或源 VLAN 与一个 RSPAN VLAN 关联在一起。这个会话的输出信息就是 SPAN 数据包流，这些数据会被发送到 RSPAN VLAN 中。为了在另一台设

备上配置 RSPAN 目的会话，用户会把目的端口与 RSPAN VLAN 关联在一起。目的会话会收集所有 RSPAN VLAN 流量并将其发送到 RSPAN 目的端口。

RSPAN 源会话与本地 SPAN 会话非常类似，只不过数据包流发往的目的地有所不同。在 RSPAN 源会话中，SPAN 数据包会被重新标记为 RSPAN VLAN ID，并通过普通的 Trunk 端口被转发到目的设备。

RSPAN 目的会话会收集 RSPAN VLAN 中收到的所有数据包，剥除掉 VLAN 标记，并发送到目的端口。目的会话会为用户提供所有 RSPAN VLAN 数据包的副本（除了二层控制数据包），以便进行分析。

一个拥有多个源和目的端口的 RSPAN 会话可以同在一个会话中，但不能与多个相同远端 VLAN 中的源会话同在一个会话中。

SPAN 会话中的流量监控有下列限制条件：

- 源可以是端口或 VLAN，但用户不能把源端口和源 VLAN 设置在同一个会话中；
- 用户可以在一台设备或设备堆栈中同时运行本地 SPAN 和 RSPAN。设备或设备堆栈总共支持 66 个源和 RSPAN 目的会话；
- 用户可以在两个不同的 SPAN 或 RSPAN 源会话中，配置不同的或重叠的 SPAN 源端口和源 VLAN。交换端口和路由端口都可以被配置为 SPAN 源和目的；
- 用户在一个 SPAN 会话中可以设置多个目的端口，但在一个设备堆栈中最多有 64 个目的端口；
- SPAN 会话不会影响设备的正常操作。但超额预订的 SPAN 目的可能会导致尾部丢弃或数据包丢失，比如通过 10 Mbit/s 端口监控 100 Mbit/s 端口；
- 当启用了 SPAN 或 RSPAN 后，每个受监控的数据包都会被发送两次，一次作为普通流量发送，一次作为监控数据包发送。监控大量端口或 VLAN 会在不知不觉中生成大量网络流量；
- 用户可以在禁用（Disabled）状态的端口上配置 SPAN 会话，但只有当目的端口和至少一个源端口/VLAN 启用后，这个 SPAN 会话才会生效；
- 设备不支持在单个会话中结合本地 SPAN 和 RSPAN：
  - RSPAN 源会话中不能有本地目的端口；
  - RSPAN 目的会话中不能有本地源端口；
  - 使用了相同 RSPAN VLAN 的 RSPAN 目的会话和 RSPAN 源会话不能运行在同一台设备或同一个设备堆栈中。

受监控流量

SPAN 会话可以监控以下流量类型：

- 接收（Rx）SPAN——接收（或入向）SPAN 能够监控源接口或源 VLAN 收到的所有数据包，并且这些数据包都是没有经过设备修改或处理的。源会收到这些数据包的副本，并将其发送到这个 SPAN 会话的目的端口。

  需要由路由或服务质量（QoS）工具进行修改的数据包是在修改前被复制的，比如修改差分服务代码点（DSCP）。

  会在服务处理期间造成数据包丢弃的特性并不会对入向 SPAN 产生什么影响；即使实际入站的数据包已被丢弃，目的端口也会收到该数据包的副本。这些特性包括 IP 标准和扩展入向访问控制列表（ACL）、入向 QoS 策略、VLAN ACL 和出向 QoS 策略；
- 传输（Tx）SPAN——传输（或出向）SPAN 能够监控源接口发送的所有数据包，并且这些数据包都经过了设备的修改和处理。源发出的所有数据包副本都会被发送到该 SPAN 会话的目的端口。副本是复制的修改后的数据包。

  由路由功能（比如修改的生存时间[TTL]、MAC 地址或 QoS 值）修改后的数据包会被复制（带有修改后的值）给目的端口。

  会导致数据包在传输处理过程中被丢弃的特性也会影响为这个 SPAN 实施的数据包复制工作。这些特性包括 IP 标准和扩展出向 ACL，以及出向 QoS 策略。
- 双向——在 SPAN 会话中，用户也可以监控端口或 VLAN 接收和发送数据包的情况。这是默认设置。

本地 SPAN 会话的默认配置是以未打标的方式发送所有数据包。但当用户在配置目的端口是使用了关键字 **encapsulation replicate**，会发生以下变化：

- 数据包在被发送到目的端口时，还携带源端口为其封装的信息（未打标或打上 IEEE 802.1Q 标记）；
- 所有类型的数据包都会被监控，其中包括 BPDU 和二层协议数据包。

因此启用了封装复制（encapsulation replicate）的本地 SPAN 会话会在向目的端口发送的数据包中包含未打标的数据包和携带 IEEE 802.1Q 标记的数据包。

设备上的拥塞会导致数据包被丢弃，其中丢弃位置包括入向源端口、出向源端口，或 SPAN 目的端口。一般来说，这些特征之间相互是独立的。举例来说：

- 数据包可能会被正常穿法，但由于 SPAN 目的端口超额订阅的关系，监控数据包会被丢弃；
- 如像数据包可能会在正常转发过程中被丢弃，但仍可能会被转发到 SPAN 目的端口；

- 由于设备拥塞而被丢弃的出向数据包，也会被出向 SPAN 丢弃。

在一些 SPAN 配置中，同一个源数据包会由多个副本被发送到 SPAN 目的端口。比如设备上配置了双向（Rx 和 Tx）SPAN 会话，接收（Rx）会话监控端口 A，发送（Tx）会话监控端口 B。如果一个数据包从端口 A 进入设备，并被交换到端口 B，那么入站和出站数据包都会被发送到目的端口。这两个数据包是相同的，除非三层信息被重写，那样的话就会由于数据包更改行为使数据包发生变化。

源端口

源端口（也称为受监控端口）可以是交换端口，也可以是路由端口，用户监控这个端口的行为来进行网络流量分析。在本地 SPAN 会话或 RSPAN 会话中，用户可以在一个方向上，或者在双方向上监控源端口或源 VLAN。设备支持任意数量的源端口（最大数量为设备上可用端口的数量）以及任意数量的源 VLAN（最大数量为设备上支持的 VLAN 数量）。虽然设备支持在本地 SPAN 或 RSPAN 中配置多个源端口或源 VLAN，但用户不能在单个会话中同时设置端口和 VLAN。

源端口拥有以下特征：

- 它可以由多个 SPAN 会话进行监控；
- 每个源端口都可以配置一个方向（入向、出向，或双向）进行监控；
- 它可以是任意端口类型（比如 EtherChannel、千兆以太网接口等）；
- 对于 EtherChannel 源，用户可以监控整个 EtherChannel，也可以只监控某个参与这个 Port-Channel 的物理端口；
- 它可以是 Access 端口、Trunk 端口、路由端口或语音 VLAN 端口；
- 它不能是目的端口；
- 源端口可以属于相同的 VLAN，或者属于不同的 VLAN；
- 用户可以在一个会话中监控多个源端口。

源 VLAN

基于 VLAN 的 SPAN（VSPAN）负责监控一个或多个 VLAN 的网络流量。VSPAN 中的 SPAN 或 RSPAN 源接口是 VLAN ID，并且 VSPAN 会监控这个 VLAN 中所有端口的流量。

VSPAN 拥有以下特征：

- 源 VLAN 中的所有活跃端口都包含在源端口中，都可以受到单向或双向监控；
- 对于某个端口来说，只有受监控 VLAN 的流量会被发送给目的端口；

- 如果目的端口属于一个源 VLAN，它自己会被排除在源列表之外，不受监控；
- 如果端口被添加到源 VLAN 后，或者从源 VLAN 中移除后，这些端口从源 VLAN 上收到的流量会被添加到监控源，或者从监控源移除；
- 用户不能在以某个 VLAN 为源的会话中，过滤这个 VLAN 的流量；
- 用户可以只监控以太网 VLAN。

VLAN 过滤

当用户把 Trunk 端口作为源端口时，默认情况下，Trunk 上所有活跃的 VLAN 都会被监控。用户可以限制在这个 Trunk 源端口上监控的 SPAN 流量，也就是使用 VLAN 过滤特性来指定 VLAN。

- VLAN 过滤特性只能应用在 Trunk 端口上，或者应用在语音 VLAN 端口上；
- VLAN 过滤特性只能应用在基于端口的会话上，不能将其应用在以 VLAN 为源的会话上；
- 当用户指定了 VLAN 过滤表后，在 Trunk 端口或语音 VLAN Access 端口上，只有列表中的这些 VLAN 会被监控；
- 其他端口类型发来的 SPAN 流量不会受到 VLAN 过滤特性的影响；也就是说其他端口上允许传输所有 VLAN 的流量；
- VLAN 过滤特性只会影响被转发到目的 SPAN 端口的流量，而不会影响正常流量的交换行为。

目的端口

每个本地 SPAN 会话或 RSPAN 目的会话都必须有一个目的端口（也称为监控端口），它负责从源端口或源 VLAN 接收流量副本，并把这些 SPAN 数据包发送给用户，通常也就是网络分析设备。

目的端口拥有以下特征：

- 对于本地 SPAN 会话来说，目的端口必须与源端口位于相同的设备或设备堆栈上。对于 RSPAN 会话来说，目的端口位于配置了 RSPAN 目的会话的设备上。在只运行了 RSPAN 源会话的设备或设备堆栈上，没有目的端口；
- 但该用户把一个端口配置为 SPAN 目的端口后，这个配置会覆盖原始的端口配置。当 SPAN 目的端口的配置被移除后，端口会恢复到自己之前的配置。如果一个端口仍是 SPAN 目的端口，那么用户对它所做的配置变更并不会生效，直到用户删除 SPAN 目的配置为止；

注释： 在 SPAN 目的端口上配置 QoS 时，QoS 会立即生效。

- 如果端口属于一个 EtherChannel 组，当用户把它配置为目的端口时，它就从 EtherChannel 组中移除了。如果它是路由端口，被配置为目的端口后，它就不再是路由端口了；
- 它可以是任意以太网物理端口；
- 它不能是安全端口；
- 它不能是源端口；
- 它可以是 EtherChannel 组（仅限于 **ON** 模式）；
- 它不能是 VLAN；
- 它同时只能参与一个 SPAN 会话（一个 SPAN 会话中的目的端口不能充当另一个 SPAN 会话中的目的端口）；
- 当它是活跃状态时，入站流量会被禁用。除了 SPAN 会话所需流量外，端口并不传输任何其他流量。目的端口上不会学到或转发任何入站流量；
- 如果用户为网络安全设备启用了入站流量转发，那么目的端口会在二层转发这些流量；
- 它不参与任何二层协议（STP、VTP、CDP、DTP、PagP）；
- 属于任意 SPAN 会话中源 VLAN 的目的端口会被排除在源列表之外，不会受到监控；
- 一台设备或一个设备堆栈中，目的端口的最大数量是 64。

本地 SPAN 和 RSPAN 目的端口的功能，在 VLAN 标记和封装上有所不同：

- 对于本地 SPAN 来说，如果用户设置了关键字 **encapsulation replicate**，那么从 SPAN 目的端口离开的出站数据包会携带原始的封装头部（未打标、ISL 或 IEEE 802.1Q）。如果没有指定这个关键字，数据包会以未打标的形式进行发送。因此对于启用了 **encapsulation replicate** 的本地 SPAN 会话来说，它的输出内容中会包含未打标、标记 ISL 或标记 IEEE 802.1Q 的数据包；
- 对于 RSPAN 来说，原始的 VLAN ID 会被 RSPAN VLAN ID 覆盖。因此目的端口上的所有数据包都是未打标的。

RSPAN VLAN

RSPAN VLAN 承载着 RSPAN 源和目的会话之间的 SPAN 流量。RSPAN VLAN 拥有以下特性：

- RSPAN VLAN 中的所有流量总是泛洪的；
- RSPAN VLAN 中没有 MAC 地址学习行为；
- RSPAN VLAN 流量只会在 Trunk 端口上传输；
- 用户必须在 VLAN 配置模式中，使用 VLAN 配置模式的命令 **remote-span**，来配置 RSPAN VLAN；
- STP 可以运行在 RSPAN VLAN Trunk 上，但不能运行在 SPAN 目的端口上；

- RSPAN VLAN 不能是私有 VLAN、主用或备用 VLAN。

对于 VLAN 1 至 1005 这些 VLAN Trunk 协议（VTP）能够识别的 VLAN 来说，VLAN ID 及其相关联的 RSPAN 特征都会由 VTP 进行传播。如果用户使用扩展 VLAN 范围（1006 至 4094）分配 RSPAN VLAN ID 的话，用户必须手动配置中间的所有设备。

网络中可能同时会有多个 RSPAN VLAN，每个 RSPAN VLAN 定义了一个网络范围内的 RSPAN 会话。也就是说，位于网络中任何位置的多个 RSPAN 源会话都可以为 RSPAN 会话提供数据包。网络中也可能会有多个 RSPAN 目的会话，监控相同的 RSPAN VLAN，并且为用户提供流量。RSPAN VLAN ID 用来区分这些会话。

SPAN 和 RSPAN 与其他特性的交互

SPAN 接口可以与以下特性进行交互：

- 路由——SPAN 不监控路由流量。VSPAN 只监控进入或离开设备的流量，不监控在 VLAN 间路由的流量。举例来说，如果用户配置了监控一个 VLAN 的接收（Rx）方向，那么当设备把其他 VLAN 的流量路由到受监控 VLAN 中时，这些流量并不会受到监控，SPAN 目的端口上也不会收到这些流量；
- STP——对于目的端口来说，当它的 SPAN 或 RSPAN 会话是活跃状态时，它不参与 STP。目的端口可以在 SPAN 或 RSPAN 会话禁用后参与 STP。对于源端口来说，SPAN 并不影响它的 STP 状态。承载 RSPAN VLAN 的 Trunk 端口上可以使用 STP；
- CDP——对于 SPAN 目的端口来说，当它的 SPAN 会话是活跃状态时，它不参与 CDP。目的端口可以在 SPAN 会话禁用后重新参与 STP；
- VTP——用户可以使用 VTP 来修剪设备之间的 RSPAN VLAN；
- VLAN 和 Trunk 技术——用户可以随时为源端口或目的端口修改 VLAN 成员关系或 Trunk 设置。但改变目的端口的 VLAN 成员关系或 Trunk 设置后，设置并不会马上生效，需要用户先移除 SPAN 目的端口配置后才会生效。为源端口改变 VLAN 成员关系或 Trunk 设置后，设置会马上生效，并且相应的 SPAN 会话会自动进行调整；
- EtherChannel——用户可以把 EtherChannel 组设置为源端口或 SPAN 目的端口。当把 EtherChannel 组配置为 SPAN 源时，整个组都会受到监控。

  如果用户把物理端口添加到一个受监控的 EtherChannel 组中，这个新加入的端口也会被放入 SPAN 源端口列表中。如果用户把一个端口从受监控的 EtherChannel 组中移除，它也会自动从源端口列表中被删除。

  用户可以把属于 EtherChannel 组的物理端口配置为 SPAN 源端口，并且它仍为 EtherChannel 的一部分。在这种情况中，当这个物理参与 EtherChannel 中的数据转发时，

来自于它的流量会受到监控。但是如果物理端口所属的 EtherChannel 组被指定为 SPAN 目的，那么它就会从这个组中移除。在端口从 SPAN 会话中移除后，它会重新加入 EtherChannel 组。端口从 EtherChannel 组中移除后，仍保留这个组的成员，但它们都处于非活跃或抑制状态。

如果属于 EtherChannel 组的物理端口是目的端口，而 EtherChannel 组是源，那么端口会从 EtherChannel 组中移除，并且也会从受监控端口的列表中移除。

- 组播流量也可以受到监控。对于出向和入向端口监控来说，只有单个未经编辑的数据包会被发送到 SPAN 目的端口。它并不会影响组播数据包发送的次数；
- 私有 VLAN 端口不能充当 SPAN 目的端口；
- 安全端口不能充当 SPAN 目的端口；

  对于 SPAN 会话来说，当在目的端口上启用了入向转发时，用户不能在受监控的出向端口上启用端口安全特性。对于 RSPAN 源会话来说，用户不能在任何受监控的出向端口上启用端口安全特性。
- IEEE 802.1x 端口可以是 SPAN 源端口。用户可以在 SPAN 目的端口上启用 IEEE 802.1x；但当用户把该端口的 SPAN 目的配置移除时，IEEE 802.1x 也会被禁用。

  对于 SPAN 会话来说，当在目的端口上启用了入向转发时，用户不能在受监控的出向端口上启用 IEEE 802.1x 特性。对于 RSPAN 源会话来说，用户不能在任何受监控的出向端口上启用 IEEE 802.1x 特性。

SPAN 和 RSPAN 以及设备堆栈

由于多台设备的堆栈表现为一台逻辑设备，因此本地 SPAN 源端口和目的端口可以分别位于堆栈中的不同设备上。因此在堆栈中添加或删除设备的操作，会影响本地 SPAN 会话，也会影响 RSPAN 源会话或目的会话。当用户从堆栈中移除一台设备后，一个活跃的会话可能变为非活跃状态；当用户向堆栈中添加一台设备后，一个非活跃的会话可能变为活跃状态。

基于流的 SPAN

用户可以通过使用基于流的 SPAN（FSPAN）或基于流的 RSPAN（FRSPAN），来控制 SPAN 或 RSPAN 会话中监控的网络流量类型，这两项特性可以在源端口的受监控流量上应用访问控制列表（ACL）。FSPAN ACL 可以用来过滤受监控的 IPv4、IPv6 和非 IP 流量。

用户可以通过接口，向 SPAN 会话应用 ACL。这个 ACL 会应用到这个 SPAN 会话中所有接口上的所有受监控流量上。ACL 中允许的数据包会被复制给 SPAN 目的端口。其他数据包不会

被复制给 SPAN 目的端口。

原始流量会继续进行转发，并且与其相关联的端口 ACL、VLAN ACL 和路由器 ACL 也都会进行应用。FSPAN ACL 不会对转发决策造成任何影响。类似的，端口 ACL、VLAN ACL 和路由器 ACL 也不会对流量监控行为带来任何影响。如果安全入向 ACL 拒绝了一个数据包，这个数据包就不会继续被转发出去，但如果 FSPAN ACL 中允许这个数据包的话，它仍会被发送到 SPAN 目的端口。但如果安全出向 ACL 拒绝了数据包而导致数据包没有被发送出去，那么这个数据包也不会被复制到 SPAN 目的端口。然而，如果安全出向 ACL 允许数据包被发送出去，那么这个数据包也只有当 FSPAN ACL 允许的情况下，才会被复制给 SPAN 目的端口。上述规则对于 RSPAN 会话来说也适用。

用户可以在 SPAN 会话上配置三种类型的 FSPAN ACL：

- IPv4 FSPAN ACL——只过滤 IPv4 数据包
- IPv6 FSPAN ACL——只过滤 IPv6 数据包
- MAC FSPAN ACL——只过滤非 IP 数据包

如果用户在一个堆栈上配置了基于 VLAN 的 FSPAN 会话，但它不适用于一台或多台设备上的硬件内存，那么这些设备就不会加载这个会话，并且从这些设备发出的匹配这个 FSPAN ACL 的流量也不会被复制到 SPAN 目的端口上。FSPAN ACL 仍会继续发挥作用，在那些它能够使用的硬件内存中，相关流量会被复制到 SPAN 目的端口。

当用户关联了一个空的 FSPAN ACL 时，有些硬件功能会由于这个 ACL，把所有流量都复制给 SPAN 目的设备。如果没有足够的硬件资源，甚至连一个空的 FSPAN ACL 都不会被加载。

所有的特性集上都支持 IPv4 和 MAC FSPAN ACL。只有高级 IP 服务特性集中才支持 IPv6 FSPAN ACL。

默认的 SPAN 和 RSPAN 配置

表 78：默认的 SPAN 和 RSPAN 配置

特性	**默认设置**
SPAN 状态（SPAN 和 RSPAN）	禁用
受监控的源端口流量	接收和发送的流量（**both**）
封装类型（目的端口）	本地方式（未打标数据包）
入向转发（目的端口）	禁用
VLAN 过滤	在将 Trunk 接口作为源端口时，监控所有 VLAN 的流量

RSPAN VLAN	未配置

配置指导

SPAN 配置指导

- 要想从 SPAN 会话中删除源或目的端口/VLAN，用户可以使用全局配置命令 **no monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*}，或使用全局配置命令 **no monitor session** *session_number* **destination interface** *interface-id*。对于目的接口来说，**no** 形式的命令中要忽略关键字 **encapsulation**；
- 要想监控 Trunk 端口上的所有 VLAN，用户需要使用全局配置命令 **no monitor session** *session_number* **filter**。

相关主题

RSPAN 配置指导

- 所有 SPAN 配置指导都适用于 RSPAN；
- 由于 RSPAN VLAN 具有独特的属性，用户应该在网络中保留一些 VLAN 用作 RSPAN VLAN；不要为这些 VLAN 分配 Access 端口；
- 用户可以为 RSPAN 流量应用一个出向 ACL，以此来有选择地过滤或监控指定数据包。用户可以在 RSPAN 源设备中，在 RSPAN VLAN 中指定这些 ACL；
- 对于 RSPAN 配置来说，用户可以把源端口和目的端口分布在网络中的多台设备上；
- RSPAN VLAN 中的 Access 端口（包括语音 VLAN 端口）会被置为非活跃（Inactive）状态；
- 用户可以把任意 VLAN 配置为 RSPAN VLAN，只要这些 VLAN 满足以下条件：
  - 在所有设备上为一个 RSPAN 会话使用相同的 RSPAN VLAN；
  - 所有参与的设备都要支持 RSPAN 特性；

FSPAN 和 FRSPAN 配置指导

- 当关联了至少一个 FSPAN ACL 后，FSPAN 就启用了；
- 当用户把至少一个非空的 FSPAN ACL 关联到 SPAN 会话时，用户不必关联一个或多个其他的 FSPAN ACL（举例来说，用户已经关联了一个非空的 IPv4 ACL，并且没有关联 IPv6

和 MAC ACL），FSPAN 会阻塞那些可能会由非关联的 ACL 过滤的流量。因此这些流量并不会被监控。

如何配置 SPAN 和 RSPAN

创建本地 SPAN 会话

用户可以按照以下步骤，创建 SPAN 会话并指定源（受监控）端口或 VLAN，以及目的（监控）端口。

总步骤

1. enable

2. configure terminal

3. no monitor session {*session_number* | **all** | **local** | **remote**}

4. monitor session *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**]

5. monitor session *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**]}

6. end

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**no monitor session** {*session_number* |	删除现有的 SPAN 会话配置。

	all | **local** | **remote**} **示例：** `Device(config)# no monitor session all`	• *session_number* 的取值范围是 1 至 66 • **all**——删除所有 SPAN 会话 • **local**——删除所有本地会话 • **remote**——删除所有远端会话
步骤 4	**monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] **示例：** `Device(config)# monitor session 1 source interface gigabitethernet1/0/1`	设置 SPAN 会话和源端口（受监控端口）。 • *session_number* 的取值范围是 1 至 66 • 在 *interface-id* 部分指定受监控的源端口。有效接口包括物理接口和 Port-Channel 逻辑接口（**port-channel** *port-channel-number*）。有效的 Port-Channel 编号为 1 至 48 • 在 *vlan-id* 部分指定受监控的源 VLAN。取值范围是 1 至 4094（不包括 RSPAN VLAN） **注释：** 一个会话中可以包含多个源（端口或 VLAN），用户需要通过多条命令对多个源进行指定；但不能在一个会话中混合使用源端口和源 VLAN。 • （可选）在**[,** | **-]**部分指定一系列端口或一个端口范围。用户需要在逗号前后各输入一个空格；也需要在连字符前后各输入一个空格 • （可选）在 **both** | **rx** | **tx** 部分指定发往监控器的流量方向。如果用户没有指定流量方向，那么源端口就会同时发送它发送和接收的流量。 · **both**——监控收到和发送的

		流量 · **rx**——监控收到的流量 · **tx**——监控发送的流量 **注释：** 用户可以多次配置 **monitor session** *session_number* **source** 命令，来指定多个源端口
步骤 5	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**]} **示例：** `Device(config)# monitor session 1 destination interface gigabitethernet1/0/2 encapsulation replicate`	指定 SPAN 会话和目的端口（监控端口）。 **注释：** 对于本地 SPAN 来说，用户必须为源接口和目的接口使用相同的会话编号。 • 在 *session_number* 部分输入步骤 4 中指定的会话编号 • 在 *interface-id* 部分指定目的端口。目的接口必须是物理端口；不能是 EtherChannel，也不能是 VLAN • （可选）在**[, | -]**部分指定一系列端口或一个端口范围。用户需要在逗号前后各输入一个空格；也需要在连字符前后各输入一个空格 （可选）**encapsulation replicate** 指定让目的接口复制源接口的封装模式。如果没有选择这个关键字，默认是以本地格式（未打标）发送数据包的。 **注释：** 用户可以多次使用命令 **monitor session** *session-number* **destination**，来配置多个目的端口
步骤 6	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式

步骤 7	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 8	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置本地 SPAN 会话并配置入站流量

用户可以按照以下步骤，创建 SPAN 会话并指定源端口或 VLAN，以及目的端口，并为网络安全设备（比如 Inspur IDS 传感器应用）在目的端口上启用入站流量。

总步骤

1. enable

2. configure terminal

3. no monitor session {*session_number* | **all** | **local** | **remote**}

4. monitor session *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**]

5. monitor session *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**] [**ingress** {**dot1q vlan** *vlan-id* | **untagged vlan** *vlan-id* | **vlan** *vlan-id*}]}

6. end

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device# configure terminal`	
步骤 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} 示例： `Device(config)# no monitor session all`	删除现有的 SPAN 会话配置。 • *session_number* 的取值范围是 1 至 66 • **all**——删除所有 SPAN 会话 • **local**——删除所有本地会话 • **remote**——删除所有远端会话
步骤 4	**monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] 示例： `Device(config)# monitor session 2 source interface gigabitethernet1/0/1 rx`	设置 SPAN 会话和源端口（受监控端口）。
步骤 5	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**] [**ingress** {**dot1q vlan** *vlan-id* | **untagged vlan** *vlan-id* | **vlan** *vlan-id*}]} 示例： `Device(config)# monitor session 2 destination interface gigabitethernet1/0/2 encapsulation replicate ingress dot1q vlan 6`	指定 SPAN 会话、目的端口、数据包封装，以及入站 VLAN 和封装。 • 在 *session_number* 部分输入步骤 4 中指定的会话编号 • 在 *interface-id* 部分指定目的端口。目的接口必须是物理端口；不能是 EtherChannel，也不能是 VLAN • （可选）在**[, | -]**部分指定一系列端口或一个端口范围。用户需要在逗号或连字符前后各输入一个空格 • （可选）**encapsulation replicate** 指定让目的接口复制源接口的封装模式。如果没有选择这个关键字，默认是以本地格式（未打标）发送数据包的。

		• 关键字 **ingress** 会在目的端口上启用入站流量转发，并指定以下封装类型： · **dot1q vlan** *vlan-id*——接收携带 IEEE 802.1Q 封装的入站数据包，并把指定 VLAN 作为默认 VLAN · **untagged vlan** *vlan-id* 或 **vlan** *vlan-id*——接收未携带标记的入站数据包封装类型，并把指定 VLAN 作为默认 VLAN
步骤 6	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 7	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 8	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

指定需要过滤的 VLAN

用户需要使用以下步骤，把 SPAN 源流量限制在指定 VLAN 中。

总步骤

1. enable

2. configure terminal

3. no monitor session {*session_number* | **all** | **local** | **remote**}

4. monitor session *session_number* **source interface** *interface-id*

5. monitor session *session_number* **filter vlan** *vlan-id* [**,** | **-**]

6. monitor session *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**]}

7. end

8. show running-config

9. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} **示例：** `Device(config)# no monitor session all`	删除现有的 SPAN 会话配置。 • *session_number* 的取值范围是 1 至 66 • **all**——删除所有 SPAN 会话 • **local**——删除所有本地会话 • **remote**——删除所有远端会话
步骤 4	**monitor session** *session_number* **source interface** *interface-id* **示例：** `Device(config)# monitor session 2 source interface gigabitethernet1/0/2 rx`	设置源端口（受监控端口）和 SPAN 会话的特征。 • *session_number* 的取值范围是 1 至 66 • 在 *interface-id* 部分指定受监控的源端口。指定的接口必须已经配置为 Trunk 端口
步骤 5	**monitor session** *session_number* **filter vlan** *vlan-id* [**,**	**-**]	把 SPAN 源流量限制在指定 VLAN 中。 • 在 *session_number* 部分输入步骤 4 中指定的会话编号

	示例： `Device(config)#` **`monitor session 2 filter vlan 1 – 5 , 9`**	• *vlan-id* 的取值范围是 1 至 4094 • （可选）用户可以使用逗号（**,**）指定一系列 VLAN，也可以使用连字符（**-**）指定一个 VLAN 范围。用户需要在逗号前后各输入一个空格；也需要在连字符前后各输入一个空格
步骤 6	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**]} **示例：** `Device(config)#` **`monitor session 2 destination interface gigabitethernet1/0/1`**	指定 SPAN 会话和目的端口（监控端口）。 • 在 *session_number* 部分输入步骤 4 中指定的会话编号 • 在 *interface-id* 部分指定目的端口。目的接口必须是物理端口；不能是 EtherChannel，也不能是 VLAN • （可选）在**[, | -]**部分指定一系列端口或一个端口范围。用户需要在逗号前后各输入一个空格；也需要在连字符前后各输入一个空格 （可选）**encapsulation replicate** 指定让目的接口复制源接口的封装模式。如果没有选择这个关键字，默认是以本地格式（未打标）发送数据包的。
步骤 7	**end** **示例：** `Device(config)#` **`end`**	返回特权 EXEC 模式
步骤 8	**show running-config** **示例：** `Device#` **`show running-config`**	检查用户输入的信息
步骤 9	**copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

	示例： `Device# copy running-config startup-config`	

把一个 VLAN 配置为 RSPAN VLAN

用户可以按照以下步骤，创建一个新 VLAN，并为 RSPAN 会话把它配置为 RSPAN VLAN。

总步骤

1. enable

2. configure terminal

3. vlan *vlan-id*

4. remote-span

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** 示例： `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 3	**vlan** *vlan-id* 示例： `Device(config)# vlan 100`	输入 VLAN ID 来创建 VLAN，或者输入已有的 VLAN ID，并进入 VLAN 配置模式。取值范围是 2 至 1001，以及 1006 至 4094。 RSPAN VLAN 不能是 VLAN 1（默认 VLAN），也不能是 VLAN ID 1002 至 1005（保留作为令牌环和 FDDI VLAN）

步骤 4	**remote-span** **示例：** `Device(config-vlan)#` **`remote-span`**	把这个 VLAN 配置为 RSPAN VLAN
步骤 5	**end** **示例：** `Device(config-vlan)#` **`end`**	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device#` **`show running-config`**	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device#` **`copy running-config startup-config`**	（可选）把输入的命令保存到配置文件中

接下来做什么？

用户必须在参与 RSPAN 的所有设备上都创建 RSPAN VLAN。如果 RSPAN VLAN 的 ID 在正常范围内（小于 1005），并且网络中启用了 VTP，用户可以在一台设备上创建 RSPAN VLAN，之后 VTP 会把这个信息传播到 VTP 域中的其他设备上。对于扩展范围的 VLAN（ID 大于 1005）来说，用户必须在源和目的设备，以及在所有中间的设备上配置 RSPAN VLAN。

用户可以使用 VTP 修剪特性来更有效地接收 RSPAN 流量，或者从所有 Trunk 中手动删除不需要承载 RSPAN 流量的 RSPAN VLAN。

要想从 VLAN 中删除远端 SPAN 特征，并且把它恢复成普通 VLAN，用户可以使用 VLAN 配置命令 **no remote-van**。

要想从 SPAN 会话中删除源端口或源 VLAN，用户可以使用全局配置命令 **no monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*}。要想从会话中删除 RSPAN VLAN，用户可以使用命令 **no monitor session** *session_number* **destination remote vlan** *vlan-id*。

创建 RSPAN 源会话

用户可以按照以下步骤，创建并启用 RSPAN 源会话，并指定受监控源和目的 RSPAN VLAN。

总步骤

1. enable

2. configure terminal

3. no monitor session {*session_number* | **all** | **local** | **remote**}

4. monitor session *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**]

5. monitor session *session_number* **destination remote vlan** *vlan-id*

6. end

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**no monitor session** {*session_number*	**all**	**local**	**remote**} **示例：** `Device(config)# no monitor session 1`	删除现有的 SPAN 会话配置。 • *session_number* 的取值范围是 1 至 66 • **all**——删除所有 SPAN 会话 • **local**——删除所有本地会话 • **remote**——删除所有远端会话
步骤 4	**monitor session** *session_number* **source** {**interface** *interface-id*	**vlan** *vlan-id*} [**,**	**-**] [**both**	**rx**	**tx**]	设置 RSPAN 会话和源端口（受监控端口）。 • *session_number* 的取值范围是 1 至

	示例： `Device(config)#` **`monitor session 1 source interface gigabitethernet1/0/1 tx`**	66 • 为 RSPAN 会话输入源端口或源 VLAN： · 在 *interface-id* 部分指定受监控的源端口。有效接口包括物理接口和 Port-Channel 逻辑接口（**port-channel** *port-channel-number*）。有效的 Port-Channel 编号为 1 至 48 · 在 *vlan-id* 部分指定受监控的源 VLAN。取值范围是 1 至 4094（不包括 RSPAN VLAN） 一个会话中可以包含多个源（端口或 VLAN），用户需要通过多条命令对多个源进行指定；但不能在一个会话中混合使用源端口和源 VLAN。 • （可选）在**[, | -]**部分指定一系列端口或一个端口范围。用户需要在逗号前后各输入一个空格；也需要在连字符前后各输入一个空格 • （可选）在 **both | rx | tx** 部分指定发往监控器的流量方向。如果用户没有指定流量方向，那么源端口就会同时发送它发送和接收的流量。 · **both**——监控收到和发送的流量 · **rx**——监控收到的流量 · **tx**——监控发送的流量
步骤 5	**monitor session** *session_number* **destination remote vlan** *vlan-id*	指定 RSPAN 会话、目的 RSPAN VLAN 和目的端口组。 • 在 *session_number* 部分输入步骤 4

	示例： `Device(config)#` **`monitor session 1 destination remote vlan 100`**	中指定的会话编号 • 在 *vlan-id* 部分指定受监控的源 RSPAN VLAN
步骤 6	**end** **示例：** `Device(config)#` **`end`**	返回特权 EXEC 模式
步骤 7	**show running-config** **示例：** `Device#` **`show running-config`**	检查用户输入的信息
步骤 8	**copy running-config startup-config** **示例：** `Device#` **`copy running-config startup-config`**	（可选）把输入的命令保存到配置文件中

指定需要过滤的 VLAN

用户需要使用以下步骤配置 RSPAN 源会话，把 RSPAN 源流量限制在指定 VLAN 中。

总步骤

1. enable

2. configure terminal

3. no monitor session {*session_number* | **all** | **local** | **remote**}

4. monitor session *session_number* **source interface** *interface-id*

5. monitor session *session_number* **filter vlan** *vlan-id* [**,** | **-**]

6. monitor session *session_number* **destination remote vlan** *vlan-id*

7. end

8. show running-config

9. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} **示例：** `Device(config)# no monitor session 2`	删除现有的 SPAN 会话配置。 • *session_number* 的取值范围是 1 至 66 • **all**——删除所有 SPAN 会话 • **local**——删除所有本地会话 • **remote**——删除所有远端会话
步骤 4	**monitor session** *session_number* **source interface** *interface-id* **示例：** `Device(config)# monitor session 2 source interface gigabitethernet1/0/2 rx`	设置源端口（受监控端口）和 SPAN 会话的特征。 • *session_number* 的取值范围是 1 至 66 • 在 *interface-id* 部分指定受监控的源端口。指定的接口必须已经配置为 Trunk 端口
步骤 5	**monitor session** *session_number* **filter vlan** *vlan-id* [**,** | **-**] **示例：** `Device(config)# monitor session 2 filter vlan 1 – 5 , 9`	把 SPAN 源流量限制在指定 VLAN 中。 • 在 *session_number* 部分输入步骤 4 中指定的会话编号 • *vlan-id* 的取值范围是 1 至 4094 • （可选）用户可以使用逗号（**,**）指定一系列 VLAN，也可以使用连字符（**-**）指定一个 VLAN 范围。用户需要在逗号前后各输入一个空格；也需要在连字符前后各输入一个空格

步骤 6	**monitor session** *session_number* **destination remote vlan** *vlan-id* **示例：** `Device(config)# monitor session 2 destination remote vlan 902`	指定 RSPAN 会话和目的远端 VLAN（RSPAN VLAN）。 • 在 *session_number* 部分输入步骤 4 中指定的会话编号 • 在 vlan-*id* 部分指定 RSPAN VLAN 来承载发往目的端口的受监控流量
步骤 7	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 8	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 9	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

创建 RSPAN 目的会话

用户可以在不同的设备或设备堆栈上配置 RSPAN 目的会话；也就是说，不在配置了源会话的设备或设备堆栈上进行配置。

用户可以按照以下步骤在相关设备上定义 RSPAN VLAN，来创建 RSPAN 目的会话并指定源 RSPAN VLAN 和目的端口。

总步骤

1. enable

2. configure terminal

3. vlan *vlan-id*

4. remote-span

5. exit

6. no monitor session {*session_number* | **all** | **local** | **remote**}

7. monitor session *session_number* **source remote vlan** *vlan-id*

8. monitor session *session_number* **destination interface** *interface-id*

9. end

10. show running-config

11. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**vlan** *vlan-id* **示例：** `Device(config)# vlan 901`	指定在源设备上创建的 RSPAN VLAN 的 VLAN ID 并进入 VLAN 配置模式。 如果源和目的设备都参与了 VTP，并且 RSPAN VLAN ID 是 2 至 1005 之间的值，就不需要配置步骤 3 至步骤 5，因为 RSPAN VLAN ID 会通过 VTP 网络进行传播。

步骤 4	**remote-span**
	
	示例：
	`Device(config-vlan)#`
	`remote-span`
		把这个 VLAN 配置为 RSPAN VLAN

步骤 5	**exit** **示例：** `Device(config-vlan)# exit`	返回全局配置模式

步骤 6	**no monitor session** {*session_number* | **all** | **local** | **remote**} **示例：** `Device(config)# no monitor session 1`	删除现有的 SPAN 会话配置。 • *session_number* 的取值范围是 1 至 66 • **all**——删除所有 SPAN 会话 • **local**——删除所有本地会话 • **remote**——删除所有远端会话
步骤 7	**monitor session** *session_number* **source remote vlan** *vlan-id* **示例：** `Device(config)# monitor session 1 source remote vlan 901`	指定 RSPAN 会话和源 RSPAN VLAN。 • *session_number* 的取值范围是 1 至 66 • 在 *vlan-id* 部分指定受监控的源 RSPAN VLAN
步骤 8	**monitor session** *session_number* **destination interface** *interface-id* **示例：** `Device(config)# monitor session 1 destination interface gigabitethernet2/0/1`	指定 RSPAN 会话和目的端口。 • 在 *session_number* 部分输入步骤 7 中指定的会话编号。 在 RSPAN 目的会话中，用户必须使用与源 RSPAN VLAN 和目的端口相同的会话编号 • 在 *interface-id* 部分指定目的端口。目的接口必须是物理端口 • 虽然命令行帮助信息中可以看到 **encapsulation replicate**，但 RSPAN 并不支持。原始 VLAN ID 会被 RSPAN VLAN ID 改写，目的端口上的所有数据包都是未打标的
步骤 9	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 10	**show running-config**	检查用户输入的信息

	示例： `Device# show running-config`	
步骤 11	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

创建 RSPAN 目的会话并配置入站流量

用户可以按照以下步骤，创建 RSPAN 目的会话并指定源 RSPAN VLAN，以及目的端口，并为网络安全设备（比如 Inspur IDS 传感器应用）在目的端口上启用入站流量。

总步骤

1. enable

2. configure terminal

3. no monitor session {*session_number* | **all** | **local** | **remote**}

4. monitor session *session_number* **source remote vlan** *vlan-id*

5. monitor session *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**ingress** {**dot1q vlan** *vlan-id* | **untagged vlan** *vlan-id* | **vlan** *vlan-id*}]}

6. end

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} **示例：** `Device(config)# no monitor session 2`	删除现有的 SPAN 会话配置。 • *session_number* 的取值范围是 1 至 66 • **all**——删除所有 SPAN 会话 • **local**——删除所有本地会话 • **remote**——删除所有远端会话
步骤 4	**monitor session** *session_number* **source remote vlan** *vlan-id* **示例：** `Device(config)# monitor session 2 source remote vlan 901`	指定 RSPAN 会话和源 RSPAN VLAN。 • *session_number* 的取值范围是 1 至 66 • 在 *vlan-id* 部分指定受监控的源 RSPAN VLAN
步骤 5	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**ingress** {**dot1q vlan** *vlan-id* | **untagged vlan** *vlan-id* | **vlan** *vlan-id*}]} **示例：** `Device(config)# monitor session 2 destination interface gigabitethernet1/0/2 ingress vlan 6`	指定 SPAN 会话、目的端口、数据包封装，以及入站 VLAN 和封装。 • 在 *session_number* 部分输入步骤 5 中指定的会话编号 在 RSPAN 目的会话中，用户必须使用与源 RSPAN VLAN 和目的端口相同的会话编号 • 在 *interface-id* 部分指定目的端口。目的接口必须是物理端口 • 虽然命令行帮助信息中可以看到 **encapsulation replicate**，但 RSPAN 并不支持。原始 VLAN ID 会被 RSPAN VLAN ID 改写，目的端口上的所有数据包都是未打标的 • （可选）在**[,** | **-]**部分指定一系列端口或一个端口范围。用户需要在逗号前后各输入一个空格；也需要在连字符前后各输入一个空格 • 关键字 **ingress** 会在目的端口上启

		用入站流量转发，并指定以下封装类型： · **dot1q vlan** *vlan-id*——接收携带 IEEE 802.1Q 封装的入站数据包，并把指定 VLAN 作为默认 VLAN · **untagged vlan** *vlan-id* 或 **vlan** *vlan-id*——接收未携带标记的入站数据包封装类型，并把指定 VLAN 作为默认 VLAN
步骤 6	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 7	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 8	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 FSPAN 会话

用户可以按照以下步骤创建 SPAN 会话、指定源（受监控）端口或 VLAN，以及目的（监控）端口，并且为会话配置 FSPAN。

总步骤

1. enable

2. configure terminal

3. no monitor session {*session_number* | **all** | **local** | **remote**}

4. monitor session *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx**

| **tx**]

5. monitor session *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**]}

6. monitor session *session_number* **filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*}

7. end

8. show running-config

9. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} **示例：** `Device(config)# no monitor session 2`	删除现有的 SPAN 会话配置。 • *session_number* 的取值范围是 1 至 66 • **all**——删除所有 SPAN 会话 • **local**——删除所有本地会话 • **remote**——删除所有远端会话
步骤 4	**monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] **示例：** `Device(config)# monitor session 2 source interface gigabitethernet1/0/1`	设置 SPAN 会话和源端口（受监控端口）。 • *session_number* 的取值范围是 1 至 66 • 在 *interface-id* 部分指定受监控的源端口。有效接口包括物理接口和 Port-Channel 逻辑接口（**port-channel**

		port-channel-number）。有效的 Port-Channel 编号为 1 至 48 • 在 *vlan-id* 部分指定受监控的源 VLAN。取值范围是 1 至 4094（不包括 RSPAN VLAN） **注释：** 一个会话中可以包含多个源（端口或 VLAN），用户需要通过多条命令对多个源进行指定；但不能在一个会话中混合使用源端口和源 VLAN。 • （可选）在**[, | -]**部分指定一系列端口或一个端口范围。用户需要在逗号前后各输入一个空格；也需要在连字符前后各输入一个空格 • （可选）在[**both | rx | tx**]部分指定发往监控器的流量方向。如果用户没有指定流量方向，那么源端口就会同时发送它发送和接收的流量。 · **both**——监控收到和发送的流量 · **rx**——监控收到的流量 · **tx**——监控发送的流量 **注释：** 用户可以多次配置 **monitor session** *session_number* **source** 命令，来指定多个源端口
步骤 5	**monitor session** *session_number* **destination** {**interface** *interface-id* **[, | -]** [**encapsulation replicate**]} **示例：** `Device(config)#` **`monitor`**	指定 SPAN 会话和目的端口（监控端口）。 • 在 *session_number* 部分输入步骤 4 中指定的会话编号 • 为 **destination** 指定以下参数： · 在 *interface-id* 部分指定目的

	session 2 destination interface gigabitethernet1/0/2 encapsulation replicate	端口。目的接口必须是物理端口；不能是 EtherChannel，也不能是 VLAN · （可选）在**[,	-]**部分指定一系列端口或一个端口范围。用户需要在逗号前后各输入一个空格；也需要在连字符前后各输入一个空格 · （可选）**encapsulation replicate** 指定让目的接口复制源接口的封装模式。如果没有选择这个关键字，默认是以本地格式（未打标）发送数据包的。 **注释：** 对于本地 SPAN 来说，用户必须使用与源和目的接口相同的会话编号。 用户可以多次使用命令 **monitor session** *session-number* **destination**，来配置多个目的端口
步骤 6	**monitor session** *session_number* **filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*} **示例：** `Device(config)# monitor session 2 filter ipv6 access-group 4`	指定 SPAN 会话、需要过滤的数据包类型，以及 FSPAN 会话使用的 ACL。 • 在 *session_number* 部分输入步骤 4 中指定的会话编号 • 在 *access-list-number* 部分指定用户希望用来过滤流量的 ACL 编号 • 在 *name* 部分指定用户希望用来过滤流量的 ACL 名称
步骤 7	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式

步骤 8	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 9	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 FRSPAN 会话

用户可以按照以下步骤来启用 RSPAN 源会话、指定受监控源和目的 RSPAN VLAN，并为会话配置 FRSPAN。

总步骤

1. enable

2. configure terminal

3. no monitor session {*session_number* | **all** | **local** | **remote**}

4. monitor session *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**]

5. monitor session *session_number* **destination remote vlan** *vlan-id*

6. vlan *vlan-id*

7. remote-span

8. exit

9. monitor session *session_number* **filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*}

10. end

11. show running-config

12. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密

	示例： `Device> enable`	码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} **示例：** `Device(config)# no monitor session 2`	删除现有的 SPAN 会话配置。 • *session_number* 的取值范围是 1 至 66 • **all**——删除所有 SPAN 会话 • **local**——删除所有本地会话 • **remote**——删除所有远端会话
步骤 4	**monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] **示例：** `Device(config)# monitor session 2 source interface gigabitethernet1/0/1`	设置 SPAN 会话和源端口（受监控端口）。 • *session_number* 的取值范围是 1 至 66 • 在 *interface-id* 部分指定受监控的源端口。有效接口包括物理接口和 Port-Channel 逻辑接口（**port-channel** *port-channel-number*）。有效的 Port-Channel 编号为 1 至 48 • 在 *vlan-id* 部分指定受监控的源 VLAN。取值范围是 1 至 4094（不包括 RSPAN VLAN） **注释：** 一个会话中可以包含多个源（端口或 VLAN），用户需要通过多条命令对多个源进行指定；但不能在一个会话中混合使用源端口和源 VLAN。 • （可选）在**[,** | **-]**部分指定一系列端

		口或一个端口范围。用户需要在逗号前后各输入一个空格；也需要在连字符前后各输入一个空格 • （可选）在[**both	rx	tx**]部分指定发往监控器的流量方向。如果用户没有指定流量方向，那么源端口就会同时发送它发送和接收的流量。 · **both**——监控收到和发送的流量 · **rx**——监控收到的流量 · **tx**——监控发送的流量 **注释：** 用户可以多次配置 **monitor session** *session_number* **source** 命令，来指定多个源端口
步骤 5	**monitor session** *session_number* **source remote vlan** *vlan-id* **示例：** `Device(config)# monitor session 2 source remote vlan 5`	指定 RSPAN 会话和目的 RSPAN VLAN。 • 在 *session_number* 部分输入步骤 4 中定义的编号 • 在 *vlan-id* 部分指定监控的目的 RSPAN VLAN
步骤 6	**vlan** *vlan-id* **示例：** `Device(config)# vlan 10`	进入 VLAN 配置模式。在 *vlan-id* 部分指定监控的源 RSPAN VLAN
步骤 7	**remote-span** **示例：** `Device(config-vlan)# remote-span`	把步骤 5 中指定的 VLAN 配置为 RSPAN VLAN
步骤 8	**exit**	返回全局配置模式

	示例： `Device(config-vlan)# exit`	
步骤 9	**monitor session** *session_number* **filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*} **示例：** `Device(config)# monitor session 2 filter ip access-group 7`	指定 RSPAN 会话、需要过滤的数据包类型，以及 FRSPAN 会话使用的 ACL。 • 在 *session_number* 部分输入步骤 4 中指定的会话编号 • 在 *access-list-number* 部分指定用户希望用来过滤流量的 ACL 编号 • 在 *name* 部分指定用户希望用来过滤流量的 ACL 名称
步骤 10	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 11	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 12	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

监控 SPAN 和 RSPAN 工作

用户可以使用下面这个表格中描述的命令来查看 SPAN 和 RSPAN 的操作配置，以及监控器运行的结果。

表 79：监控 SPAN 和 RSPAN 工作

命令	目的

show monitor	显示当前的 SPAN、RSPAN、FSPAN 或 FRSPAN 配置

SPAN 和 RSPAN 配置示例

示例：配置本地 SPAN

以下示例展示了如何设置 SPAN 会话 1，使其能够把受监控的源端口流量发送到目的端口。首先，用户删除了为会话 1 配置的现有 SPAN 配置，然后设置监控源千兆以太网端口 1 的双向流量，并将流量镜像到目的千兆以太网端口 2，保留封装方式。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1
Device(config)# monitor session 1 source interface gigabitethernet1/0/1
Device(config)# monitor session 1 destination interface gigabitethernet1/0/2 encapsulation replicate
Device(config)# end
```

以下示例展示了如何把端口 1 从 SPAN 会话 1 的 SPAN 源中删除：

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1 source interface gigabitethernet1/0/1
Device(config)# end
```

以下示例展示了如何禁用对端口 1 接收到的流量进行监控，这个端口之前被配置为监控双向流量：

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1 source interface gigabitethernet1/0/1 rx
```

对于端口 1 上接收到的流量的监控被禁用了，但从这个端口发出的流量会继续受到监控。

以下示例展示了如何移除 SPAN 会话 2 上的现有配置，并配置 SPAN 会话 2 来监控所有 VLAN 1 至 3 中端口接收到的流量，并将这些流量发送到目的千兆以太网端口 2。然后用户又添加了配置，将其变更为监控 VLAN 10 中所有端口上的所有流量。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source vlan 1 - 3 rx
Device(config)# monitor session 2 destination interface gigabitethernet1/0/2
Device(config)# monitor session 2 source vlan 10
Device(config)# end
```

以下示例展示了如何移除 SPAN 会话 2 上的现有配置，并配置 SPAN 会话 2 来监控千兆以太网源端口 1 上收到的所有流量，并将这些流量发送到目的千兆以太网端口 2，使用与源端口相同的出向封装类型，用户还启用了使用 IEEE 802.1Q 的入向转发，并把 VLAN 6 配置为默认入向 VLAN。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source gigabitethernet1/0/1 rx
Device(config)# monitor session 2 destination interface gigabitethernet1/0/2 encapsulation replicate ingress dot1q vlan 6
Device(config)# end
```

以下示例展示了如何移除 SPAN 会话 2 上的现有配置，并配置 SPAN 会话 2 来监控千兆以太网 Trunk 端口 2 上收到的所有流量，并且只把 VLAN 1 至 5，以及 VLAN 9 的流量发送到目的千兆以太网端口 1：

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source interface gigabitethernet1/0/2 rx
Device(config)# monitor session 2 filter vlan 1 - 5 , 9
Device(config)# monitor session 2 destination interface gigabitethernet1/0/1
```

```
Device(config)# end
```

示例：创建 RSPAN VLAN

以下示例展示了如何创建 RSPAN VLAN 901：

```
Device> enable
Device# configure terminal
Device(config)# vlan 901
Device(config-vlan)# remote span
Device(config-vlan)# end
```

以下示例展示了如何移除 SPAN 会话 1 上的现有配置，并配置 SPAN 会话 1 来监控多个源接口，并且把 RSPAN VLAN 901 配置为目的：

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1
Device(config)# monitor session 1 source interface gigabitethernet1/0/1 tx
Device(config)# monitor session 1 source interface gigabitethernet1/0/2 rx
Device(config)# monitor session 1 source interface port-channel 2
Device(config)# monitor session 1 destination remote vlan 901
Device(config)# end
```

以下示例展示了如何移除 SPAN 会话 2 上的现有配置，并配置 SPAN 会话 2 来监控 Trunk 端口 2 上收到的流量，并且只把 VLAN 1 至 5，以及 VLAN 9 的流量发送到目的 RSPAN VLAN 902：

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source interface gigabitethernet1/0/2 rx
Device(config)# monitor session 2 filter vlan 1 - 5 , 9
Device(config)# monitor session 2 destination remote vlan 902
Device(config)# end
```

以下示例展示了如何把 VLAN 901 配置为源远端 VLAN，把端口 1 配置为目的接口：

```
Device> enable
Device# configure terminal
Device(config)# monitor session 1 source remote vlan 901
Device(config)# monitor session 1 destination interface gigabitethernet2/0/1
Device(config)# end
```

以下示例展示了如何在 RSPAN 会话 2 中把 VLAN 901 配置为源远端 VLAN、把千兆以太网源端口 2 配置为目的接口、把 VLAN 6 设置为默认接收 VLAN，并且在接口上启用入流量转发：

```
Device> enable
Device# configure terminal
Device(config)# monitor session 2 source remote vlan 901
Device(config)# monitor session 2 destination interface gigabitethernet1/0/2 ingress vlan 6
Device(config)# end
```

其他参考资料

相关文档

相关主题	文档名称
系统命令	*Network Management Command Reference, Inspur INOS*

错误消息解码器

描述	链接
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

标准和 RFC

标准/RFC	标题
无	-

技术助手

描述	链接

Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	http://www.icntnetworks.com

SPAN 和 RSPAN 的特性历史与信息

版本	特性信息
Inspur INOS 11.3.1	引入该特性

配置 ERSPAN

这部分文档描述了如何配置封装的远端交换端口分析器（ERSPAN）。Inspur ERSPAN 特性使用户能够监控端口或 VLAN 上的流量，并把受监控流量发送到目的端口。

配置 ERSPAN 的先决条件

- 只支持 IPv4 传递/传输头部；

- 在把受监控流量发送到隧道之前应用访问控制列表（ACL）过滤；
- 只支持类型 II ERSPAN 头部。

配置 ERSPAN 的限制条件

应用这个特性有以下限制条件：

- 不支持目的会话；
- 一台设备最多支持 66 个会话。最多可以配置 8 个源会话，其他会话可以被配置为 RSPAN 目的会话。源会话可以是本地 SPAN 源会话、RSPAN 源会话，也可以是 ERSPAN 源会话；
- 用户可以把一个端口列表或一个 VLAN 列表配置为源，但不能把它们同时配置给一个会话；
- 当用户通过 ERSPAN CLI 配置了一个会话，会话 ID 和地址类型是不能改变的。要想进行变更，用户必须使用 no 格式的配置命令，来删除这个会话，然后再重新配置会话；
- ERSPAN 源会话不会从承载着 RSPAN VLAN 的源 Trunk 端口，复制本地起源的远端 SPAN（RSPAN）VLAN 流量；
- ERSPAN 源会话不会从源端口复制本地起源的 ERSPAN GRE 封装的流量。

配置 ERSPAN 的相关信息

ERSPAN 概述

用户使用 Inspur ERSPAN 特性能够监控端口或 VLAN 上的流量，并把受监控流量发送到目的端口。ERSPAN 会把流量发送给网络分析器，比如交换探针设备或远端监控（RMON）探针。ERSPAN 支持不同设备上的源端口、源 VLAN 和目的端口，能够在网络中从远端监控多个设备。

ERSPAN 支持的最大封装数据包为 9180 字节。ERSPAN 由 ERSPAN 源会话、可路由 ERSPAN GRE 封装流量，以及 ERSPAN 目的会话构成。

ERSPAN 由 ERSPAN 源会话、可路由 ERSPAN GRE 封装流量，以及 ERSPAN 目的会话构成。用户可以在一台设备上配置 ERSPAN 源会话、ERSPAN 目的会话，或者同时配置两者。如果一台设备上只配置了 ERSPAN 源会话，那么它称为 ERSPAN 源设备；如果一台设备上只配置了

ERSPAN 目的会话，那么它称为 ERSPAN 终端设备。一台设备可以充当这两种角色，也就是既是 ERSPAN 源设备，也是 ERSPAN 目的设备。

对于源端口或源 VLAN 来说，ERSPAN 可以监控它的入向、出向，或双向流量。默认情况下，ERSPAN 会监控所有流量，包括组播和桥协议数据单元（BPDU）数据帧。

ERSPAN 源会话由以下参数进行定义：

- 会话 ID
- 这个会话监控的一组源端口或源 VLAN
- 目的 IP 地址和源 IP 地址，也就是在为捕获流量封装通用路由封装（GRE）时分别使用的目的 IP 地址和源 IP 地址
- ERSPAN 流 ID
- 其他属性，比如 IP 生存时间（TTL），以及与 GRE 封装相关的参数

注释： ERSPAN 源会话不会从源端口复制 ERSPAN GRE 封装的流量。每个 ERSPAN 源会话可以把端口或 VLAN 作为源，但不能同时把端口和 VLAN 作为源。

注释： 由于封装行为是在硬件中执行的，因此 CPU 性能不会受到影响。

图 80：ERSPAN 的配置

Switch D	交换机 D
Destination Switch (Data Center)	目的交换机 （数据中心）
Probe	探针
Routed GRE-Encapsulated Traffic（共 3 处）	路由的 GRE 封装 流量
Routed Network	路由 网络
Switch A	交换机 A
Switch B	交换机 B
Source Switch(es) (Access)	源交换机 （访问层）

ERSPAN 源

Inspur ERSPAN 特性支持下列源：

- 源端口——为了执行流量分析而受到监控的源端口。源端口可以属于任意 VLAN，Trunk

端口也可以和非 Trunk 源端口一起作为源端口；

- 源 VLAN——为了执行流量分析而受到监控的 VLAN。

以下接口可以作为源端口：

- 千兆以太网端口
- PortChannel 端口
- 吉比特以太网端口

如何配置 ERSPAN

配置 ERSPAN 源会话

ERSPAN 源会话中定义了会话配置参数和受监控的端口或 VLAN。

总步骤

1. enable

2. configure terminal

3. monitor session *span-session-number* **type erspan-source**

4. description *description*

5. source {**interface** *type number* | **vlan** *vlan-ID*} [**,** | **-**| **both** | **rx** | **tx**]

6. filter {**ip access-group** {*standard-access-list* | *expanded-access-list* | *acl-name*} | **ipv6 access-group** *acl-name* | **mac access-group** *acl-name* | **vlan** *vlan-ID* [**,** | **-**]}

7. no shutdown

8. destination

9. ip address *ip-address*

10. erspan-id *erspan-ID*

11. origin *ip-address*

12. ip ttl *ttl-value*

13. end

具体配置

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式 • 在提示时输入密码

	示例： `Switch> enable`	
步骤 2	**configure terminal** **示例：** `Switch# configure terminal`	进入全局配置模式
步骤 3	**monitor session** *span-session-number* **type erspan-source** **示例：** `Switch(config)# monitor session 1 type erspan-source`	使用会话 ID 和会话类型来定义 ERSPAN 源会话，并进入 ERSPAN 监控器源会话配置模式。 • 源会话或目的会话所使用的会话 ID 都属于同一个全局 ID 空间，因此两种会话类型各自的会话 ID 在全局是唯一的 • *span-session-number* 和会话类型（通过关键字 **erspan-source** 配置的）一旦配置了就不能改变。用户可以使用 no 格式的命令删除会话，之后再使用新的会话 ID 或新的会话类型重新创建会话
步骤 4	**description** *description* **示例：** `Switch(config-mon-erspan-src)# description source1`	描述 ERSPAN 源会话
步骤 5	**source** {**interface** *type number* | **vlan** *vlan-ID*} [**,** | **-**| **both** | **rx** | **tx**] **示例：** `Switch(config-mon-erspan-crc)#`	配置受监控的源接口或 VLAN，以及流量方向

	souce interface fastethernet 0/1 rx	
步骤 6	**filter** {**ip access-group** {*standard-access-list* | *expanded-access-list* | *acl-name*} | **ipv6 access-group** *acl-name* | **mac access-group** *acl-name* | **vlan** *vlan-ID* [**,** | **-**]} **示例：** Switch(config-mon-erspan-src)# filter vlan 3	（可选）当 ERSPAN 源是 Trunk 端口时，配置源 VLAN 过滤。 • **注释：** 用户不能在一个会话中同时包含源 VLAN 和过滤 VLAN
步骤 7	**no shutdown** **示例：** Switch(config-mon-erspan-src)# no shutdown	启用配置的会话
步骤 8	**destination** **示例：** Switch(config-mon-erspan-src)# destination	定义 ERSPAN 目的会话并进入 ERSPAN 监控器目的会话配置模式
步骤 9	**ip address** *ip-address* **示例：** Switch(config-mon-erspan-src-dst)# ip address 192.0.2.9	为 ERSPAN 目的会话配置 IP 地址
步骤 10	**erspan-id** *erspan-ID* **示例：** Switch(config-mon-erspan-src-dst)# erspan-id 2	配置目的会话使用的 ID，以此识别 ERSPAN 流量
步骤 11	**origin** *ip-address* **示例：**	为 ERSPAN 流量配置作为目的使用的 IP 地址

	Switch(config-mon-erspan-src-dst)# origin ip-address 203.0.113.2	
步 骤 12	**ip ttl** *ttl-value* **示例：** Switch(config-mon-erspan-src-dst)# erspan ttl 32	为 ERSPAN 流量中的数据包配置生存时间（TTL）
步 骤 13	**end** **示例：** Switch(config-mon-erspan-src-dst)# end	退出 ERSPAN 监控器目的会话配置模式，并返回特权 EXEC 模式

ERSPAN 的配置示例

示例：配置 ERSPAN 源会话

```
Switch> enable
Switch# configure terminal
Switch(config)# monitor session 1 type erspan-source
Switch(config-mon-erspan-src)# description source1
Switch(config-mon-erspan-src)# source interface fastethernet 0/1 rx
Switch(config-mon-erspan-src)# filter vlan 3
Switch(config-mon-erspan-src)# no shutdown
Switch(config-mon-erspan-src)# destination
Switch(config-mon-erspan-src-dst)# ip address 192.0.2.9
Switch(config-mon-erspan-src-dst)# erspan-id 2
Switch(config-mon-erspan-src-dst)# origin ip-address 203.0.113.2
Switch(config-mon-erspan-src-dst)# ip ttl 32
Switch(config-mon-erspan-src-dst)# end
```

验证 ERSPAN

要想验证 ERSPAN 的配置，用户可以使用以下命令：

以下为命令 **show monitor session erspan-source** 的样例输出信息：

```
Switch# show monitor session erspan-source session
Type : ERSPAN Source Session
Status : Admin Enabled
Source Ports :
RX Only : Gi1/4/33
Destination IP Address : 192.0.2.1
Destination ERSPAN ID : 110
Origin IP Address : 10.10.10.216
IPv6 Flow Label : None
```

以下为命令 **show monitor session erspan-source detail** 的样例输出信息：

```
Switch# show monitor session erspan-source detail
Type : ERSPAN Source Session
Status : Admin Enabled
Description : -Source Ports :
RX Only : Gi1/4/33
TX Only : None
Both : None
Source VLANs :
RX Only : None
TX Only : None
Both : None
Source RSPAN VLAN : None
Destination Ports : None
Filter VLANs : None
Filter Addr Type :
RX Only : None
TX Only : None
Both : None
```

```
Filter Pkt Type :
RX Only : None
Dest RSPAN VLAN : None
IP Access-group : None
IPv6 Access-group : None
Destination IP Address : 192.0.2.1
Destination IPv6 Address : None
Destination IP VRF : None
Destination ERSPAN ID : 110
Origin IP Address : 10.10.10.216
IP QOS PREC : 0
IP TTL : 255
```

以下为命令 **show capability feature monitor erspan-source** 的样例输出信息，显示了配置的 ERSPAN 源会话：

```
Switch# show capability feature monitor erspan-source
ERSPAN Source Session Supported: true
No of Rx ERSPAN source session: 8
No of Tx ERSPAN source session: 8
ERSPAN Header Type supported: II
ACL filter Supported: true
Fragmentation Supported: true
Truncation Supported: false
Sequence number Supported: false
QOS Supported: true
```

以下为命令 **show capability feature monitor erspan-destination** 的样例输出信息，显示了所有配置的全局内置模版：

```
Switch# show capability feature monitor erspan-destination
ERSPAN Destination Session Supported: false
```

其他参考资料

相关文档

相关主题	文档名称
Inspur INOS 命令	*Inspur INOS Master Commands List, All Releases*
Inspur 6650 交换机命令	
Inspur 6850 交换机命令	

标准和 RFC

标准/RFC	标题
RFC 2784	通用路由封装（GRE）

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	http://www.icntnetworks.com

配置 ERSPAN 的特性信息

下面这个表格提供了这部分内容中描述的特性版本信息。这个表格中只列出了指定软件版本系列中，引入该特性的软件版本。除非另行说明，否则这个软件版本的后续版本也支持该特性。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导

航系统。

表 80：配置 ERSPAN 的特性信息

特性名称	版本	特性信息
ERSPAN	Inspur INOS 11.3.1	这一部分描述了如何配置封装的远端交换端口分析器（ERSPAN）。Inspur ERSPAN 特性能够使用户监控端口和 VLAN 上的流量，并把受监控流量通过任意 VRF 中的通用路由封装（GRE）隧道发送到目的端口。 在 Inspur INOS 11.3.1 中，该特性首次在 Inspur 6650 系列交换机和 Inspur 6850 系列交换机上得以实现

配置数据包捕获

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导

航系统。

配置数据包捕获的先决条件

配置数据包捕获的先决条件

- Inspur 6850 和 Inspur 6650 都支持数据包捕获（Packet Capture）特性；
- 只有运行 IP Base 镜像或 IP Services 镜像的交换机上才支持使用 Wireshark；
- 只有运行 LAN Base 镜像的交换机上才支持内嵌数据包捕获特性。

内嵌数据包捕获（EPC）软件子系统会在运行期间消耗 CPU 和内存资源。用户必须有足够的系统资源来支持不同类型的操作。下面这个表格中提供了一些需要占用的系统资源说明。

表 81：EPC 子系统需要的系统资源

系统资源	**需求**
硬件	CPU 的利用率需求与平台有关
内存	数据包缓存是存储在 DRAM 中的。数据包缓存的大小由用户指定
硬盘空间	EPC 可以把数据包发送到外部设备上。不需要在 Flash 硬盘上提供中间存储空间

配置数据包捕获的限制条件

配置数据包捕获的限制条件

- 从 Inspur INOS 11.3.1 版本开始，不再支持 Wireshark 上的全局数据包捕获功能；
- Wireshark 上支持显示过滤器；
- 用户在 CLI 中配置 Wireshark 时，要求只能在 EXEC 模式中执行特性命令。通常在配置子模式（比如定义捕获点）中定义的行为也是在 EXEC 模式中进行设置的。所有重要命令都不具备 NVGEN（非易失性内容）特点，并且不会同步到 NSF 和 SSO 环境中的备用引擎中；
- 在接口出方向上捕获的数据包可能不会携带重写后的信息（其中包括 TTL、VLAN 标记、CoS、校验和、MAC 地址、DSCP、优先级、UP 等）；

- 不捕获入向和出向数据包的重写信息；
- 不支持通过文件大小来限制循环文件的存储；
- 文件限制继承了 IP Base 和 IP Services 镜像中 Flash 的限制大小；
- IP Base 和 IP Services 镜像中支持协议解码，~~比如无线接入点的控制和部署（CAPWAP）~~；
- 在 IP Base 和 IP Services 镜像中，在文件模式下，数据包会写入到文件中，而无需导出；
- LAN Base 镜像能够支持嵌入 Wireshark，并且拥有以下限制条件：
  - 不支持捕获过滤器和显示过滤器；
  - 不支持活跃捕获解码；
  - 输出格式与以前的版本有所不同。
- 内嵌数据包捕获（EPC）只能够捕获入方向上的组播数据包，而不能捕获出方向上的复制数据包。

配置限制条件

- 最多可以定义 8 个捕获点，但同时只能有一个处于活跃状态。用户需要在开启另一个之前，先停止正在运行的捕获点；
- VRF、管理端口和私有 VLAN 都不能用作接合点；
- Wireshark class-map 中只能使用一个 ACL（IPv4、IPv6 或 MAC）；
- Wireshark 不能在 SPAN 目的端口上捕获数据包；
- 当连其中一个接合点（接口）所连接的捕获点停止工作时，Wireshark 也会停止捕获数据包。比如接合点上所关联的设备从该交换机上拔出的情况。要想恢复捕获行为，用户必须手动进行重新启用；
- 需要 CPU 进行处理的数据包被认为是控制平面数据包。因此接口上的出向捕获功能并不会捕获这类数据包；
- MAC ACL 只用来过滤非 IP 数据包，比如 ARP。三层端口或 SVI 接口上不支持使用 MAC ACL；
- MAC 过滤器并不捕获 IP 数据包，哪怕这个数据包的 MAC 地址与之相匹配。这一点适用于所有接口（二层交换接口和三层路由端口）；
- MAC 过滤器不会在三层接口上捕获二层数据包（ARP）；
- VACL 中不支持基于 IPv6 的 ACL；
- 不支持二层 EtherChannel；
- 从 Inspur INOS 16.1 版本开始支持三层 PortChannel；
- 当捕获功能已经处于活跃状态或者已经开启时，用户不能修改捕获点参数；
- ACL 日志记录和 Wireshark 不兼容。一旦激活了 Wireshark，它就拥有了优先权。所有流量都会被重定向到 Wireshark，其中包括应该由端口上的 ACL 日志记录功能捕获的数据包。建议用户在开始使用 Wireshark 前先禁用 ACL 日志记录功能。否则 Wireshark 流量

会受到 ACL 日志流量的影响；

- Wireshark 不会捕获由 Floodblock 丢弃的数据包；
- 如果用户在同一个端口上同时使用 PACL 和 RACL 进行捕获，只会有一个副本会发送到 CPU。如果用户捕获到 DTLS 加密的 CAPWAP 接口，会有两个副本发送到 Wireshark，一个加密的，另一个解密的。在如果用户捕获二层接口上承载的 DTLS 加密 CAPWAP 流量，效果也是相同的。核心过滤其是基于外部 CAPWAP 头部设置的；
- 从 Inspur INOS 16.1 版本开始：
  - 开始支持三层 PortChannel；
  - 显示格式发生了些许变化；
  - 能够在 cap 文件中显示数据包数量；
  - 在清除捕获缓存的同时删除它的内容。不能在数据包捕获处于活跃状态时执行；
  - 为控制平面捕获添加了警告消息；
  - 在缓存模式中，只能在捕获停止后显示数据包；
  - 在 IP Services 和 IP Base 镜像中，在捕获停止后显示数据包统计状态信息；
  - 能够查询 pcap 文件中捕获的数据包数量；
  - 在以 cap 文件进行查看时，可以使用数据包编号来查看所选数据包的详细信息；
  - 文件模式中可以使用显示过滤器；
  - 可以在捕获期间或捕获停止后，显示数据包捕获的统计状态信息（接收、丢弃的数据包数和字节数）；
  - 与 Wireshark 一样，系统可以查询 pcap/cap 文件内容的统计状态信息；
  - 不管缓存的大小是多少，数据包捕获会话总是使用流模式。不再使用 Lock-Step 模式；
  - 只有 LAN Base 镜像中支持在活跃的捕获点上清除缓存，这种行为只会清除其内容。在所有其他许可镜像中，这种行为会删除缓存本身，因此不能在活跃捕获期间使用。

警告： 控制平面数据包是没有速率限制和性能影响的。用户需要使用过滤器来限制控制平面的数据包捕获。

- 如果用户改变了接口模式，从交换端口改为路由端口（二层改为三层）或者反之，必须在接口启用后删除捕获点并重新进行创建。仅仅停止/开启捕获点的行为不解决问题；
- 如果用户删除了活跃捕获会话使用的文件，捕获会话无法创建新文件，接下来不活的所有数据包都会丢失。用户需要重新启用捕获点

数据包捕获介绍

数据包捕获工具概述

数据包捕获特性是一项内置的数据包捕获功能，使网络管理员能够捕获流向、流经和离开设备的数据包，并在本地进行分析，或者保存并导出捕获的数据包，继而使用 Wireshark 和内嵌数据包捕获（EPC）等工具进行离线分析。这个特性简化了网络运维工作，它使设备变成了网络管理和运维工作中的主动参与者。它能够通过收集数据包格式信息提供排错功能，还提供了应用分析和安全功能。

LAN Base 镜像能够支持内嵌数据包捕获特性。IP Base 和 IP Services 镜像能够支持内嵌数据包捕获特性和 Wireshark。

有关 Wireshark 的信息

Wireshark 概述

Wireshark 是一款数据包分析程序，以前称为 Ethereal，它能够支持多种协议，并以基于文本的用户界面来呈现相关信息。

捕获和分析流量的功能能够提供网络活动中的数据。SPAN 和 debug 平台数据包都有使用限制。SPAN 是捕获数据包的理想选择，但它只能把数据包转发到某些指定的本地或远端目的地；它不提供本地显示或分析功能。

因此用户需要一种流量捕获和分析机制，能够同时应用于硬件和软件转发的流量，并且提供强大的数据包捕获、显示和分析功能，并最好使用一个常见接口。

Wireshark 会使用名为.pcap 的常见格式，把数据包储存到一个文件中，用户可以在某个接口上应用或启用 Wireshark。用户需要在 EXEC 模式中指定接口、过滤器和其他参数。Wireshark 应用只会在用户输入了命令 **start** 后才开始应用，并且它会在 Wireshark 停止捕获数据包时，自动或手动移除。

注释： 当前交换机中安装的 Wireshark 版本是 1.10.8。

捕获点

捕获点（Capture Point）是 Wireshark 特性的中心策略定义。捕获点描述了所有与指定 Wireshark 实例相关联的特征：捕获哪些数据包、从哪里捕获数据包、对捕获的数据包做什么，以及何时停止捕获。用户可以在创建捕获点后对其进行修改，但在使用命令 **start** 激活后无法修改。这个过程叫做激活捕获点或启用捕获点。捕获点是通过名称进行识别的，可以手动或自动停用或停止。

用户可以定义多个捕获点，但同时只能有一个处于活跃状态。用户需要在开启另一个之前，先停止正在运行的捕获点。

在堆栈系统中，用户可以在活跃的成员上激活捕获点。故障倒换会终结所有活跃的数据包捕获会话，用户需要重新启用数据包捕获行为。

接合点

接合点（Attachment Point）是把逻辑数据包处理路径和捕获点关联在一起的点。接合点是捕获点的一个属性。进入接合点的数据包会与捕获点过滤器进行比较；匹配的数据包会被复制并发送到与这个捕获点实例相关联的 Wireshark。一个捕获点可以与多个接合点相关联，但不能混用不同类型的接合点。在用户指定不同类型的接合点时，也有一些限制条件。接合点是有方向性的（入向、出向或双向），但二层 VLAN 接合点没有方向性，它总是双向的。

在堆栈系统中，所有堆栈成员上的接合点都是有效的。EPC 会从所有定义的接合点捕获数据包。但这些数据包只会在活跃成员上进行处理。

过滤器

过滤器是捕获点的一个属性，它定义并限制了穿越（与捕获点相关的）接合点的流量子集，这些流量会被复制并发送到 Wireshark。要想在 Wireshark 中进行显示，数据包必须穿越接合点，以及与捕获点相关联的过滤器。

捕获点拥有以下类型的过滤器：

- 核心系统过滤器——核心系统过滤器是由硬件提供的，它的匹配条件由硬件提供限制。这种过滤器决定了是否将硬件转发流量，为了使用 Wireshark 进行分析的目的，复制到软件中；
- 显示过滤器——显示过滤器是由 Wireshark 提供的。不匹配显示过滤器条件的数据包不

会被显示出来

核心系统过滤器

用户可以使用 class-map 或 ACL，或者使用 CLI 来定义核心系统过滤器的匹配条件。

注释： 在把 CAPWAP 定义为接合点时，不能使用核心系统过滤器。

在一些网络环境中，用户需要获得授权才能修改配置，如果审批流程比较长会导致严重的延迟。这种情况会限制网络管理员对流量实施监控和分析的能力。为了解决这个问题，Wireshark 支持使用 EXEC 模式的 CLI 命令来指定核心系统过滤器的匹配条件。用户在这种方式中能够指定的匹配条件只是 class-map 中支持的一部分，比如 MAC 地址、IP 源和目的地址、以太类型、IP 协议、TCP/UDP 源和目的端口。

如果用户倾向于使用配置模式，就可以定义 ACL，或使用 class-map 把捕获点调用给 ACL。用户可以在 class-map 和 policy-map 的结构下，明确指定匹配条件或使用 ACL 来指定匹配条件。

要注意，ACL 和 class-map 的配置都是系统的一部分，并不是 Wireshark 特性的内容。

显示过滤器

通过使用显示过滤器，用户可以让 Wireshark 进一步缩小展示数据包的范围，只从.pcap 文件中解码并展示一部分数据包。

相关主题

其他参考资料

行为

用户可以在 Wireshark 中调用实时流量或已保存的.pcap 文件。在调用实时流量时，它可以针对穿越显示过滤器的数据包执行四种类型的行为：

- 缓存到内存中来解码和分析并储存
- 储存为.pcap 文件
- 解码并显示
- 储存并显示

在调用.pacp 文件时，只能应用解码和显示行为。

把捕获的数据包储存到内存缓存中

数据包可以被储存到内存中的捕获缓存中，以便后续解码、分析，或储存为.pcap 文件。

捕获缓存可以是线性模式或循环模式。在线性模式（Linear Mode）中，当缓存已满时新数

据包会被丢弃。在循环模式（Circular Mode）中，当缓存已满时最老的数据包会被丢弃，为新数据包留出空间。尽管用户可以按需清除缓存，但这个模式主要用于调试（debugging）网络流量。但用户不能只清除缓存的内容而不删除缓存本身。因此用户需要停止并再启用当前的捕获行为，才能使这个行为生效。

注释： 如果在一个缓存中有多个捕获实例在储存数据包，用户需要在启用新的捕获实例前清除缓存，以防止内存丢失。

把捕获的数据包储存到.pcap 文件中

注释： 在堆栈中的交换机上使用 Wireshark 时，捕获的数据包只能储存在 Flash 或连接在活跃交换机上的 USB 闪存中。

举例来说，如果活跃交换机上连接了 Flash1，备用交换机上连接了 Flash2，那么只有 Flash1 能够用来储存捕获的数据包。

如果用户尝试在 Flash 或连接在活跃交换机上的 USB 闪存之外的地方储存捕获的数据包，可能会导致发生错误。

Wireshark 可以把捕获的数据包储存到.pcap 文件中。捕获的文件可以位于以下存储设备上：

- 内嵌 Flash 存储（flash:）
- USB 闪存（usbflash0:）

注释： 如果用户尝试在不支持的设备或未连接在活跃交换机的设备上储存捕获的数据包，可能会导致发生错误。

在配置 Wireshark 捕获点时，用户可以为其关联一个文件名。当用户激活捕获点时，Wireshark 会使用指定名称创建一个文件，并把数据包写入这个文件中。如果在创建捕获点时文件已存在，Wireshark 会询问用户是否要覆盖这个现有文件。如果在激活捕获点时文件已存在，Wireshark 会直接覆盖这个现有文件。只有一个捕获点可以关联到指定文件名。

如果 Wireshark 写入的目的文件系统已满，Wireshark 的写入工作就会失败，文件中只会有部分数据。因此用户必须在开启捕获会话前，确保文件系统中有足够的空间。在使用 Inspur INOS 11.3.1 版本时，有些储存设备的文件系统空间状态无法检测。

用户可以通过只保留数据包中的一部分，而不是整个数据包，来减少捕获工作所需的存储空间。通常用户并不需要前 64 或 128 字节的内容。默认行为是储存完整的数据包。

要想避免在处理和写入文件系统的过程中，有可能出现的数据包丢弃行为，Wireshark 可以（可选的）使用内存缓存，在数据包到达时暂时保存数据包。用户在把捕获点与.pcap 文件进行关联时，可以指定内存缓存的大小。

数据包解码和显示

Wireshark 可以解码并向控制器显示数据包。应用于实时流量的捕获点和应用到现有.pcap 文件的捕获点都可以使用这个功能。

注释： 解码和显示数据包可能会对 CPU 带来负担。

用户可以查看多种数据包格式的详细信息。用户可以通过命令 **monitor capture name start**，查看详细信息，并使用下列关键字选项选择不同的显示和解码模式：

- brief——每个数据包显示一行（默认）；
- detailed——解码并显示所有数据包的所有字段，同时 Wireshark 需要支持该数据包的协议。这个模式比其他两个模式需要更多的 CPU 资源；
- （十六进制）dump——每个数据包显示一行，包括数据包数据的十六进制转储格式，以及每个数据包中可以显示的字符。

在用户在 **capture** 命令中使用解码和显示选项时，Wireshark 的输出内容会返回 Inspur INOS，并在终端上原封不动地进行显示。

显示实时流量

Wireshark 会从核心系统接收到数据包的副本。Wireshark 会应用显示过滤器来丢弃用户不感兴趣的数据包，然后解码并显示其他数据包。

显示.pcap 文件

Wireshark 可以从之前储存的.pcap 文件中解码并显示数据包，然后之间对选择的数据包执行显示过滤器。

数据包储存和显示

从功能上看这个模式是前两个模式的集合。Wireshark 会把数据包储存在指定的.pcap 文件中，然后把它们解码并显示在终端控制台上。在这中模式中只能应用核心过滤器。

Wireshark 捕获点的激活和停用

在用户定义了 Wireshark 捕获点、接合点、过滤器、行为和其他选项后，还必须激活捕获点。在捕获点被激活后，它才会真正开始捕获数据包。

在用户激活捕获点之前，先要执行一些功能性检查。如果捕获点上没有关联核心系统过滤器，也没有关联接合点的话，用户不能激活它。如果用户尝试激活不满足要求的捕获点，会造成

错误。*

用户可以根据需要来定义显示过滤器。

在用户激活了 Wireshark 捕获点后，可以有多种方式将其停用。对于只把数据包储存到.pcap 文件中的捕获点来说，用户可以手动将其停用，也可以配置一个时间或指定数据包数量，达到要求后捕获点会自动停用。

在用户激活了 Wireshark 捕获点后，一个固定速率的限速器会自动应用到硬件中，这样 CPU 就不会被 Wireshark 数据包淹没了。使用速率限制的劣势在于即使有更多资源可用，用户也不能在超出设定的限速时，持续不断地捕获数据包。

数据包捕获速率被设置为每秒 1000 个数据包（pps），1000 pps 的限制是应用在所有接合点上的。举例来说，如果一个捕获会话共有 3 个接合点，那么所有这 3 个接合点的速率加在一起不能超过 1000 pps。

注释： 对于控制平面的数据包捕获功能不能使用限速器。当用户激活了控制平面捕获点后，需要格外小心不要让流量淹没 CPU。

Wireshark 特性

这一部分描述了网络环境中的 Wireshark 特性功能：

- 如果一个出向捕获实例上应用了端口安全特性和 Wireshark，那么 Wireshark 仍会捕获到由端口安全特性丢弃的数据包。如果一个入向捕获实例上应用了端口安全特性，而 Wireshark 在执行出向数据包捕获，那么 Wireshark 不会捕获到由端口安全特性丢弃的的数据包；
- Wireshark 不会捕获到由动态 ARP 监测（DAI）特性丢弃的数据包；
- 如果用户把处于 STP 阻塞状态的端口设置为接合点，并且匹配核心过滤器，那么 Wireshark 就会捕获到进入这个端口的数据包，即使数据包会被交换机丢弃；
- 基于分类的安全特性——与接合点连接在同一层的 Wireshark 捕获点无法捕获由入向基于分类的安全特性（比如 ACL 和 IPSG）丢弃的数据包。相反，与接合点连接在同一层的 Wireshark 捕获点能够捕获由出向基于分类的安全特性丢弃的数据包。这个逻辑是这样的：Wireshark 接合点的工作后于入向的端口安全特性，先于出向的端口安全特性。在入方向上，数据包可以穿越二层端口、VLAN 和三层端口/SVI。在出方向上，数据包可以穿越三层端口/SVI、VLAN 和二层端口。如果接合点工作的位置先于数据包被丢弃的位置，Wireshark 就能捕获到数据包。否则，Wireshark 就不能捕获到数据包。举例来说，二层接合点入方向上的 Wireshark 捕获策略，能够捕获由三层基于分类的安全特性丢弃的数据包。同样的，三层接合点出方向上的 Wireshark 捕获策略，能够捕获由二层

基于分类的安全特性丢弃的数据包；

- 路由端口和交换机虚拟接口（SVI）——Wireshark 不能捕获 SVI 上的出向数据包，因为从 SVI 发出的数据包都是由 CPU 生成的。要想捕获这些数据包，用户需要把控制平面作为接合点；
- VLAN——从 Inspur INOS 16.1 版本开始，当用户把 VLAN 作为 Wireshark 接合点时，Wireshark 能够在二层和三层端口上捕获入方向和出方向上的数据包；
- 重定向特性——在入方向上，通过三层进行重定向的特性流量（比如 PBR 和 WCCP）在逻辑上后于三层 Wireshark 接合点。Wireshark 会捕获到这些数据包，即使它们可能之后会被重定向到其他三层接口上。类似的，通过三层进行重定向的出向特性（比如出向 WCCP），在逻辑上先于三层 Wireshark 接合点，因此 Wireshark 无法捕获这些数据包；
- SPAN——Wireshark 不能捕获 SPAN 目的接口上的流量；
- SPAN——Wireshark 能够在入方向上捕获 SPAN 源接口的流量，也可能能够捕获出方向上的数据包；
- 用户可以同时捕获最多 1000 个 VLAN 中的数据包，如果没有使用 ACL 实施限制的话。如果用户应用了 ACL，那么能够供 Wireshark 使用的硬件资源空间就会减少。因此，同一时间能够用于数据包捕获的 VLAN 总数量就会减少。同时使用超过 1000 个 VLAN 隧道，或者大量 ACL，可能会引起不可预测的结果。比如移动性可能会降低。

注释： 强烈不建议用户同时对大量接合点进行捕获操作，因为这会带来过于繁重的 CPU 利用率和不可预测的硬件行为。

Wireshark 指导

- 在 Wireshark 进行数据包捕获期间，也可以同时进行硬件转发；
- 在开启 Wireshark 捕获进程前，用户要确保 CPU 的利用率不高，并且有足够的内存（至少 200 MB）可用；
- 如果用户计划把数据包储存到文件中，要在开启 Wireshark 捕获进程前，确保有足够的空间可用；
- Wireshark 捕获期间的 CPU 利用率取决于有多少数据包匹配了指定的条件，以及对于匹配的数据包需要执行什么操作（储存、解码和显示，或者两者同时进行）；
- 只要有可能，用户应该尽量把捕获的影响降到最低（限制数据包数量和持续时间），以避免高 CPU 利用率和其他不可预测的结果；
- 由于数据包转发通常发生在硬件中，因此数据包不会被复制到 CPU 来执行软件处理。对于 Wireshark 的数据包捕获行为来说，数据包会被复制并传输给 CPU，这会增加 CPU

的利用率。

为了避免出现高 CPU 利用率，用户可以采取以下行为：

- 只接合相关端口；
- 使用 class-map，和次一级的 access-list 来描述匹配条件。如果可以的话，使用明确配置的在线过滤器；
- 严格遵守过滤规则。限制流量类型（比如只匹配 IPv4），而不是使用宽泛的 ACL，因为这样会带来多余流量。

- 总是把数据包捕获行为限制在较短的时间段内，或者限制较少的数据包数量。用户可以使用捕获命令中的参数指定以下信息：
  - 捕获时间段
  - 捕获的数据包数量
  - 文件大小
  - 数据包分段大小
- 如果用户知道只会有非常少的流量能够匹配核心过滤器，那么可以不加限制地运行捕获进程；
- 用户可能会在遇到以下事件时，遭遇高 CPU（或内存）利用率：
  - 用户开启一个捕获会话，并无意间让它运行了很长一段时间，这会导致意外的流量激增；
  - 用户开启一个捕获会话并使用循环文件或捕获缓存，然后无意间让它运行了很长一段时间，这会引起性能或系统健康问题。
- 在捕获会话运行期间，要小心 Wireshark 可能会对性能或系统健康带来影响，比如高 CPU 利用率和内存消耗。如果发生了这些情况，用户要马上停用 Wireshark；
- 避免从较大的.pcap 文件中解码并显示数据包。而是把.pcap 文件传输到 PC，然后在 PC 上运行 Wireshark；
- 用户最多可以定义 8 个 Wireshark 实例。用户可以使用 **show** 命令，从.pcap 文件中或从计为一个实例的捕获缓存中解码并查看数据包。但只能有一个实例处于活跃状态；
- 当一个捕获文件正在运行，并且与之相关联的 ACL 发生了修改时，用户必须重新开启这个会话，才能使 ACL 的变更生效。如果用户不重新开启捕获进程，它就会持续使用原始的没有经过变更的 ACL；
- 要想避免数据包丢失，用户需要考虑以下内容：
  - 在捕获实时数据包时，（在指定显示选项时）只使用储存，而不是选择解码和保存，储存并不是 CPU 敏感的操作（尤其是在详细模式中）；

- 如果用户配置了多个捕获实例向缓存中储存数据包，那么要在开启新的捕获实例前清除缓存，防止内存丢失；
- 如果用户使用默认缓存大小，并发现正在丢失数据包，就需要增加缓存大小来避免丢包；
- 写入 Flash 硬盘不是 CPU 敏感的操作，因此如果捕获速率不足的话，用户可以使用缓存捕获；
- Wireshark 捕获会话总是以流的形式运行，速率为 1000 pps；

• 流捕获模式的速率是 1000 pps；
• 如果用户希望在控制台窗口中解码并显示实时数据包，要确保 Wireshark 会话只运行很短的时间；

警告： 如果以长时间运行 Wireshark 会话，或者不指定捕获时长（使用命令 **term len 0** 设置一次性显示），可能会导致控制台或终端程序不可用。

• 在使用 Wireshark 捕获实时流量时，如果会导致高 CPU 利用率，用户应该考虑应用 QoS 策略来暂时限制实际的流量，直到捕获进程结束为止；
• 所有与 Wireshark 有关的命令都是 EXEC 模式的；没有配置模式的 Wireshark 命令。如果用户需要在 Wireshark CLI 中使用 access-list 或 class-map，必须使用配置模式的命令来定义 access-list 和 class-map；
• 在定义捕获点时没有特殊的顺序要求；用户可以以任意顺序定义 CLI 允许用户设置的捕获点参数。Wireshark CLI 允许用户在一条命令中能配置多少个参数就配置多少个参数。这就减少了配置捕获点所需的命令数量；
• 所有参数都可以使用一个值，除了结合点之外。通常用户可以通过重新输入命令来使用新的值。但用户配置后，系统会用新的值覆盖旧的配置。用户要想配置新的值，不必使用 **no** 格式的命令，但需要移除相关参数；
• Wireshark 允许用户指定一个或多个接合点。要想添加多个接合点，用户需要为每个新的接合点重新输入一次命令。要想移除一个接合点，用户需要使用 **no** 格式的命令。用户可以把一个接口范围指定为一个接合点。举例来说，用户可以输入命令 **monitor capture mycap interface GigabitEthernet1/0/1 in**，把 GigabitEthernet1/0/1 配置为一个接合点。

  如果用户还需要把接口 GigabitEthernet1/0/2 也指定为接合点，可以另配置一条命令：**monitor capture mycap interface GigabitEthernet1/0/2 in**
• 但捕获点处于活跃状态时，用户不能对其进行修改；
• 用户希望做出的行为，决定了哪些参数是必需的。Wireshark CLI 允许用户在输入 **start** 命令前进行指定或修改。当用户输入了 **start** 命令后，Wireshark 只会在确定了所有必需

的参数都已配置后才会启动；

- 如果在创建捕获点时文件已经存在，Wireshark 会询问用户是否覆盖原文件。如果在激活捕获点时文件已经存在，Wireshark 会直接覆盖已有文件；
- 用户可以通过明确过滤器、access-list 或 class-map 来配置核心过滤器。指定这些类型的新过滤器会替换已有的过滤器；

注释： 核心过滤器是必需的，除非用户使用 CAPWAP 隧道接口作为捕获点接合点。

- 用户可以使用明确的 **stop** 命令，或者在自动显示模式中输入 **q** 来终结 Wireshark 会话。当触发了停止条件时，会话也会自动终结，比如达到了时长限制或者数据包捕获限制；如果发生了内部错误，或资源已满（尤其是文件模式中的硬盘已满）时，会话也会自动终结；
- 被丢弃的数据包不会显示在捕获实例的末端。但会显示丢弃的数据包数量和总大小。

默认 Wireshark 配置

下面这个表格中展示了默认的 Wireshark 配置。

特性	**默认设置**
时长	未限制
数据包	未限制
数据包长度	未限制（完整的数据包）
文件大小	未限制
循环文件储存	否
缓存储存模式	线性

内嵌数据包捕获的相关信息

内嵌数据包捕获概述

内嵌数据包捕获（EPC）提供了一种内嵌的系统管理功能，能够帮助用户追踪和排查数据包问题。这个特性使网络管理员能够捕获穿越、去往和来自 Inspur 设备的数据流量。网络管理员可以定义捕获缓存的大小和类型(循环或线性),以及每个数据包捕获的最大字节数量。用户可以使用其他管理控制手段来限制数据包捕获的速率。举例来说，用户可以使用访问控制列表来设置捕获数据包的过滤条件，或者（可选的）也可以定义最大数据包捕获速率，或

者定义采样间隔。

内嵌数据包捕获的优势

- 能够捕获设备中的 IPv4 和 Ipv6 数据包，也能够通过使用 MAC 过滤器或匹配 MAC 地址的方式，捕获非 IP 数据包；
- 能够在可扩展的基础设施上启用数据包捕获点。捕获点就是流量穿越的点，数据包就是在这里被捕获的，并关联到一个缓存中；
- 能够把捕获的数据包以数据包捕获文件（PCAP）的格式，导出到任意外部工具来进行分析；
- 可以按照各种详细程度来解码捕获的数据包。

数据包中的数据捕获

数据包中的数据捕获是指捕获到数据包后，把它储存到缓存中。用户可以为数据包的数据捕获定义唯一的方法和参数。

用户可以在捕获实例上执行以下行为：

- 在任意接口激活捕获实例；
- 在捕获点上应用访问控制列表（ACL）或 class-map；

  注释： 不支持基于网络的应用识别（NBAR）和 MAC 形式的 class-map。
- 破坏捕获；
- 指定缓存储存参数，比如大小和类型。大小的范围是 1 MB 至 100 MB。默认缓存为线性；其他的缓存选项还有循环；
- 指定匹配条件，其中包括有关协议的信息、IP 地址或端口地址。

配置数据包捕获

如何配置 Wireshark

用户可以按照以下步骤来配置 Wireshark：

1. 定义捕获点

2. （可选）添加或修改捕获点的参数

3. 激活或停用捕获点

4. 不再使用使删除捕获点

定义捕获点

这个示例展示了定义一个简单的捕获点所需的步骤。如果用户希望的话，可以定义一个捕获点，并使用命令 **monitor capture** 在一个实例中定义所有参数。

注释： 用户必须定义接合点、捕获方向，以及核心过滤器，才能使捕获点发挥作用。

用户可以按照以下步骤来定义捕获点。

总步骤

1. enable

2. show capwap summary

3. monitor capture {*capture-name*}{**interface** *interface-type interface-id* | **control-plane**}{**in** | **out** | **both**}

4. monitor capture {*capture-name*}[**match** {**any** | **ipv4 any any** | **ipv6**} **any any**}]

5. show monitor capture {*capture-name*}[**parameter**]

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**show capwap summary** **示例：** `Device# show capwap summary`	显示 CAPWAP 汇总信息
步骤 3	**monitor capture** {*capture-name*}{**interface** *interface-type interface-id*	**control-plane**}{**in**	**out**	定义捕获点，指定这个捕获点相关联的接合点，指定捕获方向。 命令中的关键字含义如下所示：

	both} **示例：** Device# **monitor capture mycap interface GigatitEthernet1/0/1 in**	• *capture-name*——指定用户要定义的捕获点名称（示例中使用了mycap）。捕获名称（Caputre Name）应该不大于 8 个字符。只能使用字母和下划线（_） • （可选）**interface** *interface-type interface-id*——指定捕获点所关联的接合点（示例中使用了GigabitEthernet1/0/1） **注释：** （可选）用户可以定义多个接合点，并且这个捕获点的所有参数可以用一个命令实例进行配置。这些参数会在修改捕获点参数中进行介绍。在添加和删除接合点时，也可以使用范围进行配置。 用户可以在 *interface-type* 部分配置以下内容： · **GigabitEthernet**——把接合点指定为千兆以太网接口 · **vlan**——把接合点指定为VLAN **注释：** 只有在入向捕获实例（in）中，才能把 VLAN 指定为接合点。 · **capwap**——把接合点指定为CAPWAP 隧道 **注释：** 在使用这种接口作为接合点时，不能使用核心过滤器。 • （可选）**control-plane**——把控制平面指定为接合点 • **in** | **out** | **both**——指定捕获方向

步骤 4	**monitor capture** {*capture-name*}[**match** {**any** | **ipv4 any any** | **ipv6**} **any any**] **示例：** Device# **monitor capture mycap interface GigabitEthernet1/0/1 in match any**	定义核心系统过滤器。 **注释：** 在使用 CAPWAP 隧道接口作为接合点时，不要执行这个配置步骤，因为这时不能使用核心过滤器。 命令中的关键字含义如下所示： • *capture-name*——指定用户要定义的捕获点名称（示例中使用了 mycap） • **match**——指定一个过滤器。定义的第一个过滤器是核心过滤器 **注释：** 如果捕获点上没有配置核心系统过滤器，也没有定义接合点的话，用户就不能激活这个捕获点。如果用户尝试激活不满足这些需求的捕获点，就会看到一个错误信息。 • **ipv4**——指定一个 IPv4 过滤器 • **ipv6**——指定一个 IPv6 过滤器
步骤 5	**show monitor capture** {*capture-name*}[**parameter**] **示例：** Device# **show monitor capture mycap** parameter monitor capture mycap interface GigabitEthernet1/0/1 in monitor capture mycap match any	显示用户在步骤 2 中定义的捕获点参数，并确认用户定义的捕获点配置无误
步骤 6	**show running-config**	检查用户输入的信息

	示例： Device# **show running-config**	
步骤 7	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

用户可以为捕获点定义一个 CAPWAP 接合点：

```
Device# show capwap summary
CAPWAP Tunnels General Statistics:
Number of Capwap Data Tunnels = 1
Number of Capwap Mobility Tunnels = 0
Number of Capwap Multicast Tunnels = 0
Name APName Type PhyPortIf Mode McastIf
------ -------------------------------- ---- --------- ---------
-------
Ca0 AP442b.03a9.6715 data Gi3/0/6 unicast -
Name SrcIP SrcPort DestIP DstPort DtlsEn MTU Xact
------ --------------- ------- --------------- ------- ------ -----
----
Ca0 10.10.14.32 5247 10.10.14.2 38514 No 1449 0
Device# monitor capture mycap interface capwap 0 both
Device# monitor capture mycap file location flash:mycap.pcap
Device# monitor capture mycap file buffer-size 1
Device# monitor capture mycap start
*Aug 20 11:02:21.983: %BUFCAP-6-ENABLE: Capture Point mycap
enabled.on
Device# show monitor capture mycap parameter
monitor capture mycap interface capwap 0 in
monitor capture mycap interface capwap 0 out
monitor capture mycap file location flash:mycap.pcap buffer-size 1
Device#
Device# show monitor capture mycap
```

```
Status Information for Capture mycap
Target Type:
Interface: CAPWAP,
Ingress:
0
Egress:
0
Status : Active
Filter Details:
Capture all packets
Buffer Details:
Buffer Type: LINEAR (default)
File Details:
Associated file name: flash:mycap.pcap
Size of buffer(in MB): 1
Limit Details:
Number of Packets to capture: 0 (no limit)
Packet Capture duration: 0 (no limit)
Packet Size to capture: 0 (no limit)
Packets per second: 0 (no limit)
Packet sampling rate: 0 (no sampling)
Device#
Device# show monitor capture file flash:mycap.pcap
1 0.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe
Request, SN=0, FN=0,
Flags=........
2 0.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe
Request, SN=0, FN=0,
Flags=........
3 2.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe
Request, SN=0, FN=0,
Flags=........
4 2.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe
```

Request, SN=0, FN=0,

Flags=........

5 3.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe

Request, SN=0, FN=0,

Flags=........

6 4.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe

Request, SN=0, FN=0,

Flags=........

7 4.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe

Request, SN=0, FN=0,

Flags=........

8 5.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe

Request, SN=0, FN=0,

Flags=........

9 5.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe

Request, SN=0, FN=0,

Flags=........

10 6.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe

Request, SN=0, FN=0,

Flags=........

11 8.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe

Request, SN=0, FN=0,

Flags=........

12 9.225986 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data

13 9.225986 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data

14 9.225986 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data

15 9.231998 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data

16 9.231998 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data

17 9.231998 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data

18 9.236987 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data

19 10.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe

Request, SN=0, FN=0,

Flags=........

```
20 10.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe
Request, SN=0, FN=0,
Flags=........
21 12.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe
Request, SN=0, FN=0,
Flags=........
22 12.239993 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
23 12.244997 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
24 12.244997 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
25 12.250994 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
26 12.256990 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
27 12.262987 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
28 12.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe
Request, SN=0, FN=0,
Flags=........
29 12.802012 10.10.14.3 -> 10.10.14.255 NBNS Name query NB WPAD.<00>
30 13.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe
Request, SN=0, FN=0,
Flags=........
```

接下来做什么？

用户可以添加其他接合点、修改捕获点的参数，然后激活捕获点；或者如果用户希望就这样使用这个捕获点，可以现在就激活它。

注释： 用户不能使用这一部分提供的方法来修改该捕获点的参数。

如果用户输入了错误的捕获明晨个，或者一个无效/不存在的接合点，交换机会显示出类似这样的错误信息“*Capture Name should be less than or equal to 8 characters. Only alphanumeric characters and underscore (_) is permitted" and "% Invalid input detected at '^' marker*”。

添加或修改捕获点参数

虽然这些配置是按顺序介绍的，但指定参数值的这些命令可以以任何顺序进行配置。用户也可以在一条命令、两条命令或者多条命令中指定这些参数。除了接合点可以有多个之外，用户会在配置其他参数的取值时，替换掉这个参数已有的值。用户需要在对已有参数进行修改时再次进行确认。

用户可以按照以下步骤来修改捕获点的参数。

在开始前

用户必须先定义捕获点，才能按照以下步骤对其进行修改。

总步骤

1. enable

2. monitor capture {*capture-name*} **match** {*any* | **mac** *mac-match-string* | **ipv4** {**any** | **host** | **protocol**}{**any** | **host**} | **ipv6** {**any** | **host** | **protocol**}{**any** | **host**}}

3. monitor capture {*capture-name*} **limit** {[**duration** *seconds*][**packet-length** *size*][**packets** *num*]}

4. monitor capture {*capture-name*} **file** {**location filename**}

5. monitor capture {*capture-name*} **file** {**buffer-size** *size*}

6. show monitor capture {*capture-name*}[**parameter**]

7. end

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**monitor capture** {*capture-name*} **match** {*any* | **mac** *mac-match-string* | **ipv4** {**any** | **host** | **protocol**}{**any** | **host**} | **ipv6** {**any** | **host** | **protocol**}{**any** | **host**}} **示例：** `Device# monitor capture mycap match ipv4 any any`	定义核心系统过滤器（**ipv4 any any**），明确指定或通过 ACL 或 class-map 进行指定 ~~**注释：** 如果用户使用 CAPWAP 隧道接口定义无线捕获点，这条命令就不会生效，用户不应该使用这条命令~~
步骤 3	**monitor capture** {*capture-name*} **limit** {[**duration** *seconds*][**packet-length** *size*][**packets** *num*]} **示例：** `Device# monitor capture mycap`	以秒为单位指定会话限制（60）、不活的数据包，或 Wireshark 保留的数据包长度（400）

	limit duration 60 packet-len 400	
步骤 4	**monitor capture** {*capture-name*} **file** {**location filename**} **示例：** Device# **monitor capture mycap file location flash:mycap.pcap**	指定文件关联信息，如果捕获点希望捕获数据包，而不是仅仅显示数据包的话。 **注释：** 如果文件已经存在，用户必须确认是否可以覆盖原文件 **注释：** LAN Base 镜像版本中不没有文件选项
步骤 5	**monitor capture** {*capture-name*} **file** {**buffer-size** *size*} **示例：** Device# **monitor capture mycap file buffer-size 100**	指定 Wireshark 能够用来处理流量突发所使用的内存大小
步骤 6	**show monitor capture** {*capture-name*}[**parameter**] **示例：** Device# **show monitor capture mycap parameter** monitor capture mycap interface GigabitEthernet1/0/1 in monitor capture mycap match ipv4 any any monitor capture mycap limit duration 60 packet-len 400 monitor capture point mycap file location bootdisk:mycap.pcap	显示用户之前定义的捕获点参数

	monitor capture mycap file buffer-size 100	
步骤 7	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式

示例

修改参数

捕获文件的关联或解除关联

```
Device# monitor capture point mycap file location flash:mycap.pcap
Device# no monitor capture mycap file
```

为处理数据包突发指定内存缓存大小

```
Device# monitor capture mycap buffer size 100
```

明确定义核心系统过滤器来匹配 IPv4 和 IPv6

```
Device# monitor capture mycap match any
```

接下来做什么？

如果捕获点中包含了用户希望使用的所有参数，现在可以激活它。

删除捕获点参数

虽然这些配置是按顺序介绍的，但删除参数的这些命令可以以任何顺序进行配置。用户也可以在一条命令、两条命令或者多条命令中删除这些参数。除了接合点可以有多个之外，用户可以删除任意参数。

用户可以按照以下步骤来删除捕获点的参数。

在开始前

用户必须先定义捕获点，才能按照以下步骤对其进行删除。

总步骤

1. enable

2. no monitor capture {*capture-name*} **match**

3. no monitor capture {*capture-name*} **limit** [**duration**][**packet-length**][**packets**]

4. no monitor capture {*capture-name*} **file** [**location**] [**buffer-size**]

5. show monitor capture {*capture-name*}[**parameter**]

6. end

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**no monitor capture** {*capture-name*} **match** **示例：** `Device# no monitor capture mycap match`	删除捕获点（mycap）上定义的所有过滤器

步骤 3	**no monitor capture** {*capture-name*} **limit** [**duration**][**packet-length**][**packets**]
	
	示例：
	`Device# no monitor capture mycap limit duration packet-len`
	`Device# no monitor capture mycap limit`
		删除会话的时间限制和 Wireshark 保存的数据包长度限制。用户指定的其他限制不受影响。
		删除 Wireshark 上的所有限制

步骤 4	**no monitor capture** {*capture-name*} **file** [**location**] [**buffer-size**]
	
	示例：
	`Device# no monitor capture mycap file`
	`Device# no monitor capture mycap file location`
		删除关联的文件。捕获点将不会再捕获数据包。它只会显示数据包。
		删除文件位置的关联信息。文件位置不再与这个捕获点相关联。但用户定义的其他文件关联信息不会受到影响

步骤 5	**show monitor capture** {*capture-name*}[**parameter**]	显示用户删除参数后，还剩下的捕获点参数。用户可以在删除过程中随时使用这条命令，来查看这个捕获点上还关联

	示例：
	`Device# show monitor capture mycap parameter`
	`monitor capture mycap interface GigabitEthernet1/0/1 in`
		了什么参数

步骤 6	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式

接下来做什么？

如果捕获点中包含了用户希望使用的所有参数，现在可以激活它。

注释： 如果用户在捕获点为活跃状态时删除了参数，交换机就会显示出一个错误信息“*Capture is active*”。

删除捕获点

用户可以按照以下步骤来删除捕获点。

在开始前

用户必须先定义捕获点，才能按照以下步骤对其进行删除。

总步骤

1. enable

2. no monitor capture {*capture-name*}

3. show monitor capture {*capture-name*}[**parameter**]

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：**	进入特权 EXEC 模式。在提示时输入密码

	Device> **enable**	
步骤 2	**no monitor capture** {*capture-name*} **示例：** Device# **no monitor capture mycap**	删除指定的捕获点（mycap）
步骤 3	**show monitor capture** {*capture-name*}[**parameter**] **示例：** Device# **show monitor capture mycap parameter** Capture mycap does not exist	这条命令会使交换机显示出一条信息，指出该捕获点不存在，因为已经被删除
步骤 4	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** Device# **show running-config**	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

接下来做什么？

用户可以使用与删除的捕获点相同的名称，来定义一个新的捕获点。通常用户希望重新定义一个捕获点时，会使用这个步骤。

激活和停用捕获点

用户可以按照以下步骤来激活或停用捕获点。

在开始前

只有当捕获点上定义了接合点和核心系统过滤器，并且关联的文件名已存在时，用户才可以激活这个捕获点。在这种情况中，已有的文件会被覆盖。

没有关联文件名的捕获点在激活后只能用于显示。在没有指定文件名时，捕获的数据包储存在缓存中。文件模式和缓存模式都可以执行实时显示（一边捕获一边显示）。

如果没有指定显示过滤器，数据包就无法实时显示，所有由核心系统过滤器捕获的数据包能够显示。默认显示模式为简要信息。

注释： 在用户使用 CAPWAP 隧道接口作为接合点时，不能使用核心过滤器，因此这时不用定义核心过滤器。

总步骤

1. enable

2. monitor capture {*capture-name*} **start** [**display** [**display-filter** *filter-string*]][**brief** | **detailed** | **dump**]

3. monitor capture {*capture-name*} **stop**

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**monitor capture** {*capture-name*} **start** [**display** [**display-filter** *filter-string*]][**brief**	**detailed**	**dump**] **示例：** `Device# monitor capture mycap start display display-filter "stp"`	激活捕获点并过滤显示信息，只有包含“stp”的数据包会显示出来
步骤 3	**monitor capture** {*capture-name*} **stop**	停用捕获点

	示例： Device# **monitor capture name stop**	
步骤 4	**end** 示例： Device(config)# **end**	返回特权 EXEC 模式
步骤 5	**show running-config** 示例： Device# **show running-config**	检查用户输入的信息
步骤 6	**copy running-config startup-config** 示例： Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

接下来做什么？

在激活和停用捕获点时，用户可能会遇到一些错误。下面展示几个有可能出现的错误示例。

在激活时缺少接合点

Switch#**monitor capture mycap match any**

Switch#**monitor capture mycap start**

No Target is attached to capture failed to disable provision featurefailed to remove

policyfailed to disable provision featurefailed to remove policyfailed to disable provision

featurefailed to remove policy

Capture statistics collected at software (Buffer):

Capture duration - 0 seconds

Packets received - 0

Packets dropped - 0

Packets oversized - 0

Unable to activate Capture.

Switch# unable to get action unable to get action unable to get action

Switch#**monitor capture mycap interface g1/0/1 both**

Switch#**monitor capture mycap start**

Switch#

*Nov 5 12:33:43.906: %BUFCAP-6-ENABLE: Capture Point mycap enabled.

激活时缺少过滤器

Switch#**monitor capture mycap int g1/0/1 both**

Switch#**monitor capture mycap start**

Filter not attached to capture

Capture statistics collected at software (Buffer):

Capture duration - 0 seconds

Packets received - 0

Packets dropped - 0

Packets oversized - 0

Unable to activate Capture.

Switch#**monitor capture mycap match any**

Switch#**monitor capture mycap start**

Switch#

*Nov 5 12:35:37.200: %BUFCAP-6-ENABLE: Capture Point mycap enabled.

在用户尝试激活捕获点时，另一个捕获点已经处于活跃状态

Switch#**monitor capture mycap start**

PD start invoked while previous run is active Failed to start capture :

Wireshark operation

failure

Unable to activate Capture.

Switch#**show monitor capture**

Status Information for Capture test

Target Type:

Interface: GigabitEthernet1/0/13, Direction: both

Interface: GigabitEthernet1/0/14, Direction: both

Status : Active

Filter Details:

Capture all packets

```
Buffer Details:
Buffer Type: LINEAR (default)
Buffer Size (in MB): 10
File Details:
Associated file name: flash:cchh.pcap
Limit Details:
Number of Packets to capture: 0 (no limit)
Packet Capture duration: 0 (no limit)
Packet Size to capture: 0 (no limit)
Maximum number of packets to capture per second: 1000
Packet sampling rate: 0 (no sampling)
Status Information for Capture mycap
Target Type:
Interface: GigabitEthernet1/0/1, Direction: both
Status : Inactive
Filter Details:
Capture all packets
Buffer Details:
Buffer Type: LINEAR (default)
Buffer Size (in MB): 10
File Details:
File not associated
Limit Details:
Number of Packets to capture: 0 (no limit)
Packet Capture duration: 0 (no limit)
Packet Size to capture: 0 (no limit)
Maximum number of packets to capture per second: 1000
Packet sampling rate: 0 (no sampling)
Switch#monitor capture test stop
Capture statistics collected at software (Buffer & Wireshark):
Capture duration - 157 seconds
Packets received - 0
Packets dropped - 0
```

```
Packets oversized - 0
Switch#
*Nov 5 13:18:17.406: %BUFCAP-6-DISABLE: Capture Point test disabled.
Switch#monitor capture mycap start
Switch#
*Nov 5 13:18:22.664: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
Switch#
```

清除捕获点缓存

用户可以按照以下步骤，来清除缓存内容，或把其内容保存到外部文件中。

注释： 如果有多个捕获实例在向缓存中储存数据包，用户要在开启新捕获实例之前清除缓存，以防止内存丢失。不要尝试清除活跃捕获点上的缓存。

注释： 只有 LAN Base 镜像支持在活跃的捕获点上清除缓存，因为这种操作只会清除缓存内容。在所有其他版本的镜像中，这种行为会删除缓存本身，因此用户不能在捕获实例为活跃状态时执行这个操作。

总步骤

1. enable

2. monitor capture {*capture-name*} [**clear** | **export** *filename*]

3. end

4. show running-config

5. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**monitor capture** {*capture-name*} [**clear** | **export** *filename*] **示例：**	**clear**——完全删除缓存。 **注释：** 当用户执行清除命令时， • 在 LAN Base 镜像中——该命令会清除缓存内容，

	Device# **monitor capture mycap clear**	而不会删除缓存本身 • 在其他版本镜像中——该命令会删除缓存本身 **export**——保存缓存中捕获的数据包，同时删除缓存
步骤 3	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 4	**show running-config** **示例：** Device# **show running-config**	检查用户输入的信息
步骤 5	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

示例：捕获点的缓存控制

把捕获的数据包导出到文件中

```
Device# monitor capture mycap export flash:mycap.pcap
Storage configured as File for this capture
```

清除捕获点的缓存

```
Device# monitor capture mycap clear
Capture configured with file options
```

接下来做什么？

注释： 如果用户在除 LAN Base 镜像版本之外的环境中尝试删除捕获点的缓存，交换机会弹出错误消息“*Failed to clear capture buffer : Capture Buffer BUSY*”。

如何实施嵌入数据包捕获

管理数据包的数据捕获

注释： 只有 LAN Base 镜像版本支持导出活跃的捕获点。在所有其他版本的镜像中，用户需要首先停止捕获进程，才能将其导出。

用户可以按照以下步骤，来管理缓存模式的数据包数据捕获实例：

总步骤

1. enable

2. monitor capture *capture-name* **access-list** *access-list-name*

3. monitor capture *capture-name* **limit duration** *seconds*

4. monitor capture *capture-name* **interface** *interface-name* **both**

5. monitor capture *capture-name* **buffer circular size** *bytes*

6. monitor capture *capture-name* **start**

7. monitor capture *capture-name* **stop**

8. monitor capture *capture-name* **export** *file-location/file-name*

9. end

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**monitor capture** *capture-name* **access-list** *access-list-name* **示例：** `Device# monitor capture mycap access-list v4acl`	配置监控器捕获实例并把一个访问列表指定为核心过滤器，来过滤捕获的数据包
步骤 3	**monitor capture** *capture-name* **limit duration** *seconds*	配置监控器捕获实例的限制

	示例： Device# **monitor capture mycap limit duration 1000**	
步骤 4	**monitor capture** *capture-name* **interface** *interface-name* **both** 示例： Device# **monitor capture mycap interface GigabitEthernet 0/0/1 both**	配置监控器捕获实例并指定接合点和数据包流的方向
步骤 5	**monitor capture** *capture-name* **buffer circular size** *bytes* 示例： Device# **monitor capture mycap buffer circular size 10**	为捕获的数据包数据配置缓存
步骤 6	**monitor capture** *capture-name* **start** 示例： Device# **monitor capture mycap start**	开启数据包捕获进程，在流量追踪点捕获数据包并将其保存到缓存中
步骤 7	**monitor capture** *capture-name* **stop** 示例： Device# **monitor capture mycap stop**	停止数据包捕获进程，停止在流量追踪点捕获数据包
步骤 8	**monitor capture** *capture-name* **export** *file-location/file-name* 示例： Device# **monitor capture mycap export**	导出捕获的数据用于分析

	tftp://10.1.88.9/mycap.pcap	
步骤 9	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式

接下来做什么？

注释： 如果用户在除 LAN Base 镜像版本之外的环境中尝试删除捕获点的缓存，交换机会弹出错误消息“*Failed to clear capture buffer : Capture Buffer BUSY*”。

监控和维护捕获的数据

用户可以按照以下步骤，监控和维护捕获到的数据包数据。这部分展示了捕获缓存和捕获点的详细信息。

总步骤

1. enable

2. show monitor capture *capture-buffer-name* **buffer dump**

3. show monitor capture *capture-buffer-name* **parameter**

4. debug epc capture-point

5. debug epc provision

6. end

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**show monitor capture** *capture-buffer-name* **buffer dump** **示例：** `Device# show monitor capture mycap buffer dump`	（可选）显示捕获到的数据包及其元数据的十六进制转储信息

步骤 3	**show monitor capture** *capture-buffer-name* **parameter** **示例：** `Device# show monitor capture mycap parameter`	（可选）显示用来定义这个捕获实例的命令列表
步骤 4	**debug epc capture-point** **示例：** `Device# debug epc capture-point`	（可选）启用数据包捕获点的调试
步骤 5	**debug epc provision** **示例：** `Device# debug epc provision`	（可选）启用数据包捕获部署的调试
步骤 6	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式

监控数据包捕获

示例：查看.pcap 文件的简要输出信息

用户可以通过以下命令查看.pcap 文件中的简要输出信息：

```
Device# show monitor capture file flash:mycap.pcap brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
1 0.000000000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x002e,
seq=0/0, ttl=254
2 0.000051000 10.10.10.1 -> 10.10.10.2 ICMP 114 Echo (ping) reply
```

id=0x002e,

seq=0/0, ttl=255 (request in 1)

3 0.000908000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,

seq=1/256, ttl=254

4 0.001782000 10.10.10.1 -> 10.10.10.2 ICMP 114 Echo (ping) reply id=0x002e,

seq=1/256, ttl=255 (request in 3)

5 0.002961000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,

seq=2/512, ttl=254

6 0.003676000 10.10.10.1 -> 10.10.10.2 ICMP 114 Echo (ping) reply id=0x002e,

seq=2/512, ttl=255 (request in 5)

7 0.004835000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,

seq=3/768, ttl=254

8 0.005579000 10.10.10.1 -> 10.10.10.2 ICMP 114 Echo (ping) reply id=0x002e,

seq=3/768, ttl=255 (request in 7)

9 0.006850000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,

seq=4/1024, ttl=254

10 0.007586000 10.10.10.1 -> 10.10.10.2 ICMP 114 Echo (ping) reply id=0x002e,

seq=4/1024, ttl=255 (request in 9)

11 0.008768000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,

seq=5/1280, ttl=254

12 0.009497000 10.10.10.1 -> 10.10.10.2 ICMP 114 Echo (ping) reply id=0x002e,

seq=5/1280, ttl=255 (request in 11)

13 0.010695000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request

id=0x002e,

seq=6/1536, ttl=254

14 0.011427000 10.10.10.1 -> 10.10.10.2 ICMP 114 Echo (ping) reply

id=0x002e,

seq=6/1536, ttl=255 (request in 13)

15 0.012728000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request

id=0x002e,

seq=7/1792, ttl=254

16 0.013458000 10.10.10.1 -> 10.10.10.2 ICMP 114 Echo (ping) reply

id=0x002e,

seq=7/1792, ttl=255 (request in 15)

17 0.014652000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request

id=0x002e,

seq=8/2048, ttl=254

18 0.015394000 10.10.10.1 -> 10.10.10.2 ICMP 114 Echo (ping) reply

id=0x002e,

seq=8/2048, ttl=255 (request in 17)

19 0.016682000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request

id=0x002e,

seq=9/2304, ttl=254

20 0.017439000 10.10.10.1 -> 10.10.10.2 ICMP 114 Echo (ping) reply

id=0x002e,

seq=9/2304, ttl=255 (request in 19)

21 0.018655000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request

id=0x002e,

seq=10/2560, ttl=254

22 0.019685000 10.10.10.1 -> 10.10.10.2 ICMP 114 Echo (ping) reply

id=0x002e,

seq=10/2560, ttl=255 (request in 21)

23 0.020575000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request

id=0x002e,

seq=11/2816, ttl=254

--More‹

示例：查看.pcap 文件的详细输出信息

用户可以通过以下命令查看.pcap 文件中的详细输出信息：

```
Device# show monitor capture file flash:mycap.pcap detailed
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
Frame 1: 114 bytes on wire (912 bits), 114 bytes captured (912 bits)
on interface 0
Interface id: 0
Encapsulation type: Ethernet (1)
Arrival Time: Nov 6, 2015 11:44:48.322497000 UTC
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1446810288.322497000 seconds
[Time delta from previous captured frame: 0.000000000 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 0.000000000 seconds]
Frame Number: 1
Frame Length: 114 bytes (912 bits)
Capture Length: 114 bytes (912 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ip:icmp:data]
Ethernet II, Src: Inspur_f3:63:46 (00:e1:6d:f3:63:46), Dst:
Inspur_31:f1:c6 (00:e1:6d:31:f1:c6)
Destination: Inspur_31:f1:c6 (00:e1:6d:31:f1:c6)
Address: Inspur_31:f1:c6 (00:e1:6d:31:f1:c6)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory
default)
.... ...0 .... .... .... .... = IG bit: Individual address (unicast)
Source: Inspur_f3:63:46 (00:e1:6d:f3:63:46)
Address: Inspur_f3:63:46 (00:e1:6d:f3:63:46)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory
default)
```

```
.... ...0 .... .... .... .... = IG bit: Individual address (unicast)
Type: IP (0x0800)
Internet Protocol Version 4, Src: 10.10.10.2 (10.10.10.2), Dst:
10.10.10.1 (10.10.10.1)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00:
Not-ECT (Not
ECN-Capable Transport))
0000 00.. = Differentiated Services Codepoint: Default (0x00)
.... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable
Transport)
(0x00)
Total Length: 100
Identification: 0x04ba (1210)
Flags: 0x00
0... .... = Reserved bit: Not set
.0.. .... = Don't fragment: Not set
..0. .... = More fragments: Not set
Fragment offset: 0
Time to live: 254
Protocol: ICMP (1)
Header checksum: 0x8fc8 [validation disabled]
[Good: False]
[Bad: False]
Source: 10.10.10.2 (10.10.10.2)
Destination: 10.10.10.1 (10.10.10.1)
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0xe4db [correct]
Identifier (BE): 46 (0x002e)
Identifier (LE): 11776 (0x2e00)
```

```
Sequence number (BE): 0 (0x0000)
Sequence number (LE): 0 (0x0000)
Data (72 bytes)
0000 00 00 00 00 09 c9 8f 77 ab cd ab cd ab cd ab cd .......w........
0010 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0020 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0030 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0040 ab cd ab cd ab cd ab cd ........
Data: 0000000009c98f77abcdabcdabcdabcdabcdabcdabcdabcd...
[Length: 72]
Frame 2: 114 bytes on wire (912 bits), 114 bytes captured (912 bits)
on interface 0
Interface id: 0
```

示例：查看.pcap 文件的数据包转储输出信息

用户可以通过以下命令来查看数据包转储信息：

```
Device# show monitor capture file flash:mycap.pcap dump
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
0000 00 e1 6d 31 f1 c6 00 e1 6d f3 63 46 08 00 45 00 ..m1....m.cF..E.
0010 00 64 04 ba 00 00 fe 01 8f c8 0a 0a 0a 02 0a 0a .d..............
0020 0a 01 08 00 e4 db 00 2e 00 00 00 00 00 00 09 c9 ................
0030 8f 77 ab cd ab cd ab cd ab cd ab cd ab cd ab cd .w..............
0040 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0050 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0060 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0070 ab cd ..
0000 00 e1 6d 31 f1 80 00 e1 6d 31 f1 80 08 00 45 00 ..m1....m1....E.
0010 00 64 04 ba 00 00 ff 01 8e c8 0a 0a 0a 01 0a 0a .d..............
0020 0a 02 00 00 ec db 00 2e 00 00 00 00 00 00 09 c9 ................
0030 8f 77 ab cd ab cd ab cd ab cd ab cd ab cd ab cd .w..............
0040 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0050 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
```

```
0060 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ..............
0070 ab cd ..
0000 00 e1 6d 31 f1 c6 00 e1 6d f3 63 46 08 00 45 00 ..m1....m.cF..E.
0010 00 64 04 bb 00 00 fe 01 8f c7 0a 0a 0a 02 0a 0a .d..............
0020 0a 01 08 00 e4 d7 00 2e 00 01 00 00 00 00 09 c9 ................
0030 8f 7a ab cd ab cd ab cd ab cd ab cd ab cd ab cd .z..............
0040 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
```

示例：使用显示过滤器查看.pcap 文件中的数据包

用户可以通过以下命令查看.pcap 文件中的数据包：

```
Device# show monitor capture file flash:mycap.pcap display-filter
"ip.src == 10.10.10.2" brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
1 0.000000000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x002e,
seq=0/0, ttl=254
3 0.000908000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x002e,
seq=1/256, ttl=254
5 0.002961000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x002e,
seq=2/512, ttl=254
7 0.004835000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x002e,
seq=3/768, ttl=254
9 0.006850000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x002e,
seq=4/1024, ttl=254
11 0.008768000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x002e,
seq=5/1280, ttl=254
13 0.010695000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
```

```
id=0x002e,
seq=6/1536, ttl=254
15 0.012728000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x002e,
seq=7/1792, ttl=254
17 0.014652000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x002e,
seq=8/2048, ttl=254
19 0.016682000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x002e,
seq=9/2304, ttl=254
21 0.018655000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x002e,
seq=10/2560, ttl=254
23 0.020575000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x002e,
seq=11/2816, ttl=254
```

示例：查看.pcap 文件中捕获的数据包数量

用户可以通过以下命令查看.pcap 文件中捕获的数据包数量：

```
Device# show monitor capture file flash:mycap.pcap packet-count
File name: /flash/mycap.pcap
Number of packets: 50
```

示例：查看.pcap 文件中单个数据包转储信息

用户可以通过以下命令来查看.pcap 文件中单个数据包的转储信息：

```
Device# show monitor capture file flash:mycap.pcap packet-number 10
dump
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
0000 00 e1 6d 31 f1 80 00 e1 6d 31 f1 80 08 00 45 00 ..m1....m1....E.
0010 00 64 04 be 00 00 ff 01 8e c4 0a 0a 0a 01 0a 0a .d..............
```

```
0020 0a 02 00 00 ec ce 00 2e 00 04 00 00 00 00 09 c9 ................
0030 8f 80 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0040 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0050 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0060 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0070 ab cd
```

示例：查看.pcap 文件中捕获的数据包状态统计信息

用户可以通过以下命令查看.pcap 文件中捕获的数据包状态统计信息：

```
Device# show monitor capture file flash:mycap.pcap statistics
"h225,counter"
================== H225 Message and Reason Counter ==================
RAS-Messages:
Call Signalling:
=====================================================================
====
```

示例：捕获和显示的简单示例

这个示例展示了如何在三层接口 GigabitEthernet 1/0/1 上监控流量：

步骤1. 使用以下命令定义捕获点并匹配相关流量：

```
Device# monitor capture mycap interface GigabitEthernet1/0/3 in
Device# monitor capture mycap match ipv4 any any
Device# monitor capture mycap limit duration 60 packets 50
Device# monitor capture mycap buffer size 100
```

要想避免产生高 CPU 利用率，用户设置了较少的数据包数量和捕获时长。

步骤2. 使用以下命令确认以正确定义了捕获点：

```
Device# show monitor capture mycap parameter
monitor capture mycap interface GigabitEthernet1/0/3 in
monitor capture mycap match ipv4 any any
monitor capture mycap buffer size 100
monitor capture mycap limit packets 50 duration 60
```

```
Device# show monitor capture mycap
Status Information for Capture mycap
Target Type:
Interface: GigabitEthernet1/0/3, Direction: in
Status : Inactive
Filter Details:
IPv4
Source IP: any
Destination IP: any
Protocol: any
Buffer Details:
Buffer Type: LINEAR (default)
Buffer Size (in MB): 100
File Details:
File not associated
Limit Details:
Number of Packets to capture: 50
Packet Capture duration: 60
Packet Size to capture: 0 (no limit)
Packet sampling rate: 0 (no sampling)
```

步骤3. 开启捕获进程并查看结果

```
Device# monitor capture mycap start display
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
1 0.000000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0030, seq=0/0,
ttl=254
2 0.003682 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0030,
seq=1/256, ttl=254
3 0.006586 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0030,
seq=2/512, ttl=254
4 0.008941 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
```

```
id=0x0030,
seq=3/768, ttl=254
5 0.011138 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0030,
seq=4/1024, ttl=254
6 0.014099 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0030,
seq=5/1280, ttl=254
7 0.016868 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0030,
seq=6/1536, ttl=254
8 0.019210 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0030,
seq=7/1792, ttl=254
9 0.024785 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0030,
seq=8/2048, ttl=254
--More--
```

步骤4. 使用以下命令来删除捕获点：

```
Device# no monitor capture mycap
```

注释： 在这个示例中不需要使用 **stop** 命令，因为示例中设置了捕获限制，因此一旦达到限制条件，捕获进程就会自动停止。

更多用来查看 pcap 状态统计信息的相关语法信息，用户可以查看“*其他参考资料*”部分。

示例：捕获和储存的简单示例

这个示例展示了如何把捕获的数据包储存到文件中：

步骤1. 使用以下命令来定义匹配相关流量的捕获点并把它关联到一个文件：

```
Device# monitor capture mycap interface GigabitEthernet1/0/3 in
Device# monitor capture mycap match ipv4 any any
Device# monitor capture mycap limit duration 60 packets 50
Device# monitor capture mycap file location flash:mycap.pcap
```

步骤2. 使用以下命令来确认已经正确定义了捕获点：

```
Device# show monitor capture mycap parameter
monitor capture mycap interface GigabitEthernet1/0/3 in
monitor capture mycap match ipv4 any any
monitor capture mycap file location flash:mycap.pcap
monitor capture mycap limit packets 50 duration 60
Device# show monitor capture mycap
Status Information for Capture mycap
Target Type:
Interface: GigabitEthernet1/0/3, Direction: in
Status : Inactive
Filter Details:
IPv4
Source IP: any
Destination IP: any
Protocol: any
Buffer Details:
Buffer Type: LINEAR (default)
File Details:
Associated file name: flash:mycap.pcap
Limit Details:
Number of Packets to capture: 50
Packet Capture duration: 60
Packet Size to capture: 0 (no limit)
Packet sampling rate: 0 (no sampling)
```

步骤3. 使用以下命令开启数据包捕获进程：

```
Device# monitor capture mycap start
```

步骤4. 使用以下命令在运行期间显示扩展的捕获状态统计信息：

```
Device# show monitor capture mycap capture-statistics
Capture statistics collected at software:
Capture duration - 15 seconds
Packets received - 40
Packets dropped - 0
Packets oversized - 0
```

```
Packets errored - 0
Packets sent - 40
Bytes received - 7280
Bytes dropped - 0
Bytes oversized - 0
Bytes errored - 0
Bytes sent ¬ 4560
```

步骤5. 使用以下命令在合适的时候停止捕获进程：

```
Device# monitor capture mycap stop
Capture statistics collected at software (Buffer & Wireshark):
Capture duration - 20 seconds
Packets received - 50
Packets dropped - 0
Packets oversized - 0
```

注释： 另外，用户还可以让捕获操作在一段时间后自动停止，或者在收集到一定数量的数据包后自动停止。

mycap.pcap 文件现在包含了捕获的数据包。

步骤6. 使用以下命令在停止捕获后查看扩展的捕获状态统计信息：

```
Device# show monitor capture mycap capture-statistics
Capture statistics collected at software:
Capture duration - 20 seconds
Packets received - 50
Packets dropped - 0
Packets oversized - 0
Packets errored - 0
Packets sent - 50
Bytes received - 8190
Bytes dropped - 0
Bytes oversized - 0
Bytes errored – 0
Bytes sent ¬ 5130
```

步骤7. 使用以下命令查看数据包：

```
Device# show monitor capture file flash:mycap.pcap
```

```
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
1 0.000000000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0031,
seq=0/0, ttl=254
2 0.002555000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0031,
seq=1/256, ttl=254
3 0.006199000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0031,
seq=2/512, ttl=254
4 0.009199000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0031,
seq=3/768, ttl=254
5 0.011647000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0031,
seq=4/1024, ttl=254
6 0.014168000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0031,
seq=5/1280, ttl=254
7 0.016737000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0031,
seq=6/1536, ttl=254
8 0.019403000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0031,
seq=7/1792, ttl=254
9 0.022151000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0031,
seq=8/2048, ttl=254
10 0.024722000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0031,
seq=9/2304, ttl=254
11 0.026890000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0031,
```

```
seq=10/2560, ttl=254
12 0.028862000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0031,
seq=11/2816, ttl=254
--More--
```

更多用来查看 pcap 状态统计信息的相关语法信息，用户可以查看“*其他参考资料*”部分。

步骤8. 使用以下命令来删除捕获点：

```
Device# no monitor capture mycap
```

示例：使用缓存的捕获实例

这个示例展示了如何使用缓存进行捕获：

步骤1. 使用以下命令开启设置了缓存捕获选项的捕获会话：

```
Device# monitor capture mycap interface GigabitEthernet1/0/3 in
Device# monitor capture mycap match ipv4 any any
Device# monitor capture mycap buffer circular size 1
Device# monitor capture mycap start
```

步骤2. 使用以下命令确定捕获会话是否活跃：

```
Device# show monitor capture mycap
Status Information for Capture mycap
Target Type:
Interface: GigabitEthernet1/0/3, Direction: in
Status : Active
Filter Details:
IPv4
Source IP: any
Destination IP: any
Protocol: any
Buffer Details:
Buffer Type: CIRCULAR
Buffer Size (in MB): 1
File Details:
File not associated
```

```
Limit Details:
Number of Packets to capture: 0 (no limit)
Packet Capture duration: 0 (no limit)
Packet Size to capture: 0 (no limit)
Maximum number of packets to capture per second: 1000
Packet sampling rate: 0 (no sampling)
```

步骤3. 使用以下命令在运行期间显示扩展的捕获状态统计信息：

```
Device# show monitor capture mycap capture-statistics
Capture statistics collected at software:
Capture duration - 88 seconds
Packets received - 1000
Packets dropped - 0
Packets oversized - 0
Packets errored - 0
Packets sent - 1000
Bytes received - 182000
Bytes dropped - 0
Bytes oversized - 0
Bytes errored - 0
Bytes sent - 114000
```

步骤4. 使用以下命令来停止捕获进程：

```
Device# monitor capture mycap stop
Capture statistics collected at software (Buffer):
Capture duration - 2185 seconds
Packets received - 51500
Packets dropped - 0
Packets oversized - 0
```

步骤5. 使用以下命令在停止捕获后查看扩展的捕获状态统计信息：

```
Device# show monitor capture mycap capture-statistics
Capture statistics collected at software:
Capture duration - 156 seconds
Packets received - 2000
Packets dropped - 0
```

```
Packets oversized - 0
Packets errored - 0
Packets sent - 2000
Bytes received - 364000
Bytes dropped - 0
Bytes oversized - 0
Bytes errored - 0
Bytes sent - 228000
```

步骤6. 使用以下命令来确定捕获进程是否活跃：

```
Device# show monitor capture mycap
Status Information for Capture mycap
Target Type:
Interface: GigabitEthernet1/0/3, Direction: in
Status : Inactive
Filter Details:
IPv4
Source IP: any
Destination IP: any
Protocol: any
Buffer Details:
Buffer Type: CIRCULAR
Buffer Size (in MB): 1
File Details:
File not associated
Limit Details:
Number of Packets to capture: 0 (no limit)
Packet Capture duration: 0 (no limit)
Packet Size to capture: 0 (no limit)
Maximum number of packets to capture per second: 1000
Packet sampling rate: 0 (no sampling)
```

步骤7. 使用以下命令查看缓存中的数据包：

```
Device# show monitor capture mycap buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
```

1 0.000000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0038,
seq=40057/31132, ttl=254
2 0.000030 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0038,
seq=40058/31388, ttl=254
3 0.000052 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0038,
seq=40059/31644, ttl=254
4 0.000073 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0038,
seq=40060/31900, ttl=254
5 0.000094 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0038,
seq=40061/32156, ttl=254
6 0.000115 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0038,
seq=40062/32412, ttl=254
7 0.000137 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0038,
seq=40063/32668, ttl=254
8 0.000158 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0038,
seq=40064/32924, ttl=254
9 0.000179 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0038,
seq=40065/33180, ttl=254
10 0.000200 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0038,
seq=40066/33436, ttl=254
11 0.000221 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0038,
seq=40067/33692, ttl=254

```
12 0.000243 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0038,
seq=40068/33948, ttl=254
--More--
```

注意这些数据包已经储存在缓存中了。

步骤8. 使用以下命令查看其他显示模式中的数据包：

```
Device# show monitor capture mycap buffer detailed
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
Frame 1: 114 bytes on wire (912 bits), 114 bytes captured (912 bits)
on interface 0
Interface id: 0
Encapsulation type: Ethernet (1)
Arrival Time: Nov 6, 2015 18:10:06.297972000 UTC
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1446833406.297972000 seconds
[Time delta from previous captured frame: 0.000000000 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 0.000000000 seconds]
Frame Number: 1
Frame Length: 114 bytes (912 bits)
Capture Length: 114 bytes (912 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ip:icmp:data]
Ethernet II, Src: Inspur_f3:63:46 (00:e1:6d:f3:63:46), Dst:
Inspur_31:f1:c6 (00:e1:6d:31:f1:c6)
Destination: Inspur_31:f1:c6 (00:e1:6d:31:f1:c6)
Address: Inspur_31:f1:c6 (00:e1:6d:31:f1:c6)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory
default)
.... ...0 .... .... .... .... = IG bit: Individual address (unicast)
Source: Inspur_f3:63:46 (00:e1:6d:f3:63:46)
Address: Inspur_f3:63:46 (00:e1:6d:f3:63:46)
```

.... ..0. .... = LG bit: Globally unique address (factory default)

....0 .... = IG bit: Individual address (unicast)

Type: IP (0x0800)

Internet Protocol Version 4, Src: 10.10.10.2 (10.10.10.2), Dst: 10.10.10.1 (10.10.10.1)

Version: 4

Header length: 20 bytes

Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not

ECN-Capable Transport))

0000 00.. = Differentiated Services Codepoint: Default (0x00)

.... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport)

(0x00)

Total Length: 100

Identification: 0xabdd (43997)

Flags: 0x00

0... .... = Reserved bit: Not set

.0.. .... = Don't fragment: Not set

..0. .... = More fragments: Not set

Fragment offset: 0

Time to live: 254

Protocol: ICMP (1)

Header checksum: 0xe8a4 [validation disabled]

[Good: False]

[Bad: False]

Source: 10.10.10.2 (10.10.10.2)

Destination: 10.10.10.1 (10.10.10.1)

Internet Control Message Protocol

Type: 8 (Echo (ping) request)

Code: 0

Checksum: 0xa620 [correct]

```
Identifier (BE): 56 (0x0038)
Identifier (LE): 14336 (0x3800)
Sequence number (BE): 40057 (0x9c79)
Sequence number (LE): 31132 (0x799c)
Data (72 bytes)
0000 00 00 00 00 0b 15 30 63 ab cd ab cd ab cd ab cd ......0c........
0010 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0020 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0030 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0040 ab cd ab cd ab cd ab cd ........
Data: 000000000b153063abcdabcdabcdabcdabcdabcdabcdabcd...
[Length: 72]
Frame 2: 114 bytes on wire (912 bits), 114 bytes captured (912 bits)
on interface 0
Device# show monitor capture mycap buffer dump
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
0000 00 e1 6d 31 f1 c6 00 e1 6d f3 63 46 08 00 45 00 ..m1....m.cF..E.
0010 00 64 ab dd 00 00 fe 01 e8 a4 0a 0a 0a 02 0a 0a .d..............
0020 0a 01 08 00 a6 20 00 38 9c 79 00 00 00 00 0b 15 ..... .8.y......
0030 30 63 ab cd ab cd ab cd ab cd ab cd ab cd 0c..............
0040 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0050 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0060 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0070 ab cd ..
0000 00 e1 6d 31 f1 c6 00 e1 6d f3 63 46 08 00 45 00 ..m1....m.cF..E.
0010 00 64 ab de 00 00 fe 01 e8 a3 0a 0a 0a 02 0a 0a .d..............
0020 0a 01 08 00 a6 1d 00 38 9c 7a 00 00 00 00 0b 15 .......8.z......
0030 30 65 ab cd ab cd ab cd ab cd ab cd ab cd 0e..............
0040 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0050 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0060 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd ................
0070 ab cd
```

步骤9. 使用以下命令来清除缓存：

```
Device# monitor capture mycap clear
```

注释： 清除缓存的操作会在清除缓存内容的同时删除缓存本身。

注释： 如果用户需要查看缓存内容，可以在 show 命令后运行清除命令。

步骤10. 使用以下命令来重启流量，等待 10 秒钟，然后查看缓存内容：

注释： 用户不能在捕获进程活跃期间查看缓存内容。用户应该在查看缓存内容前停止捕获进程。用户可以在捕获进程活跃期间，在 pcap 文件上查看文件模式和缓存模式的数据内容。在文件模式中，用户可以在捕获进程活跃期间，查看当前捕获进程 pcap 文件中的数据包。

```
Device# monitor capture mycap start
Switch# show monitor capture mycap
Status Information for Capture mycap
Target Type:
Interface: GigabitEthernet1/0/3, Direction: in
Status : Active
Filter Details:
IPv4
Source IP: any
Destination IP: any
Protocol: any
Buffer Details:
Buffer Type: CIRCULAR
Buffer Size (in MB): 1
File Details:
File not associated
Limit Details:
Number of Packets to capture: 0 (no limit)
Packet Capture duration: 0 (no limit)
Packet Size to capture: 0 (no limit)
Maximum number of packets to capture per second: 1000
Packet sampling rate: 0 (no sampling)
```

步骤11. 使用以下命令停止数据包捕获进程并查看缓存内容：

```
Device# monitor capture mycap stop
Capture statistics collected at software (Buffer):
Capture duration - 111 seconds
```

```
Packets received - 5000
Packets dropped - 0
Packets oversized - 0
```

步骤12. 使用以下命令来确定捕获进程是否为活跃状态：

```
Device# show monitor capture mycap
Status Information for Capture mycap
Target Type:
Interface: GigabitEthernet1/0/3, Direction: in
Status : Inactive
Filter Details:
IPv4
Source IP: any
Destination IP: any
Protocol: any
Buffer Details:
Buffer Type: CIRCULAR
Buffer Size (in MB): 1
File Details:
File not associated
Limit Details:
Number of Packets to capture: 0 (no limit)
Packet Capture duration: 0 (no limit)
Packet Size to capture: 0 (no limit)
Maximum number of packets to capture per second: 1000
Packet sampling rate: 0 (no sampling)
```

步骤13. 使用以下命令来查看缓存中的数据包：

```
Device# show monitor capture mycap buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
1 0.000000000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=0/0, ttl=254
2 0.000030000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
```

```
seq=1/256, ttl=254
3 0.000051000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=2/512, ttl=254
4 0.000072000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=3/768, ttl=254
5 0.000093000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=4/1024, ttl=254
6 0.000114000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=5/1280, ttl=254
7 0.000136000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=6/1536, ttl=254
8 0.000157000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=7/1792, ttl=254
9 0.000178000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=8/2048, ttl=254
10 0.000199000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=9/2304, ttl=254
11 0.000220000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=10/2560, ttl=254
12 0.000241000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=11/2816, ttl=254
--More‹
```

步骤14. 使用以命令把缓存内容储存到 mycap.pcap 文件中，文件位置为设备内部 Flash:储存

设备：

```
Device# monitor capture mycap export flash:mycap.pcap
Exported Successfully
```

注释： 当前的输出过程是这样的，当命令执行后，输出状态为“已开始”但在它向用户返回提示时，输出过程还没有完成。因此用户需要等到在控制台上看到 Wireshark 返回的消息后，才能查看文件中的数据包。

步骤15. 使用以下命令查看文件中捕获的数据包：

```
Device# show monitor capture file flash:mycap.pcap
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
1 0.000000000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=0/0, ttl=254
2 0.000030000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=1/256, ttl=254
3 0.000051000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=2/512, ttl=254
4 0.000072000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=3/768, ttl=254
5 0.000093000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=4/1024, ttl=254
6 0.000114000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=5/1280, ttl=254
7 0.000136000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=6/1536, ttl=254
8 0.000157000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=7/1792, ttl=254
```

```
9 0.000178000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=8/2048, ttl=254
10 0.000199000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=9/2304, ttl=254
11 0.000220000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=10/2560, ttl=254
12 0.000241000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request
id=0x0039,
seq=11/2816, ttl=254
--More--
```

步骤16. 使用以下命令删除捕获点：

```
Device# no monitor capture mycap
```

示例：在出方向上捕获和储存数据包的简单示例

这个示例展示了如何在出方向上捕获数据包：

步骤1. 使用以下命令定义匹配相关流量的捕获点并将其关联到一个文件：

```
Device# monitor capture mycap interface Gigabit 1/0/1 out match ipv4
any any
Device# monitor capture mycap limit duration 60 packets 100
Device# monitor capture mycap file location flash:mycap.pcap
buffer-size 90
```

步骤2. 使用以下命令确认已经正确定义了捕获点：

```
Device# show monitor capture mycap parameter
monitor capture mycap interface GigabitEthernet1/0/1 out
monitor capture mycap match ipv4 any any
monitor capture mycap file location flash:mycap.pcap buffer-size 90
monitor capture mycap limit packets 100 duration 60
Device# show monitor capture mycap
Status Information for Capture mycap
```

```
Target Type:
Interface: GigabitEthernet1/0/1, Direction: out
Status : Inactive
Filter Details:
IPv4
Source IP: any
Destination IP: any
Protocol: any
Buffer Details:
Buffer Type: LINEAR (default)
File Details:
Associated file name: flash:mycap.pcap
Size of buffer(in MB): 90
Limit Details:
Number of Packets to capture: 100
Packet Capture duration: 60
Packet Size to capture: 0 (no limit)
Packets per second: 0 (no limit)
Packet sampling rate: 0 (no sampling)
```

步骤3. 使用以下命令开启数据包捕获进程：

```
Device# monitor capture mycap start
A file by the same capture file name already exists, overwrite?[confirm]
Turning on lock-step mode
Device#
*Oct 14 09:35:32.661: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
```

注释： 用户可以设置让捕获进程在一段时间后自动停止，也可以在捕获到一定数量的数据包后自动停止。当用户在输出信息中看到以下消息时，表示捕获进程已经停止：

```
*Oct 14 09:36:34.632: %BUFCAP-6-DISABLE_ASYNC: Capture Point mycap disabled. Reason : Wireshark Session Ended
```

这时 mycap.pcap 文件中包含了捕获到的数据包。

步骤4. 使用以下命令来查看数据包：

```
Device# show monitor capture file flash:mycap.pcap
```

```
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
0.000000 10.1.1.30 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
1.000000 10.1.1.31 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
2.000000 10.1.1.32 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
3.000000 10.1.1.33 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
4.000000 10.1.1.34 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
5.000000 10.1.1.35 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
6.000000 10.1.1.36 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
7.000000 10.1.1.37 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
8.000000 10.1.1.38 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
9.000000 10.1.1.39 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
```

步骤5. 使用以下命令来删除捕获点：

```
Device# no monitor capture mycap
```

嵌入数据包捕获的配置示例

示例：管理数据包的数据捕获实例

这个示例展示了如何管理数据包的数据捕获实例：

```
Device> enable
Device# monitor capture mycap start
```

```
Device# monitor capture mycap access-list v4acl
Device# monitor capture mycap limit duration 1000
Device# monitor capture mycap interface GigabitEthernet 0/0/1 both
Device# monitor capture mycap buffer circular size 10
Device# monitor capture mycap start
Device# monitor capture mycap export tftp://10.1.88.9/mycap.pcap
Device# monitor capture mycap stop
Device# end
```

示例：监控和维护捕获的数据

这个示例展示了如何把数据包转储为 ASCII 格式：

```
Device# show monitor capture mycap buffer dump
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
0
0000: 01005E00 00020000 0C07AC1D 080045C0 ..^...........E.
0010: 00300000 00000111 CFDC091D 0002E000 .0..............
0020: 000207C1 07C1001C 802A0000 10030AFA .........*......
0030: 1D006369 73636F00 0000091D 0001 ..example.......
1
0000: 01005E00 0002001B 2BF69280 080046C0 ..^.....+.....F.
0010: 00200000 00000102 44170000 0000E000 . ......D.......
0020: 00019404 00001700 E8FF0000 0000 ..............
2
0000: 01005E00 0002001B 2BF68680 080045C0 ..^.....+.....E.
0010: 00300000 00000111 CFDB091D 0003E000 .0..............
0020: 000207C1 07C1001C 88B50000 08030A6E ...............n
0030: 1D006369 73636F00 0000091D 0001 ..example.......
3
0000: 01005E00 000A001C 0F2EDC00 080045C0 ..^...........E.
0010: 003C0000 00000258 CE7F091D 0004E000 .<......... X........
0020: 000A0205 F3000000 00000000 00000000 ................
0030: 00000000 00D10001 000C0100 01000000 ................
```

```
0040: 000F0004 00080501 0300
```

以下示例展示了如何查看为名为 mycap 的捕获实例配置的命令列表：

```
Device# show monitor capture mycap parameter
monitor capture mycap interface GigabitEthernet 1/0/1 both
monitor capture mycap match any
monitor capture mycap buffer size 10
monitor capture mycap limit pps 1000
```

以下示例展示了如何查看捕获点调试信息：

```
Device# debug epc capture-point
EPC capture point operations debugging is on
Device# monitor capture mycap start
*Jun 4 14:17:15.463: EPC CP: Starting the capture cap1
*Jun 4 14:17:15.463: EPC CP: (brief=3, detailed=4, dump=5) = 0
*Jun 4 14:17:15.463: EPC CP: final check before activation
*Jun 4 14:17:15.463: EPC CP: setting up c3pl infra
*Jun 4 14:17:15.463: EPC CP: Setup c3pl acl-class-policy
*Jun 4 14:17:15.463: EPC CP: Creating a class
*Jun 4 14:17:15.464: EPC CP: Creating a class : Successful
*Jun 4 14:17:15.464: EPC CP: class-map Created
*Jun 4 14:17:15.464: EPC CP: creating policy-name epc_policy_cap1
*Jun 4 14:17:15.464: EPC CP: Creating Policy epc_policy_cap1 of type
49 and client type 21
*Jun 4 14:17:15.464: EPC CP: Storing a Policy
*Jun 4 14:17:15.464: EPC CP: calling ppm_store_policy with epc_policy
*Jun 4 14:17:15.464: EPC CP: Creating Policy : Successful
*Jun 4 14:17:15.464: EPC CP: policy-map created
*Jun 4 14:17:15.464: EPC CP: creating filter for ANY
*Jun 4 14:17:15.464: EPC CP: Adding acl to class : Successful
*Jun 4 14:17:15.464: EPC CP: Setup c3pl class to policy
*Jun 4 14:17:15.464: EPC CP: Attaching Class to Policy
*Jun 4 14:17:15.464: EPC CP: Attaching epc_class_cap1 to
epc_policy_cap1
*Jun 4 14:17:15.464: EPC CP: Attaching Class to Policy : Successful
```

```
*Jun 4 14:17:15.464: EPC CP: setting up c3pl qos
*Jun 4 14:17:15.464: EPC CP: DBG> Set packet rate limit to 1000
*Jun 4 14:17:15.464: EPC CP: creating action for policy_map
epc_policy_cap1 class_map
epc_class_cap1
*Jun 4 14:17:15.464: EPC CP: DBG> Set packet rate limit to 1000
*Jun 4 14:17:15.464: EPC CP: Activating Interface
GigabitEthernet1/0/1 direction both
*Jun 4 14:17:15.464: EPC CP: Id attached 0
*Jun 4 14:17:15.464: EPC CP: inserting into active lists
*Jun 4 14:17:15.464: EPC CP: Id attached 0
*Jun 4 14:17:15.465: EPC CP: inserting into active lists
*Jun 4 14:17:15.465: EPC CP: Activating Vlan
*Jun 4 14:17:15.465: EPC CP: Deleting all temp interfaces
*Jun 4 14:17:15.465: %BUFCAP-6-ENABLE: Capture Point cap1 enabled.
*Jun 4 14:17:15.465: EPC CP: Active Capture 1
Device# monitor capture mycap1 stop
*Jun 4 14:17:31.963: EPC CP: Stopping the capture cap1
*Jun 4 14:17:31.963: EPC CP: Warning: unable to unbind capture cap1
*Jun 4 14:17:31.963: EPC CP: Deactivating policy-map
*Jun 4 14:17:31.963: EPC CP: Policy epc_policy_cap1
*Jun 4 14:17:31.964: EPC CP: Deactivating policy-map Successful
*Jun 4 14:17:31.964: EPC CP: removing povision feature
*Jun 4 14:17:31.964: EPC CP: Found action for policy-map
epc_policy_cap1 class-map
epc_class_cap1
*Jun 4 14:17:31.964: EPC CP: cleanning up c3pl infra
*Jun 4 14:17:31.964: EPC CP: Removing Class epc_class_cap1 from Policy
*Jun 4 14:17:31.964: EPC CP: Removing Class from epc_policy_cap1
*Jun 4 14:17:31.964: EPC CP: Successfully removed
*Jun 4 14:17:31.964: EPC CP: Removing acl mac from class
*Jun 4 14:17:31.964: EPC CP: Removing acl from class : Successful
*Jun 4 14:17:31.964: EPC CP: Removing all policies
```

```
*Jun 4 14:17:31.964: EPC CP: Removing Policy epc_policy_cap1
*Jun 4 14:17:31.964: EPC CP: Removing Policy : Successful
*Jun 4 14:17:31.964: EPC CP: Removing class epc_class_cap1
*Jun 4 14:17:31.965: EPC CP: Removing class : Successful
*Jun 4 14:17:31.965: %BUFCAP-6-DISABLE: Capture Point cap1 disabled.
*Jun 4 14:17:31.965: EPC CP: Active Capture 0
```

以下示例展示了如何查看嵌入数据包捕获（EPC）部署的调试信息：

```
Device# debug epc provision
EPC provisionioning debugging is on
Device# monitor capture mycap start
*Jun 4 14:17:54.991: EPC PROV: No action found for policy-map
epc_policy_cap1 class-map
epc_class_cap1
*Jun 4 14:17:54.991: EPC PROV:
*Jun 4 14:17:54.991: Attempting to install service policy
epc_policy_cap1
*Jun 4 14:17:54.992: EPC PROV: Attached service policy to epc idb
subblock
*Jun 4 14:17:54.992: EPC PROV: Successful. Create feature object
*Jun 4 14:17:54.992: EPC PROV:
*Jun 4 14:17:54.992: Attempting to install service policy
epc_policy_cap1
*Jun 4 14:17:54.992: EPC PROV: Successful. Create feature object
*Jun 4 14:17:54.992: %BUFCAP-6-ENABLE: Capture Point cap1 enabled.
Device# monitor capture mycap stop
*Jun 4 14:18:02.503: EPC PROV: Successful. Remove feature object
*Jun 4 14:18:02.504: EPC PROV: Successful. Remove feature object
*Jun 4 14:18:02.504: EPC PROV: Destroyed epc idb subblock
*Jun 4 14:18:02.504: EPC PROV: Found action for policy-map
epc_policy_cap1 class-map
epc_class_cap1
*Jun 4 14:18:02.504: EPC PROV: Deleting EPC action
*Jun 4 14:18:02.504: EPC PROV: Successful. CLASS_REMOVE, policy-map
```

```
epc_policy_cap1, class
epc_class_cap1
*Jun 4 14:18:02.504: %BUFCAP-6-DISABLE: Capture Point cap1 disabled.
```

其他参考资料

相关文档

相关主题	文档名称
显示过滤器	有关显示过滤器的语法可参考： 显示过滤器参考资料
Pcap 文件状态统计信息	有关 pcap 文件装填统计信息的语法可参考“-z”详细信息： Tshark 命令参考
Inspur 6850 交换机命令	

错误消息解码器

描述	链接
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

标准和 RFC

标准/RFC	标题
无	-

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical	http://www.icntnetworks.com

Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	

配置 Flexible NetFlow

配置 Flexible NetFlow 的先决条件

配置 Flexible NetFlow 有以下先决条件：

- 用户必须配置一个源接口。如果用户没有配置源接口的话，导出器会保持在禁用状态；
- 用户必须为每个流监控器配置有效的记录名称；
- 用户必须启用 IPv6 路由，把流记录导出到 IPv6 目的服务器；
- 用户必须为流导出器配置 IPFIX 导出协议，才能把流记录导出为 IPFIX 格式；
- 用户必须熟悉 Inspur INOS Flexible NetFlow 命令参考中，用来定义 Flexible NetFlow 命令中的重要字段：
  - **match datalink**——数据链路（第 2 层）字段
  - **match flow**——流识别字段
  - **match interface**——接口字段
  - **match ipv4**——IPv4 字段
  - **match ipv6**——IPv6 字段
  - **match transport**——传输层字段
- 用户需要熟悉 Inspur INOS Flexible NetFlow 命令参考中，用来定义 Flexible NetFlow 命令中的非重要字段：
  - **collect counter**——计数器字段
  - **collect flow**——流识别字段

- **collect interface**——接口字段
- **collect timestamp**——时间戳字段
- **collect transport**——传输层字段

IPv4 流量

- 网络设备上必须配置 IPv4 路由；
- 在想要启用 Flexible NetFlow 的设备和接口上，用户必须在该设备和该接口上启用以下特性之一：Inspur 快速转发特性，或者分布式 Inspur 快速转发特性。

IPv6 流量

- 网络设备上必须配置 IPv6 路由；
- 在想要启用 Flexible NetFlow 的设备和接口上，用户必须在该设备和该接口上启用以下特性之一：Inspur IPv6 快速转发特性，或者分布式 Inspur 快速转发特性。

配置 Flexible NetFlow 的限制条件

配置 Flexible NetFlow 有以下限制条件：

- 二层 PortChannel 接口上不支持 Flexible NetFlow，但二层 PortChannel 成员端口上可以支持 Flexible NetFlow；
- 三层 PortChannel 接口上不支持 Flexible NetFlow，但三层 PortChannel 成员端口上可以支持 Flexible NetFlow；
- 不支持传统 NetFlow（TNF）审计功能；
- 支持 Flexible NetFlow 版本 9 和版本 10 的导出格式。但如果用户没有配置导出协议的话，默认使用版本 9 的导出格式；
- 微流（Microflow）限速特性与 FNF 共享 NetFlow 硬件资源；
- 在一个接口的一个方向上，只支持一个流监控器；
- 支持二层、IPv4 和 IPv6 流量类型；但在指定方向和指定接口上，设备只能够针对其中一种类型的流量应用流监控器；
- 支持二层、VLAN 和三层接口，但设备不支持 SVI 和隧道（Tunnel）接口；
- 支持以下 NetFlow 表大小：

镜像级别	入向 NetFlow 表	出向 NetFlow 表
LAN Base	不支持	不支持
IP Base	8 K	16 K
IP Services	8 K	16 K

- 根据交换机类型，一台交换机上会有一个或两个转发 ASIC。上面这个表格中列出的容量是针对每个 ASIC 进行描述的；
- 交换机可以支持一个或两个 ASIC。每个 ASIC 中有 8K 入向条目和 16K 出向条目，但每个 TCAM 可以处理多达 6K 入向条目和 12K 出向条目；
- 每个 NetFlow 表都是单独的，不能将其合并。根据数据包是由哪个 ASIC 进行处理的，设备会在这个 ASIC 中的 NetFlow 表中创建流条目；
- NetFlow 的硬件实现能够支持 4 个硬件采样器。用户可以选择采样速率，范围是 1/2 至 1/1024。只支持随机采样模式；
- NetFlow 硬件在其内部使用散列表。硬件中可能会出现散列冲突。因此不考虑内部溢出内容可寻址存储器（CAM），实际的 NetFlow 表利用率是 80%左右；
- 根据用户为流使用的字段，一条流可以占用两个连续的条目。IPv6 流也会占用两个条目。在这种情况中，NetFlow 条目的有效使用情况是表大小的一半，这一点与上述散列冲突相分离；
- 设备最多支持 63 个流监控器；
- 支持基于 SSID 的 NetFlow 审计功能。SSID 会被当作一个接口。但不支持特定的字段，比如用户 ID；
- NetFlow 软件的部署支持分布式 NetFlow 导出，也就是从创建流的设备中把流导出；
- 首先接收到流中第一个数据包的 ASIC，会为这个流创建入向流。数据包真正离开设备时使用的 ASIC，会为这个流创建出向流；
- 字节计数字段（称为“字节长度”）中报告的数值是二层数据包大小——18 字节。对于经典以太网流量（802.3）来说，这是很精确的。对于其他以太网类型来说，这个字段就不精确了。用户可以使用“第 2 层字节（bytes layer2）”字段，它总是报告精确的二层数据包大小。有关设备能够支持的更多 Flexible NetFlow 字段，用户可以参考支持的 Flexible NetFlow 字段；
- 不支持在 AVC 流监控器上配置 IPFIX 导出器；
- 以太网管理端口（Gi0/0）上不支持 Flexible NetFlow 导出；
- 如果流记录汇总过只有源组标记（SGT）和目的组标记（DGT）字段（或者只有这两个字段之一），并且如果它们的值不可应用的话，设备还是会创建流，并在 SGT 和 DGT 字段中设置 0 值。流记录中应该会有源和目的 IP 地址，以及 SGT 和 DGT 字段。

Flexible NetFlow 的相关信息

Flexible NetFlow 概述

Flexible NetFlow 使用流来为审计、网络监控和网络规划提供状态统计信息。

流（Flow）是指到达源接口的一个单向数据包流，并且这个数据包流拥有相同的键值。键（Key）是数据包中字段的标识值。用户可以使用流记录来创建一个流，并为其定义唯一的键值。

设备能够支持 Flexible NetFlow 特性，从而启用高级网络异常和安全检测。用户通过使用 Flexible NetFlow 特性，能够从众多预定义的字段中，为指定的应用程序选择键值，来为其定义最优流记录。

指定流中所有数据包的键值必须相同。根据用户配置的导出记录版本的不同，一个流中可能会汇集其他感兴趣字段。流储存在Flexible NetFlow缓存中。

用户可以使用导出器，导出Flexible NetFlow为用户的流收集到的数据，并把这些数据导出到远端系统中，比如Flexible NetFlow收集器。Flexible NetFlow收集器可以使用IPv4地址或IPv6地址。

用户可以使用监控器，来指定希望针对一个流收集的数据大小。监控器会把流记录汇集在一起，与Flexible NetFlow缓存信息一起导出。

最初的 NetFlow 和 Flexible NetFlow 的优势

最初的NetFlow使用IP信息中的固定7个字段来识别一个流。

Flexible NetFlow能够让用户自定义流。使用Flexible NetFlow的优势包括：

- 高容量的流识别功能，其中包括可扩展性和流信息的汇聚；
- 增强的流架构，用来实现安全监控，以及dDoS检测和识别；
- 使用数据包中的新信息，来匹配网络中特定服务或操作生成的流信息。Flexible NetFlow用户能够自定义可用的流信息；
- 支持Flexible NetFlow版本9和版本10的导出格式。~~在使用版本10的导出格式时，支持为无线客户端SSID使用可变长字段~~；
- 全面的IP审计特性能够代替很多审计特性，比如IP审计、边界网关协议（BGP）策略审计和持续性缓存；
- 支持入向和出向NetFlow审计；

- 支持全部流审计和抽样NetFlow审计。

最初的NetFlow能够让用户理解网络中的行为，从而优化网络设计并减少运营成本。

Flexible NetFlow能够让用户更有效地理解网络行为，它能够为网络中使用的各种服务提供量身定制的流信息。比如Flexible NetFlow特性能够用于以下应用：

- Flexible NetFlow把Inspur NetFlow升级为安全监控工具。举例来说，用户可以把数据包长度或MAC地址定义为新的键值，这样可以在网络中搜索特定类型的攻击；
- Flexible NetFlow能够通过数据包中的服务类别（CoS）字段，追踪特定的TCP或UDP应用，以此来快速识别主机之间发送了多少这种应用的流量；
- 对进入多协议标签交换（MPLS）网络或IP核心网络的流量进行审计，记录每一跳的服务类别。这一功能可以使用户构建出边界到边界的流量模型。

下图展示了Flexible NetFlow在网络中的部署示例。

图81：Flexible NetFlow的常见部署

Peering Flows Dest. AS Dest. Traffic Index BGP Next Hop DSCP	**对等体流** 目的AS 目的流量索引 BGP下一跳 DSCP
Branch	分支机构
Data Center	数据中心
Campus	园区网
Multicast Flows Protocol Ports IP Address TCP Flags Packet Section	**组播流** 协议 端口 IP地址 TCP标记 数据包选择
Security Flows Protocol Ports IP Address TCP Flags Packet Section	**安全流** 协议 端口 IP地址 TCP标记 数据包选择

IP Flows	**IP流**
IP Subnets	IP子网
Ports	端口
Protocol	协议
Interfaces	接口
Egress/Ingress	出向/入向

Flexible NetFlow 的组成部分

Flexible NetFlow的组成部分能够以不同的方式组合在一起使用，来执行流量分析和数据导出。用户定义的流记录和Flexible NetFlow架构的组成部分，为用户在网络设备上创建各种流量分析和数据导出配置提供了便利，把所需的配置命令数量降到最低。每个流监控器都有唯一的参数组成结构：流记录、流导出器和缓存类型。如果用户更改了参数，比如为流导出器更改了目的IP地址，Flexible NetFlow能够自动为所有使用这个流导出器的流监控器自动更新配置。一个流监控器可以与多个流采样器结合使用，在不同的接口上，以不同的速率采样相同类型的网络流量。接下来详细介绍Flexible NetFlow的组成部分：

注释： 从Inspur INOS 11.3.1版本开始，可配置的流记录字段从32增加到了40。

流记录

在 Flexible NetFlow 中，由多个键值（key）和非键值（nonkey）字段构成的组合称为一个记录。Flexible NetFlow 记录会被分配给 Flexible NetFlow 流监控器，用来定义流监控器储存流数据所用的缓存。Flexible NetFlow 中包含几个预定义的记录，这些记录能够帮助用户上手 Flexible NetFlow 特性。

一个流记录中定义了 Flexible NetFlow 用来识别流中数据包的键值，以及 Flexible NetFlow 为这个流收集的其他感兴趣字段。用户可以使用键值和感兴趣字段的任意组合方式，来定义一个流记录。设备支持大量键值。流记录中还能定义为每个流收集的计数器类型。用户可以配置 64 比特数据包或字节计数器。在用户创建流记录时，设备默认会匹配以下字段：

- match datalink——二层属性
- match flow direction——指定标识了流量方向的字段
- match interface——接口属性
- match ipv4——IPv4 属性
- match ipv6——IPv6 属性

- match transport——传输层字段

NetFlow 预定义记录

Flexible NetFlow 中包含一些预定义的记录，用户可以在网络中使用这些记录来开始监控流量。预定义的记录可以帮助用户快速部署 Flexible NetFlow，并且比用户定义的流记录更容易使用。用户可以从预定义记录列表中进行选择，这些记录可能会满足网络监控的需求。随着 Flexible NetFlow 的发展，用户可以和使用预定义记录一样，使用广受欢迎的用户自定义流记录，使部署便得更简单。

预定义记录能够向后兼容用户已有的用来导出数据的 NetFlow 收集器配置。每个预定义记录都有一组唯一的键值和非键值字段组合，用户可以直接使用内建的监控器，来监控网络中的各种流量类型，而无需在路由器中自定义 Flexible NetFlow。

有两个预定义记录（NetFlow original 和 NetFlow IPv4/IPv6 original output）的功能相同，在最初的 NetFlow 中分别模拟原始（入向）NetFlow 和出向 NetFlow 审计特性。一些其他的 Flexible NetFlow 预定义记录都基于最初的 NetFlow 中使用的汇聚缓存机制。基于汇聚缓存机制的 Flexible NetFlow 预定义记录在最初的 NetFlow 中不执行汇聚功能。而是每个流分别由预定义记录进行追踪。

用户定义的记录

Flexible NetFlow 允许用户根据自己的需求，指定键值和非键值来个性化数据收集规则，为 Flexible NetFlow 流监控器缓存定义自己的记录。用户在为 Flexible NetFlow 流监控器缓存定义自己的记录时，这些记录称为*用户定义的记录*。在流中添加的非键值字段中的值提供了流中的其他信息。用户更改非键值字段的取值并不会创建出新的流。在大多数情况下，非键值字段的取值只是来自于流中的第一个数据包。Flexible NetFlow 使用户能够捕获计数器值，比如流中的字节数和数据包数。

用户可以为应用创建用户定义的记录，比如 QoS 和带宽监控、应用和终端用户流量分析描述、以及用于 dDoS 攻击的安全监控。Flexible NetFlow 中还包含多个预定义的记录，模拟了最初的 NetFlow。Flexible NetFlow 中用户定义的记录能够根据用户配置的大小，监控数据包中连续的段落（Section），并在一个流中将这些信息用作键值或非键值字段，与数据包的其他字段和属性一起使用。数据包中的段落可以包含数据包中的任何三层数据。数据包段落字段也使用户能够监控并没有涵盖在 Flexible NetFlow 预定义键值中的任意数据包字段。对没有收集在预定义键值中的数据包字段执行分析，这一功能提供了更详细的流量监控行为，简化了对 dDoS 攻击的检测，也使用户能够实施其他安全应用，比如 URL 监控。

Flexible NetFlow 中预定义了各种类型的数据包段落字段，并且用户可以配置其大小。用户可以使用以下 Flexible NetFlow 命令（在 Flexible NetFlow 流记录配置模式中使用），来配置预定义数据包类型的段落字段大小：

- **collect ipv4 section header size** *bytes*——捕获从每个数据包 IPv4 头部开始算起，*bytes* 变量中指定的字节数；
- **collect ipv4 section payload size** *bytes*——捕获从每个数据包 IPv4 头部之后，*bytes* 变量中指定的字节数；
- **collect ipv6 section header size** *bytes*——捕获从每个数据包 IPv6 头部开始算起，*bytes* 变量中指定的字节数；
- **collect ipv6 section payload size** *bytes*——捕获从每个数据包 IPv6 头部之后，*bytes* 变量中指定的字节数。

bytes 值是流记录中这些字段的字节数。如果数据包中相应的段落小于用户指定的大小，Flexible NetFlow 会在流记录中这一段落的剩余字节中填补上 0。如果数据包类型不匹配指定的段落类型，Flexible NetFlow 会在流记录中的这一段落填补上 0。

Flexible NetFlow 中为头部和数据包段落字节，添加了版本 9 的导出格式字段类型。Flexible NetFlow 会把版本 9 导出模版字段中配置的段落大小通知 NetFlow 收集器。负载部分也会有一个相应的长度字段，可以用来收集这一部分的实际字节大小。

注释： 在 Inspur INOS 12.2(50)SY 版本中，不支持数据包段落和负载。

Flexible NetFlow 匹配参数

下面这个表格中描述了 Flexible NetFlow 匹配参数。用户必须为流记录配置至少一个匹配参数。

表 82：匹配参数

命令	目的
match datalink {**dot1q** | **ethertype** | **mac** | **vlan**}	匹配数据链路层或二层字段。用户可以选择使用以下命令选项： • **dot1q**——匹配 802.1Q 字段 • **ethertype**——匹配数据包的以太类型字段 • **mac**——匹配源或目的 MAC 地址字段 • **vlan**——匹配数据包所属的 VLAN（入站或出站）
match flow direction	匹配流标识字段
match interface {**input** | **output**}	匹配接口字段。用户可以选择使用以下命令选项： • **input**——匹配入站接口

	• **output**——匹配出站接口
match ipv4 {**destination** | **protocol** | **source** | **tos** | **ttl** | **version**}	匹配 IPv4 字段。用户可以选择使用以下命令选项： • **destination**——匹配 IPv4 目的地址字段 • **protocol**——匹配 IPv4 协议 • **source**——匹配 IPv4 源地址字段 • **tos**——匹配 IPv4 服务类型字段 • **ttl**——匹配 IPv4 生存时间字段 • **version**——匹配 IPv4 头部中的 IP 版本
match ipv6 {**destination** | **hop-limit** | **protocol** | **source** | **traffic-class** | **version**}	匹配 IPv6 字段。用户可以选择使用以下命令选项： • **destination**——匹配 IPv6 目的地址字段 • **hop-limit**——匹配 IPv6 跳数限制字段 • **protocol**——匹配 IPv6 协议 • **source**——匹配 IPv6 源地址字段 • **traffic-class**——匹配 IPv6 流量类别字段 • **version**——匹配 IPv6 头部中的 IP 版本
match transport {**destination-port** | **igmp** | **icmp** | **source-port**}	匹配传输层字段。用户可以选择使用以下命令选项： • **destination-port**——匹配传输层目的端口 • **icmp**——匹配 ICMP 字段，其中包括 ICMP IPv4 和 IPv6 字段 • **igmp**——匹配 IGMP 字段 • **source-port**——匹配传输层源端口

Flexible NetFlow 收集参数

下面这个表格中描述了 Flexible NetFlow 收集参数。

表 83：收集参数

命令	目的
collect counter { **bytes** { **layer2** { **long** } | **long** } | **packets** { **long** } }	收集计数器字段，总字节数和总数据包数
collect interface {**input** | **output**}	从入站或出站接口收集字段

collect timestamp absolute {first | last}	在指定时间收集字段，第一次见到数据包时，或最后看到数据包时（以毫秒为单位）
collect transport tcp flags	收集下列传输层 TCP 标记： • **ack——**TCP 确认标记 • **cwr**——TCP 拥塞窗口减小标记 • **ece**——TCP ECN Echo 标记 • **fin**——TCP 完成标记 • **psh**——TCP 推送标记 • **rst**——TCP 重置标记 • **syn**——TCP 同步标记 • **urg**——TCP 紧急标记 **注释：** 用户不能在设备上指定具体收集哪个 TCP 标记。用户只能指定收集传输层 TCP 标记。使用这条命令会收集所有 TCP 标记。

流导出器

流导出器会把数据从流监控器缓存中导出到远端系统中，比如运行了 NetFlow 收集器的服务器，从而使用户能够进行分析和储存。流导出器是以一个不同的实体进行配置的。流导出器会被分配给流监控器，来为流监控器提供数据导出功能。用户可以创建多个流导出器，并把它们分配给一个或多个流监控器，以此提供多个导出目的地。用户可以创建一个导出器，并把它应用到多个流监控器上。

NetFlow 数据导出格式版本 9

NetFlow 的基本输出信息是流记录。随着 NetFlow 的逐渐成熟，出现了多种不同格式的流记录。最新的 NetFlow 导出格式是版本 9。NetFlow 版本 9 导出格式的最大与众不同之处是：它是基于模版的。模版提供了多种记录格式设计，这个特性能够对 NetFlow 服务的未来发展提供支持，无需改变基本的流记录格式。使用模版可以获得以下好处：

- 为 NetFlow 提供收集器或显示服务应用程序的第三方业务合作伙伴，无需在每次添加新的 NetFlow 特性时重新编译自己的应用。它们应该可以使用外部数据文件来记录已知的模版格式；
- 能够向 NetFlow 中快速添加新特性，无需中断当前的部署；
- NetFlow 是“面向未来”的，能够适应新协议或发展中的协议，因为版本 9 格式可以为它们提供支持。

版本 9 导出格式由数据包头部加上一个或多个模版流或数据流的集合构成。模版流集合描述了在未来数据流集合中会呈现的字段。这些数据流集合之后可能会出现在同一个导出数据包或后续导出数据包中。模版流和数据流集合可以混合在一个导出数据包中，如下图所示：

图 82：版本 9 导出数据包

Packet Header	数据包 头部
Template FlowSet（共 2 处）	模版 流集合
Data FlowSet（共 3 处）	数据 流集合

NetFlow 版本 9 会周期性导出模版数据，因此 NetFlow 收集器能够知道要发送哪些数据，以及为模版导出数据流集合。对于 Flexible NetFlow 来说有一个重要优势，那就是用户配置的流记录可以有效地转换成版本 9 模版，然后转发给收集器。下图详细展示了 NetFlow 版本 9 导出格式示例，其中包括头部、模版流和数据流集合。

图 83：NetFlow 版本 9 导出格式的详细示例

Header	头部
First Template FlowSet	第 1 个模版流集合
Template Record（共 3 处）	模版记录
First Record FlowSet (Template ID 256)	第 1 个记录流集合 （模版 ID 256）
First Data Record	第 1 个数据记录
Second Data Record	第 2 个数据记录
Thrid Data Record	第 3 个数据记录
Second Template FlowSet	第 2 个模版流集合
Second Template FlowSet (Template ID 257)	第 2 个模版流集合 （模版 ID 257）
Data Record（共 4 处）	数据记录
NetFlow Version 9 Header: 32 bits	NetFlow 版本 9 头部：32 比特
Version 9	版本 9
Count = 4 (FlowSets)	计数= 4（流集合）
System Uptime	系统启动时间
UNIX Seconds	UNIX 秒

Package Sequence	包序列
Source ID	源 ID
Template FlowSet: 16 bits	模版流集合：16 比特
FlowSet ID = 0	流集合 ID = 0
Length = 28 bytes	长度= 28 字节
Field Count = 5	字段计数=5
Length = 4（共 5 处）	长度=4
Data FlowSet: 32 bits	数据流集合：32 比特
FlowSet ID = 256	流集合 ID = 256
Length = 64 bytes	长度= 64 字节

更多有关版本 9 导出格式的信息，可以参考名为 Inspur INOS NetFlow Version 9 Flow-Record Format 的白皮书，网址为：http://www.icntnetworks.com

流监控器

流监控器是 Flexible NetFlow 中的组成部分，在接口上执行网络流量监控。

流控制器由用户定义的记录、可选的流导出器和缓存构成，当用户把流监控器应用在第一个接口上时，设备会自动创建缓存。

流数据是在监控进程执行期间，根据流记录中的键值和非键值字段，从网络流量中收集到的，并会被添加到流监控器缓存中。

Flexible NetFlow 可以用来对相同的流量执行不同类型的分析。在下面这幅图中，入站接口上的标准流量分析功能和出站接口上的安全分析功能都对数据包 1 执行分析。

图 84 使用两个流监控器分析相同流量

Traffic	流量
Flow Monitor 1	流监控器 1
Flow Monitor 2	流监控器 2
Key Fields（共 2 处）	键值字段
Packet 1（共 2 处）	数据包 1
Nonkey Fields（共 2 处）	非键值字段
Source IP（共 4 处）	源 IP

Packets（共 2 处）	数据包
Destination IP（共 2 处）	目的 IP
Bytes	字节
Source port	源端口
Time Stamps（共 2 处）	时间戳
Destination port	目的端口
Next-Hop Address	下一跳地址
Layer 3 Protocol	第 3 层协议
TOS Byte	TOS 字节
Input Interface	入站接口
SYN Flag	SYN 标记
Traffic Analysis Cache（共 2 处）	流量分析缓存
Dest. IP（共 2 处）	目的 IP
Dest. I/F（共 2 处）	目的 I/F
Protocol（共 2 处）	协议
Pkts（共 2 处）	数据包

下图中展示了一个比较复杂的示例，说明用户可以使用自定义记录来应用不同类型的流量监控器。

图 85：使用自定义记录部署多种流监控器类型的复杂案例

Branch	分支机构
Data Center	数据中心
Campus	园区网
Peering Flows	对等体流
IP Flows	IP流
Application Flows	应用流
Security Flows	安全流
Multicast Flows	组播流
Teleworker	远程工作

流监控器缓存有三种类型。用户可以在创建流监控器后修改这个流监控器使用的缓存类型。接下来介绍这三种类型的监控器缓存：

普通

默认的缓存类型为“普通”。在这个模式中，缓存中的条目会根据超时活跃和超时非活跃的设置，在指定时间后超时。当一个缓存条目超时后，它会从缓存中移除并通过任何配置的导出器导出。

立即

“立即”类型的缓存会在每个条目刚一创建的时候马上将其超时。因此每个流中只包含一个数据包。用来查看缓存内容的命令可以展示出这种类型的缓存中曾经看到的数据包历史。

如果用户想要分析的是非常小的流，而且希望最小化看到数据包和导出报告之间的延迟，就可以使用这个模式。

注意： 这个模式可能会产生大量导出数据，这些数据会使低速链路超载，也会淹没接收导出数据的系统。建议用户配置采样来减少需要处理的数据包。

注释： 缓存超时设置在这个模式中不起作用。

永久

“永久”类型的缓存永远不超时任何流。如果用户希望看到的流速率很低，并且没有必要在路由器上维护长时间的状态统计信息的话，就可以使用永久缓存。举例来说，如果流记录中唯一的一个键值字段是 8 比特的 IP ToS 字段，只能监控 256 个流。要想长时间监控网络流量对于 IP ToS 字段的使用，用户可以设置永久缓存。永久缓存有助于应用记账，并且有助于在边界到边界流量模型中追踪一组固定的流。设备会根据“超时更新”的设置，周期性向所有流导出器发送更新消息。

注释： 当永久模式中的缓存已满后，新的流就不会受到监控了。如果缓存已满，缓存状态统计信息中会出现“Flows not added”消息。

注释： 永久缓存使用更新计数器而不是差量计数器。这意味着当一个流被导出时，计数器显示的数值是这个流全部生命周期中看到的总数值，而不是从上一次导出之后的数据包和字节数。

流采样器

流采样器在路由器的配置中是单独创建的。流采样器用来减少设备在运行 Flexible NetFlow 时的负担，也就是限制了用来进行分析的数据包数量。

采样器使用随机采样技术（模式）；也就是说，在每次采样时，随机选择采样位置。

流采样特性为照顾路由器性能，降低了监控精度。当用户对流监控器应用采样器时，路由器上运行流监控器的开销降低了，因为流监控器必须分析的数据包数量减少了。流监控器需要分析的数据包数量的减少，也导致流监控器缓存中储存的信息精度降低了。

用户在接口上配置命令 **ip flow monitor** 时，采样器是和流监控器一起使用的。

支持的 Flexible NetFlow 字段

下面这个表格中列出了 Flexible NetFlow（FNF）为各种流量类型和流量方向，支持的字段。

注释： 如果数据包携带 VLAN 字段，不计算这部分长度。

字段	**二层入站**	**二层出站**	**IPv4入站**	**IPv4出站**	**IPv6入站**	**IPv6出站**	**注释**
键值或收集字段							
接口入站	是	—	是	—	是	—	如果用户在入站方向应用了流监控器： • 使用 **match** 关键字，并把入站接口作为一个键值字段 • 使用 **collect** 关键字，并把出站接口作为一个收集字段。这个字段会出现在导出的记录中，但取值为 0
接口出站	—	是	—	是	—	是	如果用户在出站方向应用了流监控器： • 使用 **match** 关键字，并把出站接口作为一个键值字段 • 使用 **collect** 关键字，并把入站接口作为一个收集字段。这个字段会出现在导出的记录中，但取值为 0

字段	**二层入站**	**二层出站**	**IPv4入站**	**IPv4出站**	**IPv6入站**	**IPv6出站**	**注释**
键值字段							
流方向	是	是	是	是	是	是	
以太类型	是	是	—	—	—	—	

VLAN 入站	是	—	是	—	是	—	只支持交换端口
VLAN 出站	—	是	—	是	—	是	只支持交换端口
802.1Q VLAN 入站	是	—	是	—	是	—	只支持交换端口
802.1Q VLAN 出站	—	是	—	是	—	是	只支持交换端口
802.1Q 优先级	是	是	是	是	是	是	只支持交换端口
MAC 源地址 入站	是	是	是	是	是	是	
MAC 源地址 出站	—	—	—	—	—	—	
MAC 目的地址 入站	是	—	是	—	是	—	
MAC 目的地址 出站	—	是	—	是	—	是	
IPv4 版本	—	—	是	是	是	是	
IPv4 TOS	—	—	是	是	是	是	
IPv4 协议	—	—	是	是	是	是	如果使用了以下字段：源/目的端口、ICMP 代码/类型、IGMP 类型或 TCP 标记，则该字段必须使用
IPv4 TTL	—	—	是	是	是	是	
IPv4 源地址	—	—	是	是	—	—	

IPv4 目的地址	—	—	是	是	—	—	
ICMP IPv4 类型	—	—	是	是	—	—	
ICMP IPv4 代码	—	—	是	是	—	—	
IGMP 类型	—	—	是	是	—	—	
IPv6 版本	—	—	是	是	是	是	与 IP 版本相同
IPv6 协议	—	—	是	是	是	是	与 IP 协议相同。如果使用了以下字段：源/目的端口、ICMP 代码/类型、IGMP 类型或 TCP 标记，则该字段必须使用
IPv6 源地址	—	—	—	—	是	是	
IPv6 目的地址	—	—	—	—	是	是	
IPv6 流量类别	—	—	是	是	是	是	与 IP TOS 相同
IPv6 跳数限制	—	—	是	是	是	是	与 IP TTL 相同
ICMP IPv6 类型	—	—	—	—	是	是	
ICMP IPv6 代码	—	—	—	—	是	是	
源端口	—	—	是	是	是	是	
目的端口	—	—	是	是	是	是	

字段	**二层入站**	**二层出站**	**IPv4入站**	**IPv4出站**	**IPv6入站**	**IPv6出站**	**注释**
收集字段							
字节长度	是	是	是	是	是	是	数据包大小=（以太网数据帧大

							小，其中包含 FCS – 18 字节） **建议：** 避免使用这个字段，使用字节二层长度
数据包长度	是	是	是	是	是	是	
时间戳绝对时间第一次	是	是	是	是	是	是	
时间戳绝对时间最后一次	是	是	是	是	是	是	
TCP 标记	是	是	是	是	是	是	收集所有标记
字节二层长度	是	是	是	是	是	是	

默认设置

下面这个表格中列出了设备上 Flexible NetFlow 的默认设置。

表 84：默认的 Flexible NetFlow 设置

设置	**默认**
流活跃超时时间	180 秒
流超时时间非活跃	15 秒

如何配置 Flexible NetFlow

用户可以按照以下步骤来配置 Flexible NetFlow：

1. 通过为流指定键值和非键值字段，来创建流记录；
2. 通过指定协议和传输层目的端口、目的地和其他参数，来（可选的）创建流导出器；
3. 基于流记录和流导出器创建流监控器；
4. （可选的）创建采样器；
5. 在二层端口、三层端口或 VLAN 上应用流监控器。

配置自定义流记录

用户可以通过这部分介绍的任务来自定义流记录。

自定义流记录可以用来对流量数据执行特定目的的分析。自定义流记录中必须至少有一个 **match** 条件，用来作为键值字段，通常还至少有一个 **collet** 条件，用来作为非键值字段。

自定义流记录的组合方法有上百种。这部分展示的方法能够创建一中组合。用户可以根据自己的需求，通过适当修改步骤中的命令，来创建自定义流记录。

总步骤

1. enable

2. configure terminal

3. flow record *record-name*

4. description *description*

5. match {**ipv4** | **ipv6**} {**destination** | **source**} **address**

6. 按需重复步骤 5 来为这个记录创建其他键值字段。

7.

8. 按需重复上一步来为这个记录创建其他非键值字段。

9. end

10. show flow record *record-name*

11. show running-config flow record *record-name* ***9. end***

10. show flow record *record-name*

11. show running-config flow record *record-name*

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**flow record** *record-name*	创建一个流记录并进入 Flexible

	示例： `Device(config)# flow record FLOW-RECORD-1`	NetFlow 流记录配置模式。 • 用户也可以使用这条命令来修改已有流记录
步骤 4	**description** *description* **示例：** `Device(config-flow-record)# description Used for basic traffic analysis`	**ipv4**（可选）为流记录创建描述信息
步骤 5	**match {ipv4 | ipv6} {destination | source} address** **示例：** `Device(config-flow-record)# match ipv4 destination address`	为流记录配置键值字段。 **注释：** 示例中把 IPv4 目的地址配置为键值字段。命令 **match ipv4** 中可用的其他键值字段，以及其他 **match** 命令，用户可以参考 *Inspur INOS Flexible NetFlow 命令参考*。
步骤 6	重复步骤 5，为整个记录配置其他键值字段	—
步骤 7	**示例：**	把入站接口配置为非键值字段。 **注释：** 示例中把入站接口配置为非键值字段。用来配置非键值字段的其他 **collect** 命令，用户可以参考 *Inspur INOS Flexible NetFlow 命令参考*。
步骤 8	按需重复上一步骤，为这个记录配置其他非键值字段	—
步骤 9	**end** **示例：** `Device(config-flow-record)# end`	退出 Flexible NetFlow 流记录配置模式并返回特权 EXEC 模式
步骤 10	**show flow record** *record-name*	（可选）显示指定流记录的当前状态

	示例： `Device# show flow record FLOW_RECORD-1`	
步骤 11	**show running-config flow record** *record-name* **示例：** `Device# show running-config flow record FLOW_RECORD-1`	（可选）显示指定流记录的配置

创建流导出器

用户可以创建一个流导出器，为一个流定义导出参数。

注释： 每个流导出器中只支持一个目的地。如果用户希望把数据导出到多个目的地，必须配置多个流导出器，并把它们都分配给流监控器。

用户可以使用 IPv4 地址或 IPv6 地址作为导出目的地。

总步骤

1. configure terminal

2. flow exporter *name*

3. description *string*

4. destination {*ipv4-address*|*ipv6-address*}

5. dscp *value*

6. source { *source type* }

7. transport udp *number*

8. ttl *seconds*

9. export-protocol {**netflow-v9** | **ipfix**}

10. end

11. show flow exporter [**name** *record-name*]

12. copy running-config startup-config

具体步骤

	命令或操作	目的

步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**flow exporter** *name* **示例：** `Device(config)# flow exporter ExportTest`	创建一个流导出器并进入流导出器配置模式
步骤 3	**description** *string* **示例：** `Device(config-flow-exporter)# description ExportV9`	（可选）描述这个流记录，最长为 63 字节的字符串
步骤 4	**destination** {*ipv4-address*|*ipv6-address*} **示例：** `Device(config-flow-exporter)# destination 192.0.2.1`（IPv4 目的地） `Device(config-flow-exporter)# destination 2001:0:0:24::10`（IPv6 目的地）	为这个导出器设置 IPv4/IPv6 目的地址或主机名
步骤 5	**dscp** *value* **示例：** `Device(config-flow-exporter)# dscp 0`	（可选）指定查分服务代码点值。取值范围是 0 至 63，默认值为 0
步骤 6	**source** { *source type* } **示例：** `Device(config-flow-exporter)#`	（可选）指定用来去往 NetFlow 收集器（配置的目的地）的接口。用户可以把以下接口配置为源： • **Auto Template**——Auto-Template

	source gigabitEthernet1/0/1	接口 • **Capwap**——CAPWAP 隧道接口 • **GigabitEthernet**——千兆以太网接口 IEEE 802 • **GroupVI**——组虚拟接口 • **Internal Interface**——内部接口 • **Loopback**——环回接口 • **Null**——空接口 • **Port-channel**——以太网通道接口 • **TenGigabitEthernet**——万兆以太网接口 • **Tunnel**——隧道接口 • **Vlan**——Inspur VLAN
步骤 7	**transport udp** *number* **示例：** `Device(config-flow-exporter)#` **`transport udp 200`**	（可选）指定去往 NetFlow 收集器的 UDP 端口。取值范围是 0 至 65535。对于 IPFIX 导出协议来说，默认目的端口是 4739
步骤 8	**ttl** *seconds* **示例：** `Device(config-flow-exporter)#` **`ttl 210`**	（可选）为导出器发送的数据报配置生存时间（TTL）值。取值范围是 1 至 255，默认值为 255

步骤 9	**export-protocol** {**netflow-v9** | **ipfix**}
	
	示例：
	`Device(config-flow-exporter)#`
	`export-protocol netflow-v9`
		指定导出器使用的 NetFlow 导出协议版本。
		• 默认：**netflow-v9**

步骤 10	**end** **示例：**	返回特权 EXEC 模式

	Device(config-flow-record)# **end**	
步骤 11	**show flow exporter** [**name** *record-name*] **示例：** Device **show flow exporter ExportTest**	（可选）显示有关 NetFlow 流导出器的信息
步骤 12	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

接下来做什么？

根据流记录和流导出器，定义流监控器。

创建自定义流监控器

用户可以通过完成这个任务来自定义流监控器。

每个流监控器都有一个不同的缓存与其相关联。每个流监控器都需要一个记录，来定义缓存条目的内容和输出方式。这些记录格式可以是预定义的格式，或者用户定义的格式。高级用户可以使用命令 **flow record** 来创建自定义格式。

在开始前

如果用户希望使用自定义记录，而不是 Flexible NetFlow 中某个预定义记录，用户必须在执行这个任务前，先创建自定义记录。如果用户希望为流监控器添加一个流导出器来导出数据，用户必须在完成这个任务前创建导出器。

注释： 用户必须使用命令 **no ip flow monitor**，移除所有接口上应用的流监控器，之后才能修改在流监控器上修改 **record** 命令的参数。有关 **ip flow monitor** 命令的更多信息，用户可以参考 *Inspur INOS Flexible NetFlow 命令参考*。

总步骤

1. enable

2. configure terminal

3. flow monitor *monitor-name*

4. description *description*

5. record {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6**} *record* [**peer**]}

6. cache {**entries** *number* | **timeout** {**active** | **inactive** | **update**} *seconds* | {**immediate** | **normal** | **permanent**}}

7. 按需重复步骤 6，完成对整个流监控器的缓存参数修改。

8. statistics packet protocol

9. statistics packet size

10. exporter *exporter-name*

11. end

12. show flow monitor [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]] [**statistics**]]

13. show running-config flow monitor *monitor-name*

14. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**flow monitor** *monitor-name* **示例：** `Device(config)# flow monitor FLOW-MONITOR-1`	创建一个流监控器并进入 Flexible NetFlow 流监控器配置模式。 • 用户还可以使用这条命令来修改已有的流监控器
步骤 4	**description** *description* **示例：** `Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis`	（可选）为这个流控制器创建描述信息

步骤 5	**record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6**} *record* [**peer**]} **注释：** `Device(config-flow-monitor)# record FLOW-RECORD-1`	为这个流控制器指定记录
步骤 6	**cache** {**entries** *number* | **timeout** {**active** | **inactive** | **update**} *seconds* | {**immediate** | **normal** | **permanent**}} **示例：**	当用户把缓存类型设置为 **immediate** 时，**timeout** 关键字中的相关取值不起作用。 为这个流控制器关联流缓存
步骤 7	按需重复步骤 6，以便为这个流监控器完成对缓存参数的修改	—
步骤 8	**statistics packet protocol** **示例：** `Device(config-flow-monitor)# statistics packet protocol`	（可选）为 Flexible NetFlow 监控器启用协议分布式统计状态信息的收集功能
步骤 9	**statistics packet size** **示例：** `Device(config-flow-monitor)# statistics packet size`	（可选）为 Flexible NetFlow 监控器启用大小分布式统计状态信息的收集功能
步骤 10	**exporter** *exporter-name* **示例：** `Device(config-flow-monitor)# exporter EXPORTER-1`	（可选）指定之前创建的导出器名称
步骤 11	**end**	退出 Flexible NetFlow 流监控器配置模式并返回特权 EXEC 模式

	示例： `Device(config-flow-monitor)#` **`end`**	
步骤 12	**showflowmonitor** [[**name**]*monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]] [**statistics**]] **示例：** `Device# show flow monitor FLOW-MONITOR-2 cache`	（可选）显示 Flexible NetFlow 流监控器的状态和统计状态信息
步骤 13	**show running-config flow monitor** *monitor-name* **示例：** `Device# show running-config flow monitor FLOW_MONITOR-1`	（可选）显示指定流监控器的配置
步骤 14	**copy running-config startup-config** **示例：** `Device#` **`copy running-config startup-config`**	（可选）把输入的命令保存到配置文件中

配置并启用流采样

创建流采样器

用户通过执行这个任务可以配置并启用流采样器。

注释： 当用户为流监控器使用预定义记录“NetFlow original”、“NetFlow IPv4 original input”或“NetFlow IPv6 original input”来模拟最初的 NetFlow 时，流监控器就只能用来分析入站（入向）流量。

当用户为流监控器使用预定义记录“NetFlow IPv4 original output”或“NetFlow IPv6 original output”来模拟最初的出向 NetFlow 审计特性时，流监控器就只能用来分析出站（出向）流

量。

总步骤

1. enable

2. configure terminal

3. sampler *sampler-name*

4. description *description*

5. mode {**random**} **1 out-of** *window-size*

6. exit

7. interface *type number*

8. {**ip** | **ipv6**} **flow monitor** *monitor-name* [[**sampler**] *sampler-name*] {**input** | **output**}

9. end

10. show sampler sampler-name

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。 • 在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**sampler** *sampler-name* **示例：** `Device(config)# sampler SAMPLER-1`	创建一个采样器并进入采样器配置模式： • 用户也可以使用这条命令来修改已有的采样器
步骤 4	**description** *description* **示例：** `Device(config-sampler)# description Sample at 50%`	（可选）为这个流采样器创建描述信息

步骤 5	**mode** {**random**} **1 out-of** *window-size* **示例：** `Device(config-sampler)# mode random 1 out-of 2`	指定采样器模式和流采样器窗口大小。 • *window-size* 变量的取值范围是 0 至 1024，2 至 32768
步骤 6	**exit** **示例：** `Device(config-sampler)# exit`	退出采样配置模式并返回全局配置模式
步骤 7	**interface** *type number* **示例：** `Device(config)# interface GigabitEthernet 0/0/0`	指定一个接口并进入接口配置模式
步骤 8	{**ip** | **ipv6**} **flow monitor** *monitor-name* [[**sampler**] *sampler-name*] {**input** | **output**} **示例：** `Device(config-if)# ip flow monitor FLOW-MONITOR-1 sampler SAMPLER-1 input`	把之前创建的流监控器和流采样器分配到接口上来启用采样工作
步骤 9	**end** **示例：** `Device(config-if)# end`	退出接口配置模式并返回特权 EXEC 模式
步骤 10	**show sampler** *sampler-name* **示例：** `Device# show sampler SAMPLER-1`	显示指定流采样器的状态和统计状态信息

在接口上应用流

用户可以在接口上应用流监控器，并可选地应用采样器。

总步骤

1. configure terminal

2. interface *type*

3. {**ip flow monitor** | **ipv6 flow monitor**}*name* [|**sampler** *name*] { **input**}

4. end

5. show flow interface [*interface-type number*]

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**interface** *type* **示例：** `Device(config)# interface GigabitEthernet1/0/1`	进入接口配置模式并配置接口。 二层 PortChannel 接口上不支持 Flexible NetFlow，但二层 PortChannel 成员端口上可以支持 Flexible NetFlow。 三层 PortChannel 接口上不支持 Flexible NetFlow，但三层 PortChannel 成员端口上可以支持 Flexible NetFlow。 接口配置中支持以下参数： • **GigabitEthernet**——千兆以太网接口 IEEE 802 • **Loopback**——环回接口 • **TenGigabitEthernet**——万兆以太网接口 ● **Vlan**——Inspur VLAN ● **Range**——接口范围 ● ~~**WLAN**——WLAN 接口~~

步骤 3	{**ip flow monitor** | **ipv6 flow monitor**}*name* [|**sampler** *name*] { **input**} **示例：** `Device(config-if)# ip flow monitor MonitorTest input`	在接口上关联一个 IPv4 或 IPv6 流管理器，并且可选的关联一个采样器，用来监控入站或出站数据包
步骤 4	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 5	**show flow interface** [*interface-type number*] **示例：** `Device# show flow interface`	（可选）显示有关接口上 NetFlow 的信息
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

在 VLAN 上配置桥接的 NetFlow

用户可以在 VLAN 上应用流监控器，还可选的应用采样器。

总步骤

1. configure terminal

2. vlan [**configuration**] *vlan-id*

3. ip flow monitor *monitor name* [**sampler** *sampler name*] {**input** |**output**}

4. copy running-config startup-config

具体步骤

	命令或操作	目的

步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 2	**vlan** [**configuration**] *vlan-id*
	
	示例：
	`Device(config)# vlan configuration 30`
	`Device(config-vlan-config)#`
		进入 VLAN 或 VLAN 配置模式

步骤 3	**ip flowmonitor***monitor name* [**sampler** *sampler name*] {**input** |**output**} **示例：** `Device(config-vlan-config)# ip flow monitor MonitorTest input`	在 VLAN 上关联一个流控制器，并且可选的关联一个采样器，来监控入站或出站数据包
步骤 4	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置二层 NetFlow

用户可以在 Flexible NetFlow 记录中定义二层键值，以此来捕获二层接口中的流。

总步骤

1. configure terminal

2. flow record *name*

3. match datalink {**dot1q** |**ethertype** | **mac** | **vlan**}

4. end

5. show flow record [*name*]

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式

步骤 2	**flow record** *name*
	
	示例：
	`Device(config)# flow record L2_record`
	`Device(config-flow-record)#`
		进入流记录配置模式

步骤 3	**match datalink** {**dot1q** |**ethertype** | **mac** | **vlan**} 示例： `Device(config-flow-record)# match datalink ethertype`	把二层属性指定为一个键值
步骤 4	**end** 示例： `Device(config-flow-record)# end`	返回特权 EXEC 模式
步骤 5	**show flow record** [*name*] 示例： `Device# show flow record`	（可选）显示接口上相关的 NetFlow 信息
步骤 6	**copy running-config startup-config** 示例： `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

监控 Flexible NetFlow

用户可以使用下面表格中列出的命令来对 Flexible NetFlow 实施监控。

表 85：Flexible NetFlow 监控命令

命令	目的
show flow exporter [**broker** | **export-ids** | **name** | *name* | **statistics** | **templates**]	显示有关 NetFlow 流导出器和统计状态的信息
show flow exporter [**name** *exporter-name*]	显示有关 NetFlow 流导出器和统计状态的信息
show flow interface	显示有关 NetFlow 接口的信息
show flow monitor [**name** *exporter-name*]	显示有关 NetFlow 流监控器和统计状态的信息
show flow monitor statistics	显示流监控器的统计状态信息
show flow monitor cache format {**table** | **record** | **csv**}	显示流监控器缓存中的内容，用户可以指定格式
show flow record [**name** *record-name*]	显示有关 NetFlow 流记录的信息
~~**show flow ssid**~~	~~显示用于 WLAN 的 NetFlow 监控器安装信息~~
show sampler [**broker** | **name** | *name*]	显示有关 NetFlow 采样器的信息
~~**show wlan** *wlan-name*~~	~~显示设备上的 WLAN 配置~~

Flexible NetFlow 的配置示例

示例：配置一个流

这个示例展示了如何创建一个流，并将其应用在接口上：

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# flow export export1
Device(config-flow-exporter)# destination 10.0.101.254
Device(config-flow-exporter)# transport udp 2055
```

```
Device(config-flow-exporter)# exit
Device(config)# flow record record1
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match ipv4 protocol
Device(config-flow-record)# match transport source-port
Device(config-flow-record)# match transport destination-port
Device(config-flow-record)# collect counter byte long
Device(config-flow-record)# collect counter packet long
Device(config-flow-record)# collect timestamp absolute first
Device(config-flow-record)# collect timestamp absolute last
Device(config-flow-record)# exit
Device(config)# flow monitor monitor1
Device(config-flow-monitor)# record record1
Device(config-flow-monitor)# exporter export1
Device(config-flow-monitor)# exit
Device(config)# interface tenGigabitEthernet 1/0/1
Device(config-if)# ip flow monitor monitor1 input
Device(config-if)# end
```

示例：监控 IPv4 入站流量

这个示例展示了如何监控 IPv4 入站流量（接口 G1/0/11 向接口 G1/0/36 和接口 G3/0/11 发送流量）。

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# flow record fr-1
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match interface input
Device(config-flow-record)# collect counter bytes long
Device(config-flow-record)# collect counter packets long
Device(config-flow-record)# collect timestamp absolute first
```

```
Device(config-flow-record)# collect timestamp absolute last
Device(config-flow-record)# collect counter bytes layer2 long
Device(config-flow-record)# exit
Device(config)# flow exporter fe-ipfix6
Device(config-flow-exporter)# destination 2001:0:0:24::10
Device(config-flow-exporter)# source Vlan106
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit
Device(config)# flow exporter fe-ipfix
Device(config-flow-exporter)# description IPFIX format collector
100.0.0.80
Device(config-flow-exporter)# destination 100.0.0.80
Device(config-flow-exporter)# dscp 30
Device(config-flow-exporter)# ttl 210
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit
Device(config)# flow exporter fe-1
Device(config-flow-exporter)# destination 10.5.120.16
Device(config-flow-exporter)# source Vlan105
Device(config-flow-exporter)# dscp 32
Device(config-flow-exporter)# ttl 200
Device(config-flow-exporter)# transport udp 2055
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit
Device(config)# flow monitor fm-1
Device(config-flow-monitor)# exporter fe-ipfix6
Device(config-flow-monitor)# exporter fe-ipfix
Device(config-flow-monitor)# exporter fe-1
Device(config-flow-monitor)# cache timeout inactive 60
```

```
Device(config-flow-monitor)# cache timeout active 180
Device(config-flow-monitor)# record fr-1
Device(config-flow-monitor)# end
Device# show running-config interface g1/0/11
Device# show running-config interface g1/0/36
Device# show running-config interface g3/0/11
Device# show flow monitor fm-1 cache format table
```

示例：监控 IPv4 出站流量

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# flow record fr-1 out
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match interface output
Device(config-flow-record)# collect counter bytes long
Device(config-flow-record)# collect counter packets long
Device(config-flow-record)# collect timestamp absolute first
Device(config-flow-record)# collect timestamp absolute last
Device(config-flow-record)# exit
Device(config)# flow exporter fe-1
Device(config-flow-exporter)# destination 10.5.120.16
Device(config-flow-exporter)# source Vlan105
Device(config-flow-exporter)# dscp 32
Device(config-flow-exporter)# ttl 200
Device(config-flow-exporter)# transport udp 2055
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit
Device(config)# flow exporter fe-ipfix6
Device(config-flow-exporter)# destination 2001:0:0:24::10
Device(config-flow-exporter)# source Vlan106
Device(config-flow-exporter)# transport udp 4739
```

```
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit
Device(config)# flow exporter fe-ipfix
Device(config-flow-exporter)# description IPFIX format collector
100.0.0.80
Device(config-flow-exporter)# destination 100.0.0.80
Device(config-flow-exporter)# dscp 30
Device(config-flow-exporter)# ttl 210
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit
Device(config)# flow monitor fm-1-output
Device(config-flow-monitor)# exporter fe-1
Device(config-flow-monitor)# exporter fe-ipfix6
Device(config-flow-monitor)# exporter fe-ipfix
Device(config-flow-monitor)# cache timeout inactive 50
Device(config-flow-monitor)# cache timeout active 120
Device(config-flow-monitor)# record fr-1-out
Device(config-flow-monitor)# end
Device# show flow monitor fm-1-output cache format table
```

导出器

创建流监控器

监控器

注释： 在设备上，用户不能指定需要收集的 TCP 标记。用户只能指定需要收集传输层 TCP 标记。

其他参考资料

错误消息解码器

描述	链接
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

标准和 RFC

标准/RFC	标题
RFC 3954	Cisco Systems NetFlow Services Export Version9

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	http://www.icntnetworks.com

Flexible NetFlow 的特性信息

版本	变更
Inspur INOS 11.3.1	引入该特性

第 11 部分 QoS

配置 QoS

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置 Auto-QoS 的先决条件

配置 Auto-QoS 的先决条件与配置标准 QoS 的先决条件相同。

配置 Auto-QoS 的限制条件

配置 Auto-QoS 有以下限制条件：

- SVI 接口不支持 Auto-QoS；
- 绑定在 EtherChannel 中的接口不支持 Auto-QoS；
- 接口配置模式中的命令 **trust device** *device_type* 是交换机上的独立命令。在 AutoQoS 配置中使用这条命令时，如果连接的对等体设备不是对应设备（也就是符合用户信任策略的设备），那么 CoS 和 DSCP 值都会设置为“0”，任何入站策略也不会生效。如果连接的对等体设备是对应设备，那么入站策略就会生效；
- 用户在这台设备中使用 3.2.2 版本之前的软件时需要注意。如果用户在这台设备中使用了 3.2.2 版本之前的软件，那么必须按照后文中介绍的 Auto-QoS 升级流程进行升级；
- 不要为支持视频的 IP 电话配置 **auto qos voip inspur-phone** 选项。这个选项会重写视频数据包的 DSCP 标记，由于这些数据包不具备加速转发优先级，因此这种做法会导致这些数据包被分类为 class-default 类；
- 在用户使用命令 **auto qos voip inspur-phone**，把 Auto-QoS 从启动配置推送到运行配置中时，它并不会生成配置。这就是预期的效果，这样做是为了每次从启动配置文件中推送 **auto qos voip inspur-phone** 命令时，如果有用户自定义的 QoS 策略，防止默认配置覆盖用户创建的自定义 QoS 策略。

  用户可以使用以下解决方法来突破这一限制：

  - 在交换机接口上手动配置命令 **auto qos voip inspur-phone**；
  - 对于新的交换机来说，如果用户从启动配置中推送 Auto-QoS 命令，命令应该会在标准模版中包含以下部分：

    1. 接口级别：
       - **trust device inspur-phone**
       - **auto qos voip inspur-phone**
       - **service-policy input** AutoQoS-4.0-InspurPhone-Input-Policy
       - **service-policy output** AutoQos-4.0-Output-Policy
    2. 全局级别：
       - class-map
       - policy-map
       - ACL（ACE）

- 如果接口上已经配置了命令 **auto qos voip inspur-phone**，但还没有生成策略，用户就需要在所有接口上禁用这条命令，然后在每个接口上重新进行配置。

配置 Auto-QoS 的相关信息

QoS 概述

用户可以使用 Auto-QoS 特性来简化 QoS 特性的部署。Auto-QoS 能够确定网络设计，并启用 QoS 配置，这样交换机能够对不同的流量执行优先级不同的操作。

交换机上能够部署 MQC 模型。这意味着 Auto-QoS 不使用某些全局配置，而是在交换机的接口上应用一些全局 class-map 和 policy-map。

Auto-QoS 可以匹配流量，然后把匹配的数据包分到 qos-group 中。这种做法能够让出站 policy-map 把特定 qos-group 中的数据包放入指定的队列中，其中包括优先级队列。

QoS 需要双向设置，入向和出向。在入方向上，交换机端口需要信任数据包中的 DSCP（默认行为）。在出方向上，交换机端口需要为语音数据包提供“优先转发”优先级。如果语音在出向队列中的其他数据包后面排队等待发送，就会经历过长的延迟，终端主机会丢弃这个数据包，因为这个数据包到达的时间已经超出了应该接收这个数据包的时间窗口。

Auto-QoS 集合特性概述

在用户输入 Auto-QoS 命令时，交换机会显示出所有它自己生成的命令，就好像这些命令是通过 CLI 输入的一样。用户可以使用 Auto-QoS 集合（Compact）特性在运行配置中隐藏 Auto-QoS 生成的命令。这样做可以使用户更轻松地读懂运行配置，同时提高内存的利用效率。

Auto-QoS 全局配置模版

通常来说，Auto-QoS 命令会生成一系列 class-map 命令，这些命令会对 ACL 或 DSCP 和/或 CoS 值进行匹配，然后把匹配的流量放到不同的应用类别中。Auto-QoS 还会生成入向策略，它会匹配生成的类别，在一些情况中，还会把类别限速为指定带宽。Auto-QoS 会生成 8 个出向队列 class-map。实际的出向策略会把一条队列分配到（拥有这 8 条出向队列的）class-map

中的某条队列中。

Auto-QoS 命令只会按需生成模版。举例来说，当用户第一次使用新的 Auto-QoS 命令时，会生成定义了 8 条队列的出向 service-policy 全局配置。从这时开始，应用到其他接口的 Auto-QoS 命令不会再为出向队列生成模版，因为所有的 Auto-QoS 命令都使用相同的 8 队列模型，而这个模型在第一次输入新的 Auto-QoS 命令时就已经生成了。

Auto-QoS policy-map 和 class-map

在输入了适当的 Auto-QoS 命令后，会发生以下事件：

- 创建指定的 class-map；
- 创建指定的 policy-map（入向和出向）；
- 把 policy-map 关联到指定接口；
- 为接口配置信任等级。

Auto-QoS 对运行配置的影响

在启用 Auto-QoS 时，交换机会把接口配置命令 **auto qos** 和生成的全局配置添加到运行配置中。

交换机在应用 Auto-QoS 生成的命令时，就好像这些命令是通过 CLI 输入的一样。现有的用户配置可能会导致生成的命令应用失败，或者被生成的命令覆盖。这些行为在发生时并不会有警告信息。如果生成的所有命令都成功应用了，那么用户输入的那些没被覆盖的配置会保留在运行配置中。用户输入的被覆盖了的配置可以在不把当前配置保存到内存中，并重启交换机来恢复。如果生成的命令没有应用成功，则会恢复之前的运行配置。

Auto-QoS 集合特性对运行配置的影响

如果用户启用了 Auto-QoS 集合特性：

- 在运行配置中只会显示出用户在 CLI 中输入的 Auto-QoS 命令；
- 生成的全局配置和接口配置是隐藏的；
- 在用户保存配置时，只有用户输入的 Auto-QoS 命令会被保存（隐藏配置不会被保存）；
- 在用户重启交换机后，系统会检测并重新执行保存的 Auto-QoS 命令，并生成符合 AutoQoS SRND4.0 的配置集。

注释： 在启用了 Auto-QoS 集合特性后，用户不要对 Auto-QoS 生成的命令做修改，因为用

户的变更会在交换机重启时被覆盖。

如果用户启用了 auto qos global compact：

- 可以使用 **show derived-config** 命令来查看隐藏的 AQC 命令；
- AQC 命令不会储存到内存中。在每次交换机重启后会生成这些命令；
- 在启用了 Auto-QoS 集合后，用户不应该修改 Auto-QoS 生成的命令；
- 如果接口上配置了 Auto-QoS，并且如果用户需要禁用 AQC，那么用户应该先在接口上禁用 Auto-QoS。

如何配置 Auto-QoS

配置 Auto-QoS（CLI）

为了优化 QoS 的性能，用户应该在网络中的所有设备上都配置 Auto-QoS。

总步骤

1. configure terminal

2. interface *interface-id*

3. 根据用户的Auto-QoS配置，使用以下命令之一：

- **auto qos voip** {**inspur-phone** | **inspur-softphone** | **trust**}
- **auto qos video** {**cts** | **ip-camera** | **media-player**}
- **auto qos classify** [**police**]
- **auto qos trust** {**cos** | **dscp**}

4. end

5. show auto qos interface *interface-id*

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**interface** *interface-id*	指定端口：连接着 VoIP 端口、视频设备，或上行端口连接着另一台网络内部

	示例： `Device(config)# interface gigabitethernet 3/0/1`	的可信交换机或路由器，并进入接口配置模式
步骤 3	根据用户的Auto-QoS配置，使用以下命令之一： • **auto qos voip** {**inspur-phone** | **inspur-softphone** | **trust**} • **auto qos video** {**cts** | **ip-camera** | **media-player**} • **auto qos classify** [**police**] • **auto qos trust** {**cos** | **dscp**} **示例：** `Device(config-if)# auto qos trust dscp`	以下命令为 VoIP 启用 Auto-QoS： • **auto qos voip inspur-phone**——如果端口连接着 Inspur IP 电话，那么只有当检测到了 IP 电话，入站数据包上的 QoS 标签才会被信任（通过 CDP 实现条件信任）。 **注释：** 不要为支持视频的 IP 电话配置命令 **auto qos voip inspur-phone**。这个选项会重写视频数据包的 DSCP 标记，由于这些数据包不具备加速转发优先级，因此这种做法会导致这些数据包被分类为 class-default 类。 • **auto qos voip inspur-softphone**——这个端口连接着运行 Inspur 软电话特性的设备。这条命令会为运行 Inspur IP 软电话应用的 PC 生成 QoS 配置，可以标记和限速从这个接口进入的流量。配置了这条命令的接口是不受信任的接口 • **auto qos voip trust**——上行链路端口连接着可信交换机或路由器，并且信任入向数据包中的 VoIP 流量分类。 用户可以使用以下命令为指定的视频设备（系统、摄像头或媒体播放器）启用 Auto-QoS： • **auto qos video cts**——端口连接着

		Inspur 网真系统。只有当检测到 Inspur 网真系统时，才会信任入站数据包的 QoS 标签（通过 CDP 实现条件信任） • **auto qos video ip-camera**——端口连接着 Inspur 视频监视摄像头。只有当检测到 Inspur 摄像头时，才会信任入站数据包的 QoS 标签（通过 CDP 实现条件信任） • **auto qos video media-player**——端口连接着支持 CDP 的 Inspur 数字媒体播放器。只有当检测到数字媒体播放器时，才会信任入站数据包的 QoS 标签（通过 CDP 实现条件信任） 用户可以使用以下命令来启用 Auto-QoS 分类功能： • **auto qos classify police**——这条命令为不可信接口生成 QoS 配置。这个配置会在接口上应用 service-policy，用来分类从不可信的桌面/设备进来的流量，为其分配相应的标记。生成的 service-policy 也可以用来实现限速 用户可以使用以下命令来为可信接口启用 Auto-QoS： • **auto qos trust cos**——服务类别 • **auto qos trust dscp**——查分服务代码点
步骤 4	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式

步骤 5	**showauto qos interface** *interface-id* **示例：** Device# **show auto qos interface gigabitethernet 3/0/1**	（可选）显示启用了 Auto-QoS 的接口上的 Auto-QoS 命令。用户可以使用 **show running-config** 命令来查看 Auto-QoS 配置和用户对其的修改

升级 Auto-QoS（CLI）

只有在设备上使用3.2.2版本之前的软件时，用户才需要按照这个步骤来升级Auto-QoS特性。如果用户确实在设备上使用了 3.2.2 版本之前的软件，就必须执行这个 Auto-QoS 升级过程。

在开始前

在开始升级之前，用户需要移除当前交换机上的所有 Auto-QoS 配置。示例流程中展示了这个过程。

在完成示例步骤后，用户必须用新的或升级后的软件镜像来重启交换机并重新配置 Auto-QoS。

总步骤

1. show auto qos

2. no auto qos

3. show running-config | i autoQos

4. no policy-map *policy-map_name*

5. show running-config | i AutoQoS

6. show auto qos

7. write memory

具体步骤

步骤1. show auto qos

示例：

```
Device# show auto qos
GigabitEthernet2/0/3
auto qos voip inspur-phone
GigabitEthernet2/0/27
auto qos voip inspur-softphone
```

在特权 EXEC 模式中，输入这条命令记录所有当前的 Auto-QoS 配置。

步骤2. **no auto qos**

示例：

```
Device(config-if)#no auto qos
```

在接口配置模式中，为所有拥有 Auto-QoS 配置的接口使用命令 **no auto qos**。

步骤3. **show running-config | i autoQos**

示例：

```
Device# show running-config | i autoQos
```

返回到特权 EXEC 模式中，输入这条命令并记录所有剩下的 Auto-QoS class-map、policy-map、访问列表、table-map 或其他配置。

步骤4. **no policy-map** *policy-map_name*

示例：

```
Device)config# no policy-map pmap_101
Device)config# no class-map cmap_101
Device)config# no ip access-list extended AutoQos-101
Device)config# no table-map 101
Device)config# no table-map policed-dscp
```

在全局配置模式中，输入以下命令删除 QoS class-map、policy-map、access-list、table-map，以及任何其他 Auto-QoS 配置：

- **no policy-map** *policy-map-name*
- **no class-map** *class-map-name*
- **no ip access-list extended** *Auto-QoS-x*
- **no table-map** *table-map-name*
- **no table-map policed-dscp**

步骤5. **show running-config | i AutoQoS**

示例：

```
Device# show running-config | i AutoQos
```

返回到特权 EXEC 模式，再次使用这条命令确认设备中已经没有 Auto-QoS 配置，或者已经没有 Auto-QoS 的残留配置。

步骤6. **show auto qos**

示例：

```
Device# show auto qos
```

使用这条命令确保配置中已经不存在 Auto-QoS 配置或遗留的部分配置。

步骤7. **write memory**

示例：

```
Device# write memory
```

使用 **write memory** 命令，把 Auto-QoS 配置的变更写入 NV 内存中。

接下来做什么？

使用新的或升级后的软件镜像重启交换机。

使用新的或升级后的软件镜像重启交换机后，按照步骤 1 中命令 **show auto qos** 的输出内容个，为相应的交换机接口重新配置 Auto-QoS。

注释： 每台交换机或堆栈中只有一个 table-map 用来为超出流量进行标记，另一个 table-map 对违规流量进行标记。如果交换机在超出行为下已经有了 table-map，用户就无法应用 Auto-QoS 策略了。

启用 Auto-QoS 集合

用户需要使用以下命令来启用 Auto-QoS 集合：

总步骤

1. configure terminal

2. auto qos global compact

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**auto qos global compact** **示例：** `Device(config)# auto qos global compact`	启用 Auto-QoS 集合，并为 Auto-QoS 生成（隐藏的）全局配置。 用户可以在接口配置模式中输入想要配置的 Auto-QoS 命令，由系统生成的接口配置也是隐藏的。 要想查看已经应用的 Auto-QoS 配置，用户可以使用以下特权 EXEC 模式的命令： **• show derived-config**

		• **show policy-map** • **show access-list** • **show class-map** • **show table-map** • **show auto-qos** • **show policy-map interface** • **show ip access-lists** 这些命令中都有关键字“**AutoQos-**”

接下来做什么？

要想禁用 Auto-QoS 集合，用户需要通过 **no** 格式的 Auto-QoS 命令，删除所有接口上的 Auto-QoS 实例，然后在全局配置模式中输入命令 **no auto qos global compact**。

监控 Auto-QoS

表 86：监控 Auto-QoS 的命令

命令	**描述**
show auto qos [**interface** [*interface-id*]]	显示初始的 Auto-QoS 配置。 用户可以对比命令 **show auto qos** 和命令 **show running-config** 的输出内容，来找出哪些是用户定义的 QoS 设置。
show running-config	显示有可能受到 Auto-QoS 影响的 QoS 配置信息。 用户可以对比命令 **show auto qos** 和命令 **show running-config** 的输出内容，来找出哪些是用户定义的 QoS 设置。
show derived-config	显示隐藏的 **mls qos** 命令，这是由于 Auto-QoS 模版，配置在运行配置中的。

QoS 的排错

为了对 Auto-QoS 进行排错，用户需要使用特权 EXEC 命令 **debug auto qos**。更多信息用户可

以参考这个版本设备的命令参考手册中，有关命令 **debug auto qos** 的内容。

要想在一个端口上禁用 Auto-QoS，用户需要在接口配置模式中，使用 **no** 格式的 **auto qos** 命令，比如 **no auto qos voip**。这时只有 Auto-QoS 为这个端口生成的接口配置会被移除。如果这是最后一个启用了 Auto-QoS 的端口，并且用户输入了命令 **no auto qos voip**，交换机也会认为 Auto-QoS 已禁用，即使 Auto-QoS 生成的全局配置命令还在（避免影响其他与全局配置相关的端口上的流量）。

Auto-QoS 的配置示例

示例：auto qos trust cos

以下示例展示了 **auto qos trust cos** 命令的用法，并且应用了 policy-map 和 class-map。

用户在使用这条命令时，还创建并应用了以下 policy-map：

- AutoQos-4.0-Trust-Cos-Input-Policy
- AutoQos-4.0-Output-Policy

用户在使用这条命令时，还创建并应用了以下 class-map：

- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface gigabitEthernet1/0/17
Device(config-if)# auto qos trust cos
Device(config-if)# end
Device# show policy-map interface GigabitEthernet1/0/17
GigabitEthernet1/0/7
Service-policy input: AutoQos-4.0-Trust-Cos-Input-Policy
Class-map: class-default (match-any)
```

```
0 packets
Match: any
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
cos cos table AutoQos-4.0-Trust-Cos-Table
Service-policy output: AutoQos-4.0-Output-Policy
queue stats for all priority classes:
Queueing
priority level 1
(total drops) 0
(bytes output) 0
Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
0 packets
Match: dscp cs4 (32) cs5 (40) ef (46)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 5
0 packets, 0 bytes
5 minute rate 0 bps
Priority: 30% (300000 kbps), burst bytes 7500000,
Priority Level: 1
Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
0 packets
Match: dscp cs2 (16) cs3 (24) cs6 (48) cs7 (56)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 3
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
queue-limit dscp 16 percent 80
queue-limit dscp 24 percent 90
```

```
queue-limit dscp 48 percent 100
queue-limit dscp 56 percent 100
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
0 packets
Match: dscp af41 (34) af42 (36) af43 (38)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 4
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
0 packets
Match: dscp af21 (18) af22 (20) af23 (22)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 2
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
```

```
0 packets
Match: dscp af11 (10) af12 (12) af13 (14)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 1
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 4%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
0 packets
Match: dscp cs1 (8)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 1%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
0 packets
Match: dscp af31 (26) af32 (28) af33 (30)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: class-default (match-any)
```

```
0 packets
Match: any
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 25%
queue-buffers ratio 25
```

示例：auto qos trust dscp

以下示例展示了 **auto qos trust dscp** 命令的用法，并且应用了 policy-map 和 class-map。

用户在使用这条命令时，还创建并应用了以下 policy-map：

- AutoQos-4.0-Trust-Dscp-Input-Policy
- AutoQos-4.0-Output-Policy

用户在使用这条命令时，还创建并应用了以下 class-map：

- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface GigabitEthernet1/0/18
Device(config-if)# auto qos trust dscp
Device(config-if)# end
Device#show policy-map interface GigabitEthernet1/0/18
GigabitEthernet1/0/18
Service-policy input: AutoQos-4.0-Trust-Dscp-Input-Policy
Class-map: class-default (match-any)
0 packets
```

```
Match: any
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp dscp table AutoQos-4.0-Trust-Dscp-Table
Service-policy output: AutoQos-4.0-Output-Policy
queue stats for all priority classes:
Queueing
priority level 1
(total drops) 0
(bytes output) 0
Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
0 packets
Match: dscp cs4 (32) cs5 (40) ef (46)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 5
0 packets, 0 bytes
5 minute rate 0 bps
Priority: 30% (300000 kbps), burst bytes 7500000,
Priority Level: 1
Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
0 packets
Match: dscp cs2 (16) cs3 (24) cs6 (48) cs7 (56)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 3
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
queue-limit dscp 16 percent 80
queue-limit dscp 24 percent 90
queue-limit dscp 48 percent 100
```

```
queue-limit dscp 56 percent 100
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
0 packets
Match: dscp af41 (34) af42 (36) af43 (38)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 4
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
0 packets
Match: dscp af21 (18) af22 (20) af23 (22)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 2
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
0 packets
```

```
Match: dscp af11 (10) af12 (12) af13 (14)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 1
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 4%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
0 packets
Match: dscp cs1 (8)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 1%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
0 packets
Match: dscp af31 (26) af32 (28) af33 (30)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: class-default (match-any)
0 packets
```

```
Match: any
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 25%
queue-buffers ratio 25
```

示例：auto qos video cts

以下示例展示了 **auto qos video cts** 命令的用法，并且应用了 policy-map 和 class-map。

用户在使用这条命令时，还创建并应用了以下 policy-map：

- AutoQos-4.0-Trust-Cos-Input-Policy
- AutoQos-4.0-Output-Policy

用户在使用这条命令时，还创建并应用了以下 class-map：

- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface gigabitEthernet1/0/12
Device(config-if)# auto qos video cts
Device(config-if)# end
Device# show policy-map interface gigabitEthernet1/0/12
GigabitEthernet1/0/12
Service-policy input: AutoQos-4.0-Trust-Cos-Input-Policy
Class-map: class-default (match-any)
0 packets
```

```
Match: any
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
cos cos table AutoQos-4.0-Trust-Cos-Table
Service-policy output: AutoQos-4.0-Output-Policy
queue stats for all priority classes:
Queueing
priority level 1
(total drops) 0
(bytes output) 0
Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
0 packets
Match: dscp cs4 (32) cs5 (40) ef (46)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 5
0 packets, 0 bytes
5 minute rate 0 bps
Priority: 30% (300000 kbps), burst bytes 7500000,
Priority Level: 1
Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
0 packets
Match: dscp cs2 (16) cs3 (24) cs6 (48) cs7 (56)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 3
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
queue-limit dscp 16 percent 80
queue-limit dscp 24 percent 90
queue-limit dscp 48 percent 100
```

```
queue-limit dscp 56 percent 100
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
0 packets
Match: dscp af41 (34) af42 (36) af43 (38)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 4
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
0 packets
Match: dscp af21 (18) af22 (20) af23 (22)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 2
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
0 packets
```

```
Match: dscp af11 (10) af12 (12) af13 (14)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 1
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 4%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
0 packets
Match: dscp cs1 (8)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 1%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
0 packets
Match: dscp af31 (26) af32 (28) af33 (30)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: class-default (match-any)
0 packets
```

```
Match: any
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 25%
queue-buffers ratio 25
```

示例：auto qos video ip-camera

以下示例展示了 **auto qos video ip-camera** 命令的用法，并且应用了 policy-map 和 class-map。

用户在使用这条命令时，还创建并应用了以下 policy-map：

- AutoQos-4.0-Trust-Dscp-Input-Policy
- AutoQos-4.0-Output-Policy

用户在使用这条命令时，还创建并应用了以下 class-map：

- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface GigabitEthernet1/0/9
Device(config-if)# auto qos video ip-camera
Device(config-if)# end
Device# show policy-map interface GigabitEthernet1/0/9
GigabitEthernet1/0/9
Service-policy input: AutoQos-4.0-Trust-Dscp-Input-Policy
Class-map: class-default (match-any)
0 packets
Match: any
```

```
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp dscp table AutoQos-4.0-Trust-Dscp-Table
Service-policy output: AutoQos-4.0-Output-Policy
queue stats for all priority classes:
Queueing
priority level 1
(total drops) 0
(bytes output) 0
Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
0 packets
Match: dscp cs4 (32) cs5 (40) ef (46)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 5
0 packets, 0 bytes
5 minute rate 0 bps
Priority: 30% (300000 kbps), burst bytes 7500000,
Priority Level: 1
Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
0 packets
Match: dscp cs2 (16) cs3 (24) cs6 (48) cs7 (56)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 3
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
queue-limit dscp 16 percent 80
queue-limit dscp 24 percent 90
queue-limit dscp 48 percent 100
queue-limit dscp 56 percent 100
```

```
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
0 packets
Match: dscp af41 (34) af42 (36) af43 (38)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 4
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
0 packets
Match: dscp af21 (18) af22 (20) af23 (22)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 2
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
0 packets
Match: dscp af11 (10) af12 (12) af13 (14)
```

```
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 1
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 4%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
0 packets
Match: dscp cs1 (8)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 1%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
0 packets
Match: dscp af31 (26) af32 (28) af33 (30)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: class-default (match-any)
0 packets
Match: any
```

```
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 25%
queue-buffers ratio 25
```

示例：auto qos video media-player

以下示例展示了 **auto qos video media-player** 命令的用法，并且应用了 policy-map 和 class-map。

用户在使用这条命令时，还创建并应用了以下 policy-map：

- AutoQos-4.0-Trust-Dscp-Input-Policy
- AutoQos-4.0-Output-Policy

用户在使用这条命令时，还创建并应用了以下 class-map：

- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface GigabitEthernet1/0/25
Device(config-if)# auto qos video media-player
Device(config-if)# end
Device# show policy-map interface GigabitEthernet1/0/25
GigabitEthernet1/0/25
Service-policy input: AutoQos-4.0-Trust-Dscp-Input-Policy
Class-map: class-default (match-any)
0 packets
Match: any
0 packets, 0 bytes
```

```
5 minute rate 0 bps
QoS Set
dscp dscp table AutoQos-4.0-Trust-Dscp-Table
Service-policy output: AutoQos-4.0-Output-Policy
queue stats for all priority classes:
Queueing
priority level 1
(total drops) 0
(bytes output) 0
Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
0 packets
Match: dscp cs4 (32) cs5 (40) ef (46)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 5
0 packets, 0 bytes
5 minute rate 0 bps
Priority: 30% (300000 kbps), burst bytes 7500000,
Priority Level: 1
Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
0 packets
Match: dscp cs2 (16) cs3 (24) cs6 (48) cs7 (56)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 3
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
queue-limit dscp 16 percent 80
queue-limit dscp 24 percent 90
queue-limit dscp 48 percent 100
queue-limit dscp 56 percent 100
(total drops) 0
```

```
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
0 packets
Match: dscp af41 (34) af42 (36) af43 (38)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 4
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
0 packets
Match: dscp af21 (18) af22 (20) af23 (22)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 2
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
0 packets
Match: dscp af11 (10) af12 (12) af13 (14)
0 packets, 0 bytes
```

```
5 minute rate 0 bps
Match: cos 1
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 4%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
0 packets
Match: dscp cs1 (8)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 1%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
0 packets
Match: dscp af31 (26) af32 (28) af33 (30)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: class-default (match-any)
0 packets
Match: any
0 packets, 0 bytes
```

```
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 25%
queue-buffers ratio 25
```

示例：auto qos voip trust

以下示例展示了 **auto qos voip trust** 命令的用法，并且应用了 policy-map 和 class-map。

用户在使用这条命令时，还创建并应用了以下 policy-map：

- AutoQos-4.0-Trust-Cos-Input-Policy
- AutoQos-4.0-Output-Policy

用户在使用这条命令时，还创建并应用了以下 class-map：

- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface gigabitEthernet1/0/31
Device(config-if)# auto qos voip trust
Device(config-if)# end
Device# show policy-map interface GigabitEthernet1/0/31
GigabitEthernet1/0/31
Service-policy input: AutoQos-4.0-Trust-Cos-Input-Policy
Class-map: class-default (match-any)
0 packets
Match: any
0 packets, 0 bytes
5 minute rate 0 bps
```

QoS Set

cos cos table AutoQos-4.0-Trust-Cos-Table

Service-policy output: AutoQos-4.0-Output-Policy

Queueing

priority level 1

(total drops) 0

(bytes output) 0

Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)

0 packets

Match: dscp cs4 (32) cs5 (40) ef (46)

0 packets, 0 bytes

5 minute rate 0 bps

Match: cos 5

0 packets, 0 bytes

5 minute rate 0 bps

Priority: 30% (300000 kbps), burst bytes 7500000,

Priority Level: 1

Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)

0 packets

Match: dscp cs2 (16) cs3 (24) cs6 (48) cs7 (56)

0 packets, 0 bytes

5 minute rate 0 bps

Match: cos 3

0 packets, 0 bytes

5 minute rate 0 bps

Queueing

queue-limit dscp 16 percent 80

queue-limit dscp 24 percent 90

queue-limit dscp 48 percent 100

queue-limit dscp 56 percent 100

(total drops) 0

(bytes output) 0

bandwidth remaining 10%

```
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any
0 packets
Match: dscp af41 (34) af42 (36) af43 (38)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 4
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
0 packets
Match: dscp af21 (18) af22 (20) af23 (22)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 2
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
0 packets
Match: dscp af11 (10) af12 (12) af13 (14)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 1
```

```
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 4%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
0 packets
Match: dscp cs1 (8)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 1%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any
0 packets
Match: dscp af31 (26) af32 (28) af33 (30)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: class-default (match-any)
0 packets
Match: any
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
```

```
(total drops) 0
(bytes output) 0
bandwidth remaining 25%
queue-buffers ratio 25
```

示例：auto qos voip inspur-phone

以下示例展示了 **auto qos voip inspur-phone** 命令的用法，并且应用了 policy-map 和 class-map。

用户在使用这条命令时，还创建并应用了以下 policy-map：

- AutoQos-4.0-InspurPhone-Input-Policy
- AutoQos-4.0-Output-Policy

用户在使用这条命令时，还创建并应用了以下 class-map：

- AutoQos-4.0-Voip-Data-InspurPhone-Class（match-any）
- AutoQos-4.0-Voip-Signal-InspurPhone-Class（match-any）
- AutoQos-4.0-Default-Class（match-any）
- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface gigabitEthernet1/0/5
Device(config-if)# auto qos voip inspur-phone
Device(config-if)# end
Device# show policy-map interface gigabitEthernet1/0/5
GigabitEthernet1/0/5
Service-policy input: AutoQos-4.0-InspurPhone-Input-Policy
Class-map: AutoQos-4.0-Voip-Data-InspurPhone-Class (match-any)
0 packets
Match: cos 5
0 packets, 0 bytes
```

```
5 minute rate 0 bps
QoS Set
dscp ef
police:
cir 128000 bps, bc 8000 bytes
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
set-dscp-transmit dscp table policed-dscp
conformed 0000 bps, exceed 0000 bps
Class-map: AutoQos-4.0-Voip-Signal-InspurPhone-Class (match-any)
0 packets
Match: cos 3
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp cs3
police:
cir 32000 bps, bc 8000 bytes
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
set-dscp-transmit dscp table policed-dscp
conformed 0000 bps, exceed 0000 bps
Class-map: AutoQos-4.0-Default-Class (match-any)
0 packets
Match: access-group name AutoQos-4.0-Acl-Default
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp default
Class-map: class-default (match-any)
0 packets
```

```
Match: any
0 packets, 0 bytes
5 minute rate 0 bps
Service-policy output: AutoQos-4.0-Output-Policy
queue stats for all priority classes:
Queueing
priority level 1
(total drops) 0
(bytes output) 0
Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
0 packets
Match: dscp cs4 (32) cs5 (40) ef (46)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 5
0 packets, 0 bytes
5 minute rate 0 bps
Priority: 30% (300000 kbps), burst bytes 7500000,
Priority Level: 1
Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
0 packets
Match: dscp cs2 (16) cs3 (24) cs6 (48) cs7 (56)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 3
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
queue-limit dscp 16 percent 80
queue-limit dscp 24 percent 90
queue-limit dscp 48 percent 100
queue-limit dscp 56 percent 100
(total drops) 0
```

```
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
0 packets
Match: dscp af41 (34) af42 (36) af43 (38)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 4
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
0 packets
Match: dscp af21 (18) af22 (20) af23 (22)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 2
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
0 packets
Match: dscp af11 (10) af12 (12) af13 (14)
0 packets, 0 bytes
```

```
5 minute rate 0 bps
Match: cos 1
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 4%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
0 packets
Match: dscp cs1 (8)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 1%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
0 packets
Match: dscp af31 (26) af32 (28) af33 (30)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: class-default (match-any)
0 packets
Match: any
0 packets, 0 bytes
```

```
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 25%
queue-buffers ratio 25
```

示例：auto qos voip inspur-softphone

以下示例展示了 **auto qos voip inspur-softphone** 命令的用法，并且应用了 policy-map 和 class-map。

用户在使用这条命令时，还创建并应用了以下 policy-map：

- AutoQos-4.0-InspurSoftPhone-Input-Policy
- AutoQos-4.0-Output-Policy

用户在使用这条命令时，还创建并应用了以下 class-map：

- AutoQos-4.0-Voip-Data- Class（match-any）
- AutoQos-4.0-Voip-Signal-Class（match-any）
- AutoQos-4.0-Multimedia-Conf-Class（match-any）
- AutoQos-4.0-Bulk-Data-Class（match-any）
- AutoQos-4.0-Transaction-Class（match-any）
- AutoQos-4.0-Scavanger-Class（match-any）
- AutoQos-4.0-Signaling-Class（match-any）
- AutoQos-4.0-Default-Class（match-any）
- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface gigabitEthernet1/0/21
Device(config-if)# auto qos voip inspursoftphone
```

```
Device(config-if)# end
Device# show policy-map interface gigabitEthernet1/0/21
GigabitEthernet1/0/21
Service-policy input: AutoQos-4.0-InspurSoftPhone-Input-
Policy Class-map: AutoQos-4.0-Voip-Data-Class (match-any)
0 packets
Match: dscp ef (46)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 5
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp ef
police:
cir 128000 bps, bc 8000 bytes
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
set-dscp-transmit dscp table policed-dscp
conformed 0000 bps, exceed 0000 bps
Class-map: AutoQos-4.0-Voip-Signal-Class (match-any)
0 packets
Match: dscp cs3 (24)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 3
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp cs3
police:
cir 32000 bps, bc 8000 bytes
```

```
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
set-dscp-transmit dscp table policed-dscp
conformed 0000 bps, exceed 0000 bps
Class-map: AutoQos-4.0-Multimedia-Conf-Class (match-any)
0 packets
Match: access-group name AutoQos-4.0-Acl-MultiEnhanced-Conf
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp af41
police:
cir 5000000 bps, bc 156250 bytes
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
drop
conformed 0000 bps, exceed 0000 bps
Class-map: AutoQos-4.0-Bulk-Data-Class (match-any)
0 packets
Match: access-group name AutoQos-4.0-Acl-Bulk-Data
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp af11
police:
cir 10000000 bps, bc 312500 bytes
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
set-dscp-transmit dscp table policed-dscp
conformed 0000 bps, exceed 0000 bps
```

```
Class-map: AutoQos-4.0-Transaction-Class (match-any)
0 packets
Match: access-group name AutoQos-4.0-Acl-Transactional-Data
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp af21
police:
cir 10000000 bps, bc 312500 bytes
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
set-dscp-transmit dscp table policed-dscp
conformed 0000 bps, exceed 0000 bps
Class-map: AutoQos-4.0-Scavanger-Class (match-any)
0 packets
Match: access-group name AutoQos-4.0-Acl-Scavanger
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp cs1
police:
cir 10000000 bps, bc 312500 bytes
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
drop
conformed 0000 bps, exceed 0000 bps
Class-map: AutoQos-4.0-Signaling-Class (match-any)
0 packets
Match: access-group name AutoQos-4.0-Acl-Signaling
0 packets, 0 bytes
5 minute rate 0 bps
```

```
QoS Set
dscp cs3
police:
cir 32000 bps, bc 8000 bytes
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
drop
conformed 0000 bps, exceed 0000 bps
Class-map: AutoQos-4.0-Default-Class (match-any)
0 packets
Match: access-group name AutoQos-4.0-Acl-Default
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp default
police:
cir 10000000 bps, bc 312500 bytes
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
set-dscp-transmit dscp table policed-dscp
conformed 0000 bps, exceed 0000 bps
Class-map: class-default (match-any)
0 packets
Match: any
0 packets, 0 bytes
5 minute rate 0 bps
Service-policy output: AutoQos-4.0-Output-Policy
queue stats for all priority classes:
Queueing
priority level 1
(total drops) 0
```

```
(bytes output) 0
Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
0 packets
Match: dscp cs4 (32) cs5 (40) ef (46)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 5
0 packets, 0 bytes
5 minute rate 0 bps
Priority: 30% (300000 kbps), burst bytes 7500000,
Priority Level: 1
Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
0 packets
Match: dscp cs2 (16) cs3 (24) cs6 (48) cs7 (56)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 3
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
queue-limit dscp 16 percent 80
queue-limit dscp 24 percent 90
queue-limit dscp 48 percent 100
queue-limit dscp 56 percent 100
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
0 packets
Match: dscp af41 (34) af42 (36) af43 (38)
0 packets, 0 bytes
5 minute rate 0 bps
```

```
Match: cos 4
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
0 packets
Match: dscp af21 (18) af22 (20) af23 (22)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 2
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
0 packets
Match: dscp af11 (10) af12 (12) af13 (14)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 1
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 4%
```

```
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
0 packets
Match: dscp cs1 (8)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 1%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
0 packets
Match: dscp af31 (26) af32 (28) af33 (30)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: class-default (match-any)
0 packets
Match: any
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 25%
queue-buffers ratio 25
```

auto qos classify police

以下示例展示了 **auto qos classify police** 命令的用法，并且应用了 policy-map 和 class-map。

用户在使用这条命令时，还创建并应用了以下 policy-map：

- AutoQos-4.0-Classify-Police-Input-Policy
- AutoQos-4.0-Output-Policy

用户在使用这条命令时，还创建并应用了以下 class-map：

- AutoQos-4.0-Multimedia-Conf-Class（match-any）
- AutoQos-4.0-Bulk-Data-Class（match-any）
- AutoQos-4.0-Transaction-Class（match-any）
- AutoQos-4.0-Scavanger-Class（match-any）
- AutoQos-4.0-Signaling-Class（match-any）
- AutoQos-4.0-Default-Class（match-any）
- class-default（match-any）
- AutoQos-4.0-Output-Priority-Queue（match-any）
- AutoQos-4.0-Output-Control-Mgmt-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Conf-Queue（match-any）
- AutoQos-4.0-Output-Trans-Data-Queue（match-any）
- AutoQos-4.0-Output-Bulk-Data-Queue（match-any）
- AutoQos-4.0-Output-Scavenger-Queue（match-any）
- AutoQos-4.0-Output-Multimedia-Strm-Queue（match-any）

```
Device(config)# interface gigabitEthernet1/0/6
Device(config-if)# auto qos classify police
Device(config-if)# end
Device# show policy-map interface gigabitEthernet1/0/6
GigabitEthernet1/0/6
Service-policy input: AutoQos-4.0-Classify-Police-Input-Policy
Class-map: AutoQos-4.0-Multimedia-Conf-Class (match-any)
0 packets
Match: access-group name AutoQos-4.0-Acl-MultiEnhanced-Conf
0 packets, 0 bytes
5 minute rate 0 bps
```

```
QoS Set
dscp af41
police:
cir 5000000 bps, bc 156250 bytes
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
drop
conformed 0000 bps, exceed 0000 bps
Class-map: AutoQos-4.0-Bulk-Data-Class (match-any)
0 packets
Match: access-group name AutoQos-4.0-Acl-Bulk-Data
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp af11
police:
cir 10000000 bps, bc 312500 bytes
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
set-dscp-transmit dscp table policed-dscp
conformed 0000 bps, exceed 0000 bps
Class-map: AutoQos-4.0-Transaction-Class (match-any)
0 packets
Match: access-group name AutoQos-4.0-Acl-Transactional-Data
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp af21
police:
cir 10000000 bps, bc 312500 bytes
conformed 0 bytes; actions:
```

```
transmit
exceeded 0 bytes; actions:
set-dscp-transmit dscp table policed-dscp
conformed 0000 bps, exceed 0000 bps
Class-map: AutoQos-4.0-Scavanger-Class (match-any)
0 packets
Match: access-group name AutoQos-4.0-Acl-Scavanger
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp cs1
police:
cir 10000000 bps, bc 312500 bytes
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
drop
conformed 0000 bps, exceed 0000 bps
Class-map: AutoQos-4.0-Signaling-Class (match-any)
0 packets
Match: access-group name AutoQos-4.0-Acl-Signaling
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp cs3
police:
cir 32000 bps, bc 8000 bytes
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
drop
conformed 0000 bps, exceed 0000 bps
Class-map: AutoQos-4.0-Default-Class (match-any)
```

```
0 packets
Match: access-group name AutoQos-4.0-Acl-Default
0 packets, 0 bytes
5 minute rate 0 bps
QoS Set
dscp default
police:
cir 10000000 bps, bc 312500 bytes
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
set-dscp-transmit dscp table policed-dscp
conformed 0000 bps, exceed 0000 bps
Class-map: class-default (match-any)
0 packets
Match: any
0 packets, 0 bytes
5 minute rate 0 bps
Service-policy output: AutoQos-4.0-Output-Policy
queue stats for all priority classes:
Queueing
priority level 1
(total drops) 0
(bytes output) 0
Class-map: AutoQos-4.0-Output-Priority-Queue (match-any)
0 packets
Match: dscp cs4 (32) cs5 (40) ef (46)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 5
0 packets, 0 bytes
5 minute rate 0 bps
Priority: 30% (300000 kbps), burst bytes 7500000,
```

```
Priority Level: 1
Class-map: AutoQos-4.0-Output-Control-Mgmt-Queue (match-any)
0 packets
Match: dscp cs2 (16) cs3 (24) cs6 (48) cs7 (56)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 3
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
queue-limit dscp 16 percent 80
queue-limit dscp 24 percent 90
queue-limit dscp 48 percent 100
queue-limit dscp 56 percent 100
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Conf-Queue (match-any)
0 packets
Match: dscp af41 (34) af42 (36) af43 (38)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 4
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Trans-Data-Queue (match-any)
0 packets
```

```
Match: dscp af21 (18) af22 (20) af23 (22)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 2
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Bulk-Data-Queue (match-any)
0 packets
Match: dscp af11 (10) af12 (12) af13 (14)
0 packets, 0 bytes
5 minute rate 0 bps
Match: cos 1
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 4%
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Scavenger-Queue (match-any)
0 packets
Match: dscp cs1 (8)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 1%
```

```
queue-buffers ratio 10
Class-map: AutoQos-4.0-Output-Multimedia-Strm-Queue (match-any)
0 packets
Match: dscp af31 (26) af32 (28) af33 (30)
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 10%
queue-buffers ratio 10
Class-map: class-default (match-any)
0 packets
Match: any
0 packets, 0 bytes
5 minute rate 0 bps
Queueing
(total drops) 0
(bytes output) 0
bandwidth remaining 25%
queue-buffers ratio 25
```

auto qos global compact

这个示例展示了命令 **auto qos global compact** 的用法。

```
Device# configure terminal
Device(config)# auto qos global compact
Device(config)# interface GigabitEthernet1/2
Device(config-if)# auto qos voip inspurphone
Device# show auto-qos
GigabitEthernet1/2
auto qos voip inspur-phone
Device# show running-config interface GigabitEthernet 1/0/2
```

```
interface GigabitEthernet1/0/2
auto qos voip inspur-phone
end
```

配置 Auto-QoS 之后的操作

如果用户需要在自己的 Auto-QoS 配置中进行变更的话，重新看看 QoS 文档。

Auto-QoS 的其他参考资料

相关文档

相关主题	文档名称
本章中命令的完整语法和用法信息	*QoS Command Reference (Inspur 6650 Switches)* *Inspur INOS Quality of Service Solutions Command Reference*

错误消息解码器

描述	链接
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

标准和 RFC

标准/RFC	标题
无	-

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术	http://www.icntnetworks.com

信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	

Auto-QoS 的特性历史与信息

版本	变更
Inspur INOS 11.3.1	引入该特性

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置服务质量的先决条件

在开始配置标准 QoS 之前，用户必须充分理解以下内容：

- 标准 QoS 的概念；
- 经典 Inspur INOS QoS；

- 模块化 QoS CLI（MQC）；
- 理解 QoS 的实施；
- 用户网络中使用的应用类型和流量模式；
- 用户网络的流量特征和需求。比如网络中的流量是否具有突发性？是否需要为语音和视频流预留带宽？
- 网络中的带宽和速率需求；
- 网络中的拥塞点位置。

QoS 的组成部分

服务质量（QoS）由以下重要部分构组成：

- 分类——分类是区分流量类型的过程，用户可以基于访问控制列表（ACL）、差分服务代码点（DSCP）、服务类别（CoS）和其他因素进行区分；
- 标记和突变——在流量上设置标记是为了向网络中的下游设备传达特定的信息，或者把信息从一个接口传递给另一个接口。当流量被打上标记后，设备就可以针对这个流量应用 QoS 操作行为了。用户可以直接使用 **set** 命令来设置 QoS 行为，也可以通过 table-map 来进行设置，table-map 可以查看入站标记值，然后直接把入站标记值转换为出站标记值；
- 整形和限速——整形是指控制流量最大速率的过程，以防止下游设备遭到拥塞为目的，来调节流量速率。整形最常见的用法是对物理接口或逻辑接口发送的流量进行限制。限速是指对一个流量类别设定最大速率。如果超出限速了，QoS 马上会对相关流量执行相应的行为；
- 列队——列队的作用是为了防止流量拥塞。根据带宽的分配方式，QoS 会把流量分类到不同的队列中，来提供服务和调度。然后流量会接受调度，或者从端口发送出去；
- 带宽——带宽的分配方式决定了受到 QoS 策略影响的流量所能够使用的带宽容量；
- 受信——受信功能能够使流量通过交换机，并在用户没有明确指定策略配置时，保留终端携带的差分服务代码点（DSCP）、优先级或 CoS 值。

QoS 术语

在这个 QoS 配置指南中，会替换使用以下术语：

- “上游”（去往一个方向）与“入向”会替换使用；
- “下游”（从一个方向来）与“出向”会替换使用

注释：

QoS 的相关信息

QoS 概述

通过配置服务质量（QoS），用户可以以降低其他流量类型服务质量为代价，为特定类型的流量提供更好的服务。如果没有 QoS 的话，设备会对每个数据包提供尽力而为的服务，而完全不管数据包的内容或大小。设备在发送数据包的时候不提供任何可靠性保障、延迟保障，或吞吐量保障。

QoS 提供了以下特性：

- 低延迟
- 带宽保障
- 缓存空间和丢包规划
- 流量限速
- 更改数据帧或数据包头部的能力
- 相关服务

模块化 QoS 命令行界面

在设备上，QoS 特性是通过模块化 QoS 命令行界面（MQC）启用的。MQC 是一种命令行界面（CLI）结构，让用户能够创建流量策略，并将其应用在接口上。流量策略中包含流量类别，以及一个或多个 QoS 特性。流量类别是用来对流量进行分类的，之后流量策略中的 QoS 特性就能够确定如何处理这些分类后的流量。

MQC 的一大主要目标是提供与平台无关的界面，使用户在 Inspur 平台上能够使用统一的命令结构来配置 QoS。

层级式 QoS

设备能够支持层级式 QoS（HQoS）。通过使用 HQoS，用户能够实施：

- 层级式分类——根据其他类别对流量进行分类；
- 层级式限速——在层级式策略中，拥有多个等级的限速配置；
- 层级式整形——在层级式策略中，整形也可以被配置为多个等级。

注释： 只有端口整形器支持层级式整形，对于父系交换机来说，用户只能实施 class-default 配置，并且只能对 class-default 实施整形行为。

QoS 的实施

通常网络的操作行为是以尽力而为的转发为基础的，也就是说所有流量拥有相同的优先级，并且有相同的机会能够得到及时传递。在拥塞发生时，所有流量被丢弃的几率也是相同的。在用户配置 QoS 特性时可以选择指定的网络流量，根据它的重要性调整优先级，并使用拥塞管理和拥塞避免技术为其提供特殊服务。在网络中实施 QoS 能够使网络行为便得更加可以预测，并且使带宽的利用率更为高效。

QoS 的实施是以差分服务（Diff-Serv）架构为基础的，差分服务是 Internet 工程任务组（IETF）提出的一种标准。这个架构明确了每个数据包要在进入网络的时候进行分类。

IP 数据包头部中携带着分类结果，也就是使用长度为 6 比特已弃用的 IP 服务类型（ToS）字段，来携带分类（*类别*）信息。二层数据帧中也同样携带分类信息。

二层数据帧或三层数据包中的 QoS 比特如下图所示：

图 87：数据帧和数据包中的 QoS 分类层

Encapsulated Packet	封装的数据包
Layer 2 header	二层头部
IP header	IP 头部
Data（共 3 处）	数据
Layer 2 ISL Frame	二层 ISL 数据帧
ISL header (26 bytes)	ISL 头部 （26 字节）
Encapsulated frame 1 … (24.5 KB)	封装的数据帧 1…… （24.5 KB）

FCS (4 bytes)	FCS （4 字节）
3 bits used for CoS	3 比特用于 CoS
Layer 2 802.1Q and 802.1p Frame	二层 802.1Q 和 802.1q 数据帧
Preamble	前导码
Start frame delimiter	数据帧开始 分隔符
3 bits used for CoS (user priority)	3 比特用于 CoS（用户优先级）
Layer 3 IPv4 Packet	三层 IPv4 数据包
Version length	版本 长度
ToS (1 byte)	ToS （1 字节）
Len	长度
Offset	偏移
Proto	协议
IP precedence or DSCP（共 2 处）	IP 优先级或 DSCP
Layer 3 IPv6 Packet	三层 IPv6 数据包
Version	版本
Traffic class (1 byte)	流量类别 （1 字节）
Flow label	流 标签
Payload length	负载 长度
Next header	下一个 头部
HOP limit	HOP 限制
Source address	源 地址
Dest.	目的

address	地址

二层数据帧的优先级位

二层交换机间链路（ISL）数据帧头部有一个 1 字节的用户字段，其中的 3 个最低有效位用来标记 IEEE 802.1p 服务类别（CoS）值。在配置为二层 ISL 协议的 Trunk 端口上，所有流量都承载在 ISL 数据帧中。

二层 802.1Q 数据帧头部有一个 2 字节的标记控制信息（TCI）字段，其中的 3 个最高有效位用来标记 CoS 值，这 3 个比特也称为用户优先级位。在配置为二层 802.1Q 协议的 Trunk 端口上，除了本征 VLAN（Native VLAN）中的流量外，所有流量都承载在 802.1Q 数据帧中。其他类型的数据帧中不携带二层 CoS 值。

二层 CoS 值的取值范围是从 0 至 7，其中 0 表示最低优先级，7 表示最高优先级。

三层数据包的优先级位

三层 IP 数据包中携带着 IP 优先级值，或者差分服务代码点（DSCP）值。QoS 能够支持使用这两个值，因为 DSCP 值能够向后兼容 IP 优先级值。

IP 优先级值的取值范围是 0 至 7。DSCP 值的取值范围是 0 至 63。

使用分类的端到端 QoS 解决方案

所有接入到 Internet 中的交换机和路由器都依赖类别信息，来为拥有相同类别信息的数据包提供相同的转发行为，为拥有不同类别信息的数据包提供不同的转发行为。数据包中携带的类别信息是基于用户配置的策略和/或设备对于数据包的详细检查做出的，可以由终端用户进行分配，也可以由沿途经过的交换机或路由器进行分配。对于数据包的详细检查行为一般会在靠近网络边缘的位置上执行，这样做不会增加核心交换机和路由器的负担。

数据包传输路径中的交换机和路由器可以使用类别信息来限制某个流量类别所占用的资源总量。在 Diff-Serv 架构中，单台设备对流量的处理行为称作逐跳行为。如果路径中的所有设备都提供统一的逐跳行为，用户就可以构建出端到端的 QoS 解决方案。

在网络中实施 QoS 可以是一项简单的任务，也可以是一项复杂的任务，这取决于联网设备所提供的 QoS 特性、网络中的流量类型和流量模式，以及用户对于入站和出站流量控制的粒度。

数据包分类

数据包分类是指按照确定的规则，把数据包归类为用户定义策略中的某个类别。模块化 QoS CLI（MQC）是一种基于策略分类的语言，策略分类语言能够用来定义一下内容：

- class-map 模版，其中指定一个或几个匹配条件
- policy-map 模版，其中关联一个或几个类别

policy-map 模版之后会关联到交换机的一个或多个接口上。

数据包分类是识别数据包的过程，最终会确定数据包属于 policy-map 中定义的某一个类别。当设备发现数据包与某个类别中指定的过滤器相匹配，这个分类过程就结束了。这也称为第一匹配。如果数据包与策略中的多个类别都匹配，不管 policy-map 中的类别顺序是如何定义的，设备都会在数据包匹配到第一个类别后结束分类过程。

如果数据包与策略中的每个分类都不匹配，它就会被分类为策略中的默认类别中。每个 policy-map 中都有一个默认类别，这是系统定义的类别，会匹配所有与用户定义的类别不相匹配的数据包。

数据包分类特性可以归类为以下类型：

- 根据随数据包传播的信息进行分类
- 根据特定信息进行分类
- 层级式分类

根据随数据包传播的信息进行分类

这种分类方式会基于数据包中的某部分信息进行分类，并且这些信息会随着数据包端到端传输，或者在一些中间设备之间传输，通常这种分类方式会使用以下信息：

- 根据三层或四层头部进行分类
- 根据二层信息进行分类

根据三层或四层头部进行分类

这是最常见的部署环境。三层和四层头部中有很多字段都可以用来进行数据包分类。

以最精细的级别来说，分类技术可以匹配完整的流。对于这种部署类型来说，用户可以使用访问控制列表（ACL）。ACL 可以根据流的不同部分进行匹配（比如只匹配源 IP 地址、只匹配目的 IP 地址，或者同时匹配源和目的 IP 地址）。

用户还可以根据 IP 头部中的优先级或 DSCP 值进行数据包分类。IP 优先级字段用来标识这个处理数据包所需的优先级等级。它由 IP 头部服务类型（ToS）字节中的 3 比特构成。

下面这个表格中列出了不同的 IP 优先级值及其含义。

表 90：IP 优先级值和名称

IP 优先级值	IP 优先级比特	IP 优先级名称
0	000	Routine（普通）
1	001	Priority（优先）
2	010	Immediate（快速）
3	011	Flash（闪速）
4	100	Flash Override（疾速）
5	101	Critical（关键）
6	110	Internetwork Control（网间控制）
7	111	Network Control（网络控制）

注释： 网络中的所有路由控制流量默认都是用 IP 优先级值 6。IP 优先级值 7 也是为网络控制流量预留的。因此不建议把 IP 优先级值 6 和 7 分配给用户流量。

DSCP 字段由 IP 头部中的 6 比特构成，它是由 Internet 工程任务组（IETF）差分服务工作组进行标准化的。最初包含 DSCP 位的 ToS 字节已经被重命名为 DSCP 字节。DSCP 是 IP 头部中的字段，与 IP 优先级类似。DSCP 字段的范围比 IP 优先级字段的范围大，因此 DSCP 字段的描述方式与 IP 优先级值的描述方式类似。

注释： DSCP 字段的定义能够向后兼容 IP 优先级值。

根据二层头部进行分类

用户可以使用多种方法来基于二层头部信息执行数据包分类。最常用的方法如下所示：

- 基于 MAC 地址的分类（只用于 access-group）——基于源 MAC 地址（用来针对入站流量进行限速）和目的 MAC 地址（用来针对出站流量进行限速）进行分类；
- 服务类别——基于二层头部中的 3 比特进行分类，符合 IEEE 802.1p 标准。这个值通常与 IP 头部的 ToS 字段之间有映射关系；
- VLAN ID——基于数据包的 VLAN ID 进行分类

注释： 二层头部中的有些字段也可以通过策略进行设置。

基于设备指定的信息进行分类（QoS 组）

用户也可以不基于数据包头部负载中携带的信息进行分类。

有时用户可能需要把从多个入站接口进入的流量汇聚到一个出站接口的特定类别中。举例来说，可能会有多个客户边界路由器的流量从不同接口进入服务提供商网络并获得相同的服务。服务提供商可能希望对所有汇集起来的语音流量进行限速，使其以指定速率进入核心网。但语音流量可能来自不同的客户，也可能会携带不同的 ToS 设置。基于 QoS 组的分类特性就适用于这种环境。

用户在入站接口配置的策略可以把 QoS 组设置为指定值，然后出站接口上的策略可以使用

这个值来对数据包进行分类。

QoS 组是交换机内部的数据包数据结构中的一个字段。需要注意的是，QoS 组是交换机的内部标签，并不是数据包头部中的一部分。

层级式分类

用户可以基于其他类别来执行数据包分类。通常来说，如果用户需要把两个或多个类别中的分类机制（比如过滤器）结合到一个 class-map 中，就需要使用层级式分类。

QoS 有线模型

要想实施 QoS，用户必须执行以下任务：

- 流量分类——区分数据包或流；
- 流量标记和限速——当数据包在交换机中移动时，QoS 会为其分配一个代表指定服务质量的标签，使设备对于数据包的处理能够够符合用户配置的资源利用限制；
- 列队和调度——在存在资源竞争的环境中为不同流量提供不同服务；
- 整形——确保从交换机发来的流量符合特定的流量模型。

入站端口行为

交换机的入站端口上会发生以下行为：

- 分类——通过把数据包与 QoS 标签进行关联，为数据包分配不同的路径。举例来说，把数据包中的 CoS 或 DSCP 映射为一个 QoS 标签，用来区分不同类型的流量。QoS 标签表明了设备会对这个数据包执行的 QoS 行为；
- 限速——通过把入站流量的速率与用户配置的速率进行对比，限速特性能够确定数据包是符合限定的，还是超出了限定。限速器会限制一个流所使用的带宽。判断结果会发送给标记器；
- 标记——标记特性会根据限速器和配置信息来评估对数据包进行评估，当数据包超出限定条件后，它会评估要对数据包执行的行为，以此决定对数据包采取的具体做法（让数据包不做任何修改地通过、降低数据包中 QoS 标签的等级，或者丢弃数据包）。

出站端口行为

交换机的出站端口上会发生以下行为：

- 限速——通过把入站流量的速率与用户配置的速率进行对比，限速特性能够确定数据包是符合限定的，还是超出了限定。限速器会限制一个流所使用的带宽。判断结果会发送给标记器；
- 标记——标记特性会根据限速器和配置信息来评估对数据包进行评估，当数据包超出限定条件后，它会评估要对数据包执行的行为，以此决定对数据包采取的具体做法（让数

据包不做任何修改地通过、降低数据包中 QoS 标签的等级，或者丢弃数据包）；

- 列队——列队特性会对 QoS 数据包标签和相应的 DSCP 或 CoS 值进行评估，然后选择为数据包使用的出向队列。由于当多个入向端口同时向一个出向端口发送数据时会发生拥塞，因此列队特性会根据 QoS 标签，使用加权尾部丢弃（WTD）来区分流量类别，并为数据包分配不同的门限值。如果超出了门限值，数据包就会被丢弃。

分类

分类是通过查看数据包中的字段，来区分流量类型的过程。只有当交换机上启用了 QoS 时，才会启用分类特性。默认情况下，交换机上已启用了 QoS。

在分类过程中，交换机会先执行查找，然后为数据包分配一个 QoS 标签。QoS 标签标识了对数据包执行的所有 QoS 行为，以及应该把数据包发送到哪个队列中。

访问控制列表

用户可以使用 IP 标准、IP 扩展，或二层 MAC ACL 来定义一组拥有相同特征的数据包（类别）。用户也可以基于 IPv6 ACL 来分类 IP 流量。

在 QoS 环境中，访问控制条目（ACE）中的允许（permit）和拒绝（deny）行为与安全 ACL 中的允许和拒绝行为有所不同：

- 如果根据第一匹配原则，数据包匹配的条目中设置了允许行为，设备就会对这个数据包执行这个 QoS 行为；
- 如果数据包匹配的条目中设置了拒绝行为，数据包就会跳出这个 ACL，并由下一个 ACL 进行处理；
- 如果数据包没有匹配任何设置了允许行为的条目，并且与所有 ACE 都进行了匹配，那么这个数据包就不会接受 QoS 处理行为，交换机会对这个数据包提供尽力而为的服务；
- 如果端口上配置了多个 ACL，那么当数据包匹配第一个设置了允许行为的 ACL 时，查找就会结束，QoS 的处理行为就会开始。

注释： 在用户创建访问列表时，要注意默认情况下访问列表中都会包含隐含的拒绝条目，这个条目会在 ACL 末尾匹配所有未能与之前的条目相匹配的数据包。

在 ACL 中定义了流量类别后，用户可以为其关联一个策略。一个策略中可能会包含多个类别，并为每个类别指定不同行为。策略中可能会包含把类别分类为特定汇聚类的命令（比如分配 DSCP），或者对类别进行限速的命令。然后用户会把策略关联到一个端口上，让策略生效。

用户可以使用全局配置命令 **access-list** 配置 IP ACL，来分类 IP 流量；用户可以使用全局配置命令 **mac access-list extended** 配置二层 MAC ACL，来分类非 IP 流量。

class-map

用户可以使用 class-map 机制来为指定流量（或类别）进行命名，并从所有其他流量中把它隔离出来。class-map 定义了用来匹配指定流量的规则，以便将来对流量进行分类。在规则中，流量可以匹配 ACL 定义的 access-group，或者匹配指定的 DSCP 或 IP 优先级值。如果用户希望分类多种流量类型，就可以创建另一个 class-map，并为其使用不同的名称。当数据包匹配了一个 class-map 规则时，用户继而可以使用 policy-map 对其进行分类。

用户可以使用全局配置命令 **class-map** 来创建一个 class-map，或者使用 policy-map 配置命令 **class** 来创建一个 class-map。当会有多个端口共享一个 class-map 时，用户应该使用 **class-map** 命令。在用户输入 **class-map** 命令后，也就进入了 class-map 配置模式。在这个模式中，用户可以使用 class-map 配置命令 **match** 来为流量定义匹配规则。

用户可以通过使用 policy-map 配置命令 **class class-default** 来创建默认类别（class-default）。默认类别是系统定义的，并且不能配置。未分类的流量（不符合流量类别中定义的匹配条件的流量）都会被当作默认流量进行处理。

policy-map

policy-map 定义了应该为流量类别实施的行为。这些行为包括：

- 在流量类别中设置特定的 DSCP 或 IP 优先级值；
- 在流量类别中设置 CoS 值；
- 设置 QoS 组；
- 指定流量带宽限制，以及流量超出限制后的行为。

在一个 policy-map 能够生效前，用户必须把它关联到一个端口上。

用户可以使用全局配置命令 **policy-map** 来创建并命名一个 policy-map。当用户输入这条命令后，也就进入了 policy-map 配置模式。在这个模式中，用户可以使用 policy-map 配置命令 **class**，以及 policy-map 类别配置命令 **set**，来指定要为特定的流量类别实施的行为。

用户也可以使用 policy-map 类别配置命令 **police** 和 **bandwidth** 来配置 policy-map，这个 policy-map 定义了流量的限速器和带宽限制，以及当流量超过限速时采取的行为。除此之外，用户还可以使用 policy-map 类别配置命令 **priority** 来配置 policy-map，并为一个类别调节优先级；或者使用 policy-map 类别列队命令 **queue-buffers** 和 **queue-limit** 来配置 policy-map。

要想启用 policy-map，用户需要使用接口配置命令 **service-policy** 把它关联到一个接口。

物理端口上的 policy-map

用户可以在物理端口上配置非层级式 policy-map，它指定了要对哪个流量类别采取 QoS 行为。行为包括在流量类别中设置特定的 DSCP 和 IP 优先级值，为每个匹配的流量类别指定流量带宽限制（限速器），以及当流量超出限制时采取的行为（标记）。

policy-map 具有以下特征：

- 一个 policy-map 中可以包含多个 class 语句，每个 class 语句拥有不同的匹配条件和限速器；
- 一个 policy-map 中可以包含一个预定义的默认流量类别，这个默认流量类别明确放置在 policy-map 的末尾。

  当用户使用 policy-map 配置命令 **class class-default** 配置默认流量类别时，未分类的流量（不符合流量类别中定义的匹配条件的流量）都会被当作默认流量类别（**class-default**）进行处理。
- 一个端口上对不同类型的流量可以设置不同的 policy-map。

VLAN 上的 policy-map

交换机能够支持 VLAN QoS 特性，使用户能够在 VLAN 级别，使用入站数据帧的 VLAN 信息，来执行 QoS 行为（分类和 QoS 行为）。在基于 VLAN 的 QoS 中，用户可以在 SVI 接口上应用服务策略。属于一个 VLAN policy-map 的所有物理接口都需要调用这个基于 VLAN 的 policy-map，而不是基于端口的 policy-map。

尽管用户是在 VLAN SVI 接口上应用的 policy-map，但限速（速率限制）行为只能基于每个端口来执行。用户不能为多个物理端口的总流量实施限速器。每个端口都有一个单独的限速器，来管理进入这个端口的流量。

限速

当数据包分类完成后，并且数据包上已经分配了 DSCP、CoS 或 QoS 组标签后，就可以开始对其实施限速和标记特性了。

限速特性就是创建一个限速器，来为流量指定带宽限制。超出限制的数据包是*超出限制*或*不合格*的。每个限速器都以数据包为基础来确定数据包是符合限制的，还是超出限制的，并指定数据包的行为。这些行为由标记器来执行，其中包括让数据包不做任何修改地通过、丢弃数据包，或修改（降低）数据包的 DSCP 或 CoS 值并允许数据包通过。

为了避免出现失序数据包，合格流量和不合格流量通常都会从相同的队列发出。

注释： 如果用户配置了限速器，那么所有流量（无论是桥接流量还是路由流量）都受到限

速器的监控。因此，桥接数据包可能会被丢弃，或者当它们被限速和标记时，会被修改 DSCP 或 CoS 字段。

用户只能在物理端口上配置限速特性。

在用户配置了 policy-map 和限速行为后，需要使用接口配置命令 **service-policy**，把策略关联到一个入向端口或 SVI 接口上。

令牌桶算法

限速特性使用了令牌桶算法。当交换机接收到每个数据帧时，都会向桶中添加一个令牌。桶中有一个洞，并以一定的速率向外漏令牌，这个速率是由用户以比特每秒为单位指定的平均流量速率。在每次向桶中添加令牌时，交换机都会确认桶中是否有足够的空间。如果桶中没有足够的空间，相关数据包就会被标记为不合格，然后数据包会接受相应的限速器行为（丢弃或降低优先级）。

桶会多快填满是桶深度（突发字节）、漏出令牌的速度（bit/s 速率），以及平均速率之上突发的周期，所提供的功能。桶的大小限制了突发长度的上限，限制了背板到背板能够传输的数据帧数量。如果突发时间较短，并且桶不会满溢，就不会有任何行为施加在流量上。但如果突发时间较长速率较高，并且使桶变满，限速策略就会被施加在突发中的数据帧上。

用户可以使用 policy-map 类别配置命令 **police** 的突发字节（burst-byte）选项，来配置桶深度（在桶满之前能够承受的最大突发）。用户可以使用 policy-map 类别配置命令 **police** 的速率（rate）选项，来配置漏出令牌的速度（平均速率）。

标记

标记特性负责把特定信息传递到网络中的下游设备上，或者把信息从一个接口传递到另一个接口。

标记特性可以用来设置数据包头部中的特定字段/比特，或者也可以使用标记特性来设置数据包结构中的特定字段，这是交换机的内部信息。除此之外，标记特性也可以用来定义字段之间的映射关系。QoS 可以使用以下标记方法：

- 数据包头部
- 设备指定信息
- table-map

数据包头部标记

标记数据包头部中的字段可以分为以下两类：

- IPv4/IPv6 头部比特标记
- 二层头部比特标记

IP 级别的标记特性可以用来把 IP 头部的 IP 优先级和 DSCP 设置为指定的值，以此在下游设备（交换机或路由器）上实现逐跳行为，或者也可以使用标记特性把不同入站接口汇集的流量，在出站接口分到一个类别中。目前支持对 IPv4 和 IPv6 头部进行标记。

二层头部中的标记通常会被用来影响下游设备（交换机或路由器）中的丢弃行为。它会与二层头部中的匹配信息协同工作。二层头部中可以使用 policy-map 设置的比特是服务类别（CoS）。

交换机特定信息标记

这种形式的标记行为包括标记数据包数据结构中的字段，这些内容不是数据包头部的一部分，使数据路径中的其他设备也可以使用这个标记。这种标记不在交换机之间传播。对 QoS 组的标记就属于这一类。只有应用在入站接口上的策略才支持这种类型的标记。用户可以在同一台交换机的出站接口上启用相应的匹配机制和相应的 QoS 行为。

table-map 标记

table-map 标记特性能够使用一个转换表，把一个字段映射和转换为另一个字段。这个转换表就称为 table-map。

根据接口上关联的 table-map 不同，数据包的 CoS、DSCP 和 UP 值都可以被重写。交换机上能够同时配置入向 table-map 策略和出向 table-map 策略。

注释： 一个堆栈中总共支持 14 个 table-map。在一个有线端口的一个方向上只支持一个 table-map。

举例来说，一个 table-map 可以用来把二层 CoS 设置映射到三层 IP 优先级值。用户使用这个特性可以在一个 table-map 中设置多个 **set** 命令，**set** 命令指明了执行映射的方法。这个 table-map 可以由多个策略进行调用，或者在一个策略中调用多次。

下面这个表格中列出了当前支持的映射形式：

表 91：用来建立映射关系的数据包标记类型

To（去往）数据包标记类型	From（来自）数据包标记类型
优先级	CoS
优先级	QoS 组
DSCP	CoS
DSCP	QoS 组
CoS	优先级
CoS	DSCP
QoS 组	优先级
QoS 组	DSCP

基于 table-map 的策略支持以下功能：

- 突变——用户可以创建一个 table-map，其中记录了从一个 DSCP 值映射为另一个 DSCP 值的设置，这个 table-map 可以关联到出向端口上；
- 重写——根据用户配置的 table-map 来对入站数据包进行重写；
- 映射——基于 table-map 的策略可以代替使用 set 命令的策略。

用户在使用 table-map 进行标记时需要执行以下步骤：

1. 定义 table-map——用户需要使用全局配置命令 **table-map** 来设置值的映射。这个 table-map 对于它会用到的策略或类别一无所知。如果 From 字段中没有匹配信息的话，table-map 中的默认命令会用来指明被复制到 To 字段中的值；
2. 定义 policy-map——用户必须定义 table-map 会使用的 policy-map；
3. 把策略关联到一个接口。

注释： 入站端口上的 table-map 策略会改变端口上的信任设置，把它改为 From 类型的 QoS 标记。

流量调节

为了在网络中支持 QoS，进入服务提供商网络的流量需要在网络边界路由器上执行限速，来确保流量速率保持在服务限制内。即使在网络边界上，只有少部分路由器开始发送比网络核心的部署更多的流量，这写增加的流量也会导致网络的拥塞。网络性能的降低会导致难以为所有网络流量提供 QoS 保障。

流量限速功能（使用限速特性）和整形功能（使用流量整形特性）可以管理流量速率，但它们之间的区别在于当令牌用光时对于流量的处理方式。令牌的概念来自于令牌桶机制，这是一个流量计量功能。

注释： 在对网络流量进行 QoS 测试时，用户可能会在整形数据和限速策略中看到不同的结果。通过整形特性处理的网络流量数据提供了更精确的结果。

下面这个表格对比了限速功能和整形功能。

表 92：限速功能和整形功能的对比

限速功能	**整形功能**
以线路速率发送合格流量并允许突发	平缓发送流量并以恒定速率发送
令牌用完时马上采取行动	令牌用完时，先把数据包缓存起来，等有令牌可用时再发送。使用了整形特性的类别会关联一个队列，这个队列就是在这种情况中用来缓存数据包的
限速特性中可以配置多种单位——比特每秒、数据包每秒、网元每秒	整形特性中只能配置一种单位——比特每秒
限速特性可以在一个事件上关联多个行为，比如标记和丢弃行为	整形特性不能对不合格数据包进行标记
适用于入站流量和出站流量	只适用于出站流量
传输控制协议（TCP）会以线路速率来测试线路状况，但会在发生丢包后通过减小自己的窗口大小，把速率调整为用户配置的速率	TCP 能够检测到它有一条速率较低的线路，并相应地调整自己的重传计时器。结果是缩小重传范围，这个结果对于 TCP 来说能够接收

限速

QoS 限速特性的作用是为流量类别施加一个最大速率。QoS 限速特性也可以与优先级特性一起使用，来限制优先级流量。如果流量超出了限制速率，限速特性会在这个事件发生时马上执行相应行为。这个速率（承诺信息速率[CIR]和最高信息速率[PIR]）和突发参数（合格的突发大小[B_c]和超出的突发大小[B_e]）在配置时都是以字节每秒为单位的。

QoS 支持以下限速形式或限速器：

- 单速双色限速
- 双速三色限速

注释： 不支持单速三色限速。

但速率双色限速

当用户只配置了一个 CIR 和一个 B_c 时，使用的就是单速双色限速器。

B_c 是个可选参数，如果用户没有指定的话，设备默认会计算出来。在这种模式中，当入站数

据包有足够的令牌可用时，限速器就认为数据包是合格的。如果当数据包到达时，在 B_c 的限制范围中没有足够多的可用令牌，限速器就认为数据包超出了用户配置的速率限制。

双速三色限速

在使用双速限速器时，交换机只支持色盲模式。在这个模式中，用户需要配置一个承诺信息速率（CIR）和一个最高信息速率（PIR）。顾名思义，这个模式中使用两个令牌桶，一个用于最高速率，另一个用于合格速率。

在色盲模式中，入站数据包首先会与最高速率令牌桶进行比较。如果这里没有足够多的令牌可用，限速器就认为数据包违反了速率限制。如果这里有足够多的令牌可用，接着限速器会检查合格速率令牌同种的令牌，以此确定是否有足够多的令牌可用。最高速率令牌桶中的令牌数量会根据数据包的大小而减少。如果合格速率令牌桶中没有足够多的令牌可用，限速器就认为数据包超出了用户配置的速率。如果有足够多的令牌可用，限速器就会认为数据包是合格的，然后这两个桶中的令牌数量都会随数据包的大小而减少。

桶中补充令牌的速度取决于数据包的到达时间。假设有一个数据包在时间 T1 时到达，接着另一个数据包在时间 T2 时到达。T1 和 T2 之间的时间间隔决定了令牌桶中需要添加的令牌数量。计算方式为：

数据包之间的到达时间间隔（T2-T1）* CIR/8 字节

整形

整形是为流量施加最大速率的过程，并且整形的原则是为了让下游交换机和路由器不会遭遇拥塞。整形最常见的形式是用来限制从物理接口或逻辑接口发送流量的速率。

整形特性关联了一个缓存，这个缓存能够确保在没有足够多的令牌可用时，把数据包缓存下来，而不是立马丢弃。能够为整形流量的子集所使用的缓存数量也是有限的，这个值是基于多种参数计算出来的。用户也可以使用特定的 QoS 命令来调整这个缓存数量。当缓存可用时，数据包就会被缓存起来，否则数据包就会被丢弃。

基于类别的流量整形

交换机可以使用基于类别的流量整形特性。这个整形特性是在一个策略中的一个类别上启用的，这个策略与一个接口相关联。配置了整形特性的类别会被分配一个缓存号码，在没有足够多的令牌时用这个缓存来暂时储存数据包。被还存起来的数据包从这个类别中被发出时会使用 FIFO（先进先出）策略。在最常用的模式中，基于类别的整形会被用来为物理接口和逻辑接口整体施加一个最大速率。一个类别中支持以下整形模式：

- 平均速率整形
- 层级式整形

整形特性是使用令牌桶实施的。CIR、B_c 和 B_e 的值决定了数据包的发送速率，以及填充令牌的速率。

平均速率整形

用户可以使用 policy-map 类别配置命令 **shape average** 来配置平均速率整形特性。

这条命令为一个指定的类别配置了最大带宽。队列带宽会被限制为这个值，哪怕端口有更多的带宽可用。用户可以使用百分比来配置整形平均速率，也可以直接指定目标比特速率值。

层级式整形

用户也可以在层级式模型中，配置多个等级的整形规则。用户可以创建一个父系策略并在其中配置整形特性，然后在关联子系策略并在子系策略中把其他整形配置关联到父系特性。

用户能够配置以下两种类型的层级式整形特性：

- 端口整形器
- 用户配置的整形

端口整形器使用 class-default，并且只有父系策略中允许的行为才会被整形。端口整形器的列队行为是关联在子系策略中的。在使用用户配置的整形特性时，用户不能在子系策略中设置列队行为。

列队和调度

列队特性和调度特性都有助于预防流量拥塞。交换机支持下列列队和调度特性：

- 带宽
- 加权尾部丢弃
- 优先级队列
- 队列缓存

当用户在一个端口上定义列队策略时，控制数据包是映射在最优的优先级队列中的，使用最高的门限值。在以下环境中，控制数据包队列的映射工作有所不同：

- 不使用服务质量（QoS）策略——如果用户没有配置 QoS 策略的话，携带 DSCP 值 16、24、48 和 56 的控制数据包会被映射到队列 0 中，拥有门限值 2 的最高门限值；
- 使用用户定义的策略——在出向端口上配置的用户定义的列队策略，会影响控制数据包上的默认优先级队列设置。

QoS 特性会根据以下规则把控制流量重定向到最优队列中：

1. 如果用户定义了一个用户策略，最高等级的优先级队列总是会被选为最优队列；
2. 如果没有配置优先级队列，Inspur INOS 软件会选择队列 0 作为最优队列。当软件把队列 0 选择为最优路径时，用户必须为这条队列指定最高带宽，以便为控制平面流量提供

最好的 QoS 行为；

3. 如果用户没有在最优队列上配置门限值，Inspur INOS 软件会把携带差分服务代码点（DSCP）值 16、24、48 和 56 的控制数据包映射到门限值 2，并把最优队列中的其余控制流量重新分配到门限值 1。

如果一个策略中没有明确对控制流量实施配置，Inspur INOS 软件会把所有不匹配的控制流量都以门限值 2 映射到最优队列中，匹配的控制流量会按照用户配置的策略，映射到相应的队列中。

注释： 为了对三层数据包提供适当的 QoS，用户必须确保把数据包明确地分类到适当的队列中。当软件在默认队列中检测到 DSCP 值的时候，它会自动把数据包分配到最优队列中。

带宽

交换机能够支持以下带宽配置：

- 带宽百分比
- 带宽剩余率

带宽百分比

用户可以使用 policy-map 类别配置命令 **bandwidth percent** 来为指定类别分配最小带宽。用户配置值的总和不能超过 100%，如果总和小于 100%的话，其他带宽会平均分到所有带宽队列中。

注释： 如果其他队列没有消耗掉所有端口带宽的话，一条队列可以超额订阅带宽。

用户不能在一个 policy-map 中混合使用不同的带宽类型。举例来说，用户不能在一个 policy-map 中同时以带宽百分比，以及 kbits/s 来配置带宽。

带宽剩余率

用户可以使用 policy-map 类别配置命令 **bandwidth remaining ratio** 创建一个比率，用来在指定队列中分享未使用的带宽。指定队列可以在未使用的带宽中，占用用户配置的比率。用户可以在策略中，为指定队列同时使用这条命令和 **priority** 命令。

在用户分配比率时，队列中会被分配到与这些比率相同的权重。用户可以指定的比率范围是从 0 至 100。举例来说，如果用户为一个类别配置了带宽剩余率 2，为另一个类别的队列分配带宽剩余率 4。那么带宽剩余率 4 在调度的执行中就会以带宽剩余率 2 的两倍来执行。

为策略分配的总带宽率可以超过 100。举例来说，用户可以为一条队列分配带宽剩余率 50，为另一条队列分配带宽剩余率 100。

加权尾部丢弃

出向队列使用称为加权尾部丢弃（WTD）的高级版尾部丢弃拥塞避免机制。WTD 是实施在队列中的，来管理队列长度，为不同的流量类别提供丢弃优先级。

在数据帧被排列如某条队列后，WTD 会使用数据帧中被分配的 QoS 标记来为其设定不同的门限值。如果数据帧超出了对它 QoS 标签设置的门限值标准（目的队列中的空间小于数据帧大小），数据帧就会被丢弃。

每条队列中有三个可配置的门限值。QoS 标签决定了数据帧会使用这三个门限值中的哪一个。下图展示了 WTD 在一条队列上的操作，这条队列的大小是 1000 个数据帧。用户配置的三个丢弃优先级分别是 40%（400 个数据帧）、60%（600 个数据帧）和 100%（1000 个数据帧）。这些百分比表示的是在 40%门限值的队列中最多可以排列 400 个数据帧、在 60%门限值的队列中最多可以排列 600 个数据帧，以及在 100%门限值的队列中最多可以排列 1000 个数据帧。

图 88：WTD 和队列的操作

（图 88）

在这个示例中，CoS 值 6 比其他 CoS 值都更为重要，因此用户为它分配的丢弃门限值是 100%（队列排满的状态）。CoS 值 4 的门限值是 60%，CoS 值 3 的门限值是 40%。用户使用命令 **queue-limit cos** 来分配这些门限值。

假设队列中已经排入了 600 个数据帧，这时又有一个新的数据帧达到了。这个数据帧携带 CoS 值 4，因此适用于 60%门限值。如果把这个数据帧添加到队列中，就超出了门限值的限制，因此这个数据帧会被丢弃。

加权尾部丢弃默认值

以下内容为加权尾部丢弃（WTD）的默认值，以及配置 WTD 门限值的规则。

- 如果用户为 WTD 配置少于 3 个队列限制百分比，WTD 默认值就会被分配给这些门限值。以下为 WTD 门限值的默认值：

  表 93：WTD 门限值的默认值

门限值	默认值百分比
0	80
1	90
2	400

- 如果用户配置了 3 个不同的 WTD 门限值，队列就会按照用户的配置分别对应适当的门限值；

- 如果用户配置了 2 个 WTD 门限值，那么最大的值百分比就会是 400；
- 如果用户配置了 1 个门限值为 x，那么最大的值百分比就会是 400。
  - 如果 x 的值小于 90，那么门限值 1=90，门限值 0=x；
  - 如果 x 的值等于 90，那么门限值 1=90，门限值 0=80；
  - 如果 x 的值大于 90，那么门限值 1=x，门限值 0=80。

优先级队列

每个端口上支持 8 条出向队列，其中 2 条上可以设置优先级。

用户可以使用策略 class-map 命令 **priority level** 来为两个类别配置优先级。一个类别必须配置为优先级队列等级 1，另一个类别必须配置为优先级队列等级 2。这两条队列中的数据包会比其他队列中的数据包拥有更低的延迟。

注释： 用户可以只配置一个等级的优先级。

一个 policy-map 中可以值限定一个优先级或一个优先级等级。一个 policy-map 中可以不使用 kbit/s 为单位，配置多个拥有相同优先级等级的优先级队列，但前提是这些队列中都配置了限速特性。

队列缓存

交换机上的每个千兆端口都有为有线端口分配的 300 缓存。每个万兆端口都分配了 1800 缓存。在启动时，如果有线端口上没有启用 policy-map，会默认创建两条队列。用户可以在有线端口上使用 MQC 策略最多配置 8 条队列。下面这个表格中列出了了哪些数据包会进入到哪条队列中：

表 94：DSCP、优先级和 CoS 值的门限值映射表

DSCP、优先级或 CoS	**队列**	**门限值**
控制数据包	0	2
其他数据包	1	2

注释： 用户可以通过为一条队列设置对其门限值并分配最大可用缓存，来保障缓存的可用性。用户可以使用 policy-map 类别命令 **queue-buffers** 来配置队列缓存。用户可以使用 policy-map 类别命令 **queue-limit** 来配置最大门限值。

用户可以使用两种缓存分配方式：硬缓存，也就是明确地为某条队列保留；软缓存，如果指定端口没有使用这部分缓存的话，其他端口可以使用。

有线端口的默认状态为：队列 0 拥有 40%的缓存，作为硬缓存分配给接口；也就是对于千兆

端口来说，有 120 缓存分配给队列 0；对于万兆端口来说，有 720 缓存分配给队列 0。对于千兆端口和万兆端口来说，分配给这条队列的最大软缓存分别设置为 480（计算方式是 120*400/100）和 2880，其中 400 是为任意队列配置的默认最大门限值。

队列 1 上没有分配任何硬缓存，默认软缓存的限制设置为 400（这也是最大门限值）。这个门限值定义了这条队列能够从公共池中借用的软缓存最大数量。

队列缓存分配

用户可以使用 policy-map 类别配置命令 **queue-buffers ratio** 来调整分配给任意队列的缓存。

动态门限值与缩放

传统上，保留的缓存是静态分配给每条队列的。无论队列是否活跃，它的缓存就是由队列自己保留的。除此之外，随着队列数量的增长，每条队列会分配到的保留缓存就会变得越来越小。最终有可能会出现这么一种情况，那就是每条队列分配到的保留缓存都不足以支持一个巨型数据帧。

动态门限值与缩放（DTS）特性提供了一种公平且高效的缓存资源分配方式。当拥塞发生时，DTS 机制会基于全局/端口资源的占用情况，提供弹性的缓存分配方案。从概念上说，DTS 会随着自愿的消耗，逐渐缩小队列缓存分配决策，为其他队列留出空间；反之亦然。这种灵活的方式能够更加有效和公平地利用缓存资源。

如上所述，一条队列中可以配置两个限制条件——硬限制和软限制。

硬限制不是 DTS 的一部分。硬缓存只用于特定队列。硬缓存之和应该小于全局设置的硬缓存最大限制。为所有出向队列设置的全局硬缓存限制当前为 5705。在默认环境中，如果用户没有配置 MQC 策略，24 个千兆端口会占用 24*67=1608，4 个万兆端口会占用 4*720=2880，总共占用 4488 缓存，用户可以根据配置分配更多硬缓存。

软缓存限制是 DTS 特性的一部分。除此之外，有些软缓存的分配结果可以超出全局软缓存分配限制。为所有出向队列设置的全局软缓存分配限制当前为 7607。硬缓存和软缓存的总和加在一起为 13312，也就是 3.4 MB。由于软缓存的分配总和可以超出全局限制，因此当系统负载很低时，一条队列可以使用大量缓存资源。DTS 特性可以在系统的负载变得沉重时，动态调整每条队列中分配的软缓存。

注释： 默认情况下不启用队列 Q0 和 Q1。

注释： 队列 Q2 和 Q3 中的流量使用加权轮循策略。

去往上游方向只有一条队列可用。端口和比率限制只应用在去往下游的方向上。

注释： 有线端口支持 8 条队列。

信任行为

有线端口的信任行为

对于连接在交换机上的有线端口来说（端点设备比如 IP 电话、笔记本电脑、摄像头、网真设备或其他设备），这些端点设备发来的 DSCP、优先级或 CoS 值是受到交换机的信任的，因此在用户没有没有明确配置策略时，这些标记也是会保留的。

这种信任行为同时适用于上游和下游 QoS。

数据包会根据默认的初始配置被排列到适当的队列中。默认交换机上是没有优先级队列的。对于单播和组播数据包来说都是如此。

在入站数据包类型与出站数据包类型不同的情况中，信任行为和列队行为的解释详见下表。需要注意的是，一个端口默认的信任模式是基于 DSCP 值的。如果入站数据包是一个纯二层数据包的话，信任模式会“后退”为 CoS 值。用户也可以把信任设置从 DSCP 变更为 CoS。用户可以使用 MQC 策略来完成这个设置变更，也就是在 class-default 中执行“set cos cos table default default-cos”行为，其中 default-cos 是用户创建的 table-map 的名称（它只执行默认复制）。

表 95：信任和队列行为

入站数据包	出站数据包	信任行为	队列行为
三层	三层	保留 DSCP/优先级	基于 DSCP
二层	二层	不适用	基于 CoS
标记	标记	保留 DSCP 和 CoS	基于 DSCP（信任 DSCP 游先于优先级）
三层	标记	保留 DSCP，CoS 设置为 0	基于 DSCP

上述有线端口的信任默认设置再找个版本的软件中也是相同的。为了兼容已有的有线标准，默认所有流量都会排入尽力而为的队列。在下游方向上，维护语音、视频、尽力而为和背景队列的接入点会实施队列特性。接入点会根据 802.11e 标记信息来选择队列策略。

在信任边界上为 Inspur IP 电话提供的端口安全

在典型的网络中，用户把一台 IP 电话连接到一个端口，并且级联一台设备，这台设备会从电话背后生成携带数据信息的数据包。Inspur IP 电话会通过以下行为确保语音质量：它在通

过共享的数据链路发送语音数据包时，会把 CoS 等级标记为高优先级（CoS=5），并且把携带数据的数据包标记为低优先级（CoS=0）。电话发往交换机的流量通常会携带标记，这个标记会携带在 802.1Q 头部中。这个头部中还会包含 VLAN 信息和服务类别（CoS）的 3 比特字段，这个字段标明了数据包的优先级。

对于大多数 Inspur IP 电话的配置来说，电话发送的流量应该收到信息，以此来为语音流量提供比网络中其他类型的流量更高的优先级。通过使用接口配置命令 **trust device**，用户可以让连接电话的端口信任自己接收到的流量。

注释： 接口配置命令 **trust device** *device_type* 是设备上的单机命令。当用户在 Auto-QoS 的配置中使用这条命令时，如果端口连接的对等体设备不是对应设备（也就是定义为符合网络中信任策略的设别）的话，CoS 和 DSCP 值都会设置为“0”，并且入站策略也不会生效。如果连接的对等体设备是对应设备的话，入站策略就会生效。

在使用信任设置时，用户还可以使用信任边界特性，来防止滥用高优先级队列，比如用户绕过电话把 PC 直接连接在交换机上。如果没有设置信任边界的话，交换机是会信任 PC 生成的 CoS 标签的（因为设置了信任 CoS）。相反，信任边界特性会使用 CDP 来检测端口上连接的设备是不是 Inspur IP 电话（比如 Inspur IP 电话 7910、7935、7940 和 7960）。如果没有检测到电话，信任边界特性就会禁用这个端口上的信任设置，防止有人滥用高优先级队列。需要注意的是，如果 PC 和 Inspur IP 电话通过集线器连接到交换机，信任边界特性就不会生效。

相关主题

为设备类型配置信任行为

标准 QoS 的默认设置

默认的有线 QoS 配置

交换机的每个有线接口上默认配置了两条队列。所有控制流量都是由队列 0 进行处理的。所有其他流量都是由队列 1 进行处理的。

DSCP 映射

默认的 CoS 到 DSCP 映射

用户可以使用 CoS 到 DSCP 映射，把入站数据包携带的 CoS 值映射为 QoS 用来在内部表示流量优先级的 DSCP 值。下面这个表格中展示了默认的 CoS 到 DSCP 映射。如果这些值不适用于用户网络，用户就需要自行修改。

表 96：默认的 CoS 到 DSCP 映射

CoS 值	DSCP 值
0	0
1	8
2	16
3	24
4	32
5	40
6	48
7	56

默认的 IP 优先级到 DSCP 映射

用户可以使用 IP 优先级到 DSCP 映射，把入站数据包携带的 IP 优先级值映射为 QoS 用来在内部表示流量优先级的 DSCP 值。下面这个表格中展示了默认的 IP 优先级到 DSCP 映射。如果这些值不适用于用户网络，用户就需要自行修改。

表 97：默认的 IP 优先级到 DSCP 映射

IP 优先级值	DSCP 值
0	0
1	8
2	16
3	24
4	32
5	40
6	48
7	56

默认的 DSCP 到 CoS 映射

用户可以使用 DSCP 到 CoS 映射来生成一个 CoS 值，这个值用来在四条出向队列中选择一条。下面这个表格中展示了默认的 DSCP 到 CoS 映射。如果这些值不适用于用户网络，用户就需要自行修改。

表 96：默认的 DSCP 到 CoS 映射

DSCP 值	CoS 值
0～7	0
8～15	1
16～23	2

24～31	3
32～39	4
40～47	5
48～55	6
56～63	7

QoS 策略的指导

用户应该遵循以下指导，防止客户端设备由于不正确的 QoS 策略而被排除在网络之外：

- 在向设备上添加新的 QoS 策略时，同一个漫游域或移动域中的其他设备上也应该配置名称相同的 QoS 策略；
- 当一台设备上加载了一个较新的软件版本时，它能够支持新的策略格式。如果用户把软件镜像从一个较老的版本升级为一个较新的版本，用户应该分别保存配置。在设备加载较老的版本镜像时，可能有些 QoS 策略这个版本无法支持，这时用户就应该把这些 QoS 策略重新恢复为这个软件版本所支持的策略格式。

有线目标上 QoS 的限制条件

目标指的是能够在其上应用策略的实体。用户可以把策略应用到一个有线目标上。有线目标可以是端口或 VLAN。只有端口、SSID 和客户端策略是用户可以配置的。射频策略是用户无法配置的。

并且支持客户端目标。

在设备上为有线目标应用 QoS 特性时，有以下限制条件：

- 连接有线目标的设备端口上最多可以支持 8 条队列；
- 连接有线目标的有线端口上，每个策略中最多支持 63 个限速器；
- QoS 层级中最多支持两个等级；
- 在层级式策略中，父系和子系之间不允许发生覆盖行为，除非父系策略中定义了端口整形器，且子系策略中定义了队列特性；
- QoS 策略不能与任何 EtherChannel 接口相关联；
- 层级式 QoS 策略中的父系策略和子系策略都不支持限速；
- 层级式 QoS 策略中的父系策略和子系策略都不支持标记；

- 一个策略中不支持混用队列限制和队列缓存；

注释： 设备商不支持队列限制百分比，因为命令 **queue-buffer** 负责控制这个功能。只支持使用 DSCP 和 CoS 值来定义队列限制。

- 在使用整形特性时，每个数据包上都有一个 20 字节的 IPG 负载，这是在硬件内部计算的。整形特性实际上会受到它的影响，尤其是对于小数据包来说；
- 在所有上行有线端口（万兆以太网端口）上，所有基于队列的有线策略中分类顺序应该相同，在所有下行有线端口（千兆以太网端口）上也应该相同；
- 不支持空类别；
- 不支持行为为空的 class-map。如果两个策略中定义了顺序相同的 class-map，并且其中一个策略的 class-map 中没有指定行为，就有可能遇到流量丢弃事件。解决方法是为优先级队列（PRIORITY_QUEUE）中的所有类别分配最小带宽；
- 在连接有线目标的有线端口上，每个策略支持最多 256 个类别；
- policy-map 中的限速器行为拥有以下限制条件：
  - 合格流量的行为必须是传输；
  - 超出/违反行为在降低优先级标记时只能使用 CoS 到 CoS、优先级到优先级、DSCP 到 DSCP 类型的标记；
  - 一个策略中的降低优先级标记类型必须一致。
- 端口级别的标记策略会优先于 SVI 接口上的策略；但如果用户没有配置端口策略，SVI 策略就会被优先考虑。要想让端口策略获得优先权，用户需要定义一个端口级别的策略；这样 SVI 的策略就会被覆盖；
- 使用分类计数器拥有以下特殊限制条件：
  - 分类计数器会统计数据包，而不是字节；
  - 不支持基于过滤器的分类计数器；
  - 只有与标记特性或限速特性相关的 QoS 配置才会触发分类计数器；
  - 分类计数器并不是基于端口的。也就是说分类计数器会汇集不同接口上的，属于同一个策略中同一个类别的所有数据包；
  - 如果用户在策略中应用了限速行为或标记行为，class-default 类别中就会使用分类计数器；
  - 如果一个类别中有多个匹配条件，分类计数器只会显示匹配一个条件的流量。
- 使用 table-map 拥有以下限制条件：
  - 针对一个目标在一个方向上，只支持使用一个 table-map 对超出限速特性的流量进行降低优先级的标记，只支持使用一个 table-map 对违反限速特性的流量进行降低优先级的标记；

- 用户必须在 class-default 下配置 table-map；用户定义的类别中不支持 table-map。

- 使用层级式策略拥有以下需求：
  - 端口整形器
  - 汇聚限速器
  - PV 策略
  - 父系整形和子系标记/限速
- 对于连接有线目标的端口来说，只支持以下层级式策略：
  - 同一个策略中不支持使用限速链；
  - 同一个策略中不支持层级式队列特性（端口整形器除外）；
  - 在父系类别中，所有过滤器的类型必须相同。子系过滤器类型必须与父系过滤器类型相匹配，以下几点是例外：
    - 如果父系类别中配置了匹配 IP，那么子系类别中可以配置匹配 ACL；
    - 如果父系类别中配置了匹配 CoS，那么子系类别中可以配置匹配 ACL。
  - 接口配置模式中的命令 **trust device** *device_type* 是交换机上的独立命令。在 AutoQoS 配置中使用这条命令时，如果连接的对等体设备不是对应设备（也就是符合用户信任策略的设备），那么 CoS 和 DSCP 值都会设置为“0”，任何入站策略也不会生效。如果连接的对等体设备是对应设备，那么入站策略就会生效。

在 VLAN 中向有线目标应用 QoS 特性时有以下限制条件：

- 对于扁平或非层级式策略来说，只支持标记特性或 table-map。

在 EtherChannel 和 EtherChannel 成员接口上应用 QoS 特性是有以下限制条件和考量因素：

- EtherChannel 接口上不支持 QoS 特性；
- EtherChannel 成员接口的入方向上和出方向上都支持 QoS 特性。所有 EtherChannel 成员必须都应用相同的 QoS 策略。如果 QoS 策略不相同，那么不同链路上的策略会独立生效；
- 当用户在向 EtherChannel 成员上应用服务策略时，会看到以下警告消息，这个消息提示用户要在这个 EtherChannel 中所有端口上都应用相同的策略：“Warning: add service policy will cause inconsistency with port xxx in ether channel xxx.”；
- EtherChannel 成员接口上不支持 Auto-QoS 特性。

注释： 在用户向 EtherChannel 上添加服务策略时，会在控制台上看到以下信息：“Warning: add service policy will cause inconsistency with port xxx in ether channel xxx.”。这个警告消息是正常的。这个消息是为了提醒用户要在这个 EtherChannel 中的其他端口上也应用相同的策略。在启动时也会看到同一条消息。这条消息并不意味着 EtherChannel 成员端口之间有什么差异。

如何配置 QoS

配置类别、策略和 table-map

创建一个流量类别（CLI）

要想创建一个包含有匹配条件的流量类别，用户需要使用 **class-map** 命令来指定流量类别的名称，然后在 class-map 配置模式中，按照需要配置 **match** 命令。

在开始前

这个配置任务中配置的所有 match 命令都是可选的，但用户必须在一个类别中至少配置一条 match 条件。

总步骤

1. configure terminal

2. class-map {*class-map name* | **match-any**}

3. match access-group {*index number* | *name*}

4. match class-map *class-map name*

5. match cos *cos value*

6. match dscp *dscp value*

7. match ip {**dscp** *dscp value* | **precedence** *precedence value* }

8. match non-client-nrt

9. match qos-group *qos group value*

10. match vlan *vlan value*

11. match wlan user-priority *wlan value*

12. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**class-map** {*class-map name* |	进入 class-map 配置模式。

	match-any}
	
	示例：
	`Device(config)# class-map test_1000`
	`Device(config-cmap)#`
		• 创建一个 class-map，用来把数据包匹配到用户定义的类别中
		• 如果用户指定了 **match-any**，那么流量必须至少与其中一个条件相匹配，这样这个流量才会被分类到这个流量类别中。这是默认设置

步骤 3	**match access-group** {*index number* | *name*}
	
	示例：
	`Device(config-cmap)# match access-group 100`
	`Device(config-cmap)#`
		用户可以在这条命令中配置以下参数：
		• access-group
		• class-map
		• cos
		• dscp
		• ip
		• non-client-nrt
		• precedence
		• qos-group
		• vlan
		• wlan user priority
		（可选）在示例中，用户输入了 access-group ID：
		• 访问列表索引（取值从 1 至 2799）
		• 命名的访问列表

步骤 4	**match class-map** *class-map name*
	
	示例：
	`Device(config-cmap)# match class-map test_2000`
	`Device(config-cmap)#`
		（可选）匹配另一个 class-map 的名称

步骤 5	**match cos** *cos value* **示例：** `Device(config-cmap)# match cos 2 3 4 5`	（可选）匹配 IEEE 802.1Q 或 ISL 服务类别（用户）优先级值。 • 最多输入 4 个 CoS 值，以空格分隔（取值从 0 至 7）

	Device(config-cmap)#	
步骤 6	**match dscp** *dscp value* **示例：** Device(config-cmap)# **match dscp af11 af12** Device(config-cmap)#	（可选）匹配 IPv4 和 IPv6 数据包中的 DSCP 值
步骤 7	**match ip** {**dscp** *dscp value* | **precedence** *precedence value*} **示例：** Device(config-cmap)# **match ip dscp af11 af12** Device(config-cmap)#	（可选）匹配以下 IP 值： • **dscp**——匹配 IP DSCP（差分服务代码点） • **precedence**——匹配 IP 优先级（取值 0 至 7）
步骤 8	**match non-client-nrt**	
步骤 9	**match qos-group** *qos group value* **示例：** Device(config-cmap)# **match qos-group 10** Device(config-cmap)#	（可选）匹配 QoS 组值（取值 0 至 31）
步骤 10	**match vlan** *vlan value* **示例：** Device(config-cmap)# **match vlan 210** Device(config-cmap)#	（可选）匹配 VLAN ID（取值 1 至 4095）
步骤 11	**match wlan user-priority** *wlan value*	
步骤 12	**end** **示例：** Device(config-cmap)# **end**	返回特权 EXEC 模式

接下来做什么？

配置 policy-map。

创建一个流量策略（CLI）

要想创建一个流量策略，用户需要使用全局配置命令 **policy-map** 来指定流量策略名称。

用户需要使用 **class** 命令，把流量类别关联到流量策略中。用户必须在进入 policy-map 配置模式后再使用 **class** 命令。在输入 **class** 命令后，用户会自动进入到 policy-map 类别配置模式中，并在这里为这个流量策略定义 QoS 策略。

用户可以在 policy-map 中配置以下与类别相关的行为：

- admit——允许请求呼叫准入控制（CAC）
- bandwidth——带宽配置选项
- exit——离开 QoS 类别行为控制模式
- no——反向执行命令或恢复默认值
- police——限速器配置选项
- priority——为这个类别严格指定调度优先级配置选项
- queue-buffers——队列缓存配置选项
- queue-limit——为加权尾部丢弃（WTD）设置队列最大门限值的配置选项
- service-policy——配置 QoS 服务策略
- set——使用以下选项来设置 QoS 值：
  - CoS 值
  - DSCP 值
  - 优先级值
  - QoS 组值
  - WLAN 值
- shape——流量整形配置选项

在开始前

用户应该首先创建一个 class-map。

总步骤

1. configure terminal

2. policy-map *policy-map name*

3. class {*class-name* | **class-default**}

4. admit

5. bandwidth {**kb/s** *kb/s value* | **percent** *percentage* | **remaining** {*percent* | *ratio*}}

6. exit

7. no

8. police {*target_bit_rate* | **cir** | **rate**}

9. priority {*kb/s* | **level** *level value* | **percent** *percentage value*}

10. queue-buffers ratio *ratio limit*

11. queue-limit {*packets* | **cos** | **dscp** | **percent**}

12. service-policy *policy-map name*

13. set {**cos** | **dscp** | **ip** | **precedence** | **qos-group** | **wlan**}

14. shape average {*target _bit_rate* | **percent**}

15. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 2	**policy-map** *policy-map name*
	
	示例：
	`Device(config)# policy-map test_2000`
	`Device(config-pmap)#`
		进入 policy-map 配置模式。
		创建或修改一个 policy-map，用户可以把它关联到一个或多个接口，用来指定服务策略

步骤 3	**class** {*class-name*	**class-default**} **示例：** Device(config-pmap)# **class test_1000** Device(config-pmap-c)#	指定用户想要创建或更改的类别的名称。 用户也可以为未分类数据包创建一个系统默认类别
步骤 4	**admit**	
步骤 5	**bandwidth** {**kb/s** *kb/s value*	**percent** *percentage*	**remaining** {*percent*	*ratio*}}	（可选）使用以下选项之一来设置带宽： • **kb/s**——千比特每秒，输入 20000

	示例： `Device(config-pmap-c)#` **`bandwidth 50`** `Device(config-pmap-c)#`	至 10000000 之间的数值 • **percent** ——输入要为这个 policy-map 使用的总带宽的百分比 • **remaining**——输入剩余带宽的百分比 有关这条命令及其用法的更多信息，用户可以参考： 配置带宽（CLI）
步骤 6	**exit** **示例：** Device(config-pmap-c)# **exit** Device(config-pmap-c)#	（可选）离开 QoS 类别行为配置模式
步骤 7	**no** **示例：** `Device(config-pmap-c)#` **`no`** `Device(config-pmap-c)#`	（可选）反向执行命令
步骤 8	**police** {*target_bit_rate* | **cir** | **rate**} **示例：** `Device(config-pmap-c)#` **`police 100000`** `Device(config-pmap-c)#`	（可选）配置限速器： • *target_bit_rate*——输入每秒比特率，输入 8000 至 10000000000 之间的数值 • **cir**——承诺信息速率 • **rate**——指定限速速率，为层级式策略使用 PCR，或为单级别 ATM 4.0 限速器策略使用 SCR 有关这条命令及其用法的更多信息，用户可以参考： 配置限速特性（CLI）
步骤 9	**priority** {*kb/s* | **level** *level value* | **percent** *percentage value*}	（可选）为这个类别严格指定调度优先级配置选项。命令选项包括： • *kb/s*——千比特每秒，输入 1 至

	示例： `Device(config-pmap-c)#` **`priority percent 50`** `Device(config-pmap-c)#`	2000000 之间的数值 • **level**——建立一个多等级优先级队列。输入一个值（1 或 2） • **percent**——输入用于这个优先级的总带宽百分比 有关这条命令及其用法的更多信息，用户可以参考： 配置优先级（CLI）
步骤 10	**queue-buffers ratio** *ratio limit* 示例： `Device(config-pmap-c)#` **`queue-buffers ratio 10`** `Device(config-pmap-c)#`	（可选）为这个类别配置队列缓存。输入队列缓存比率限制（0 至 100） 有关这条命令及其用法的更多信息，用户可以参考： 配置队列缓存（CLI）
步骤 11	**queue-limit** {*packets* | **cos** | **dscp** | **percent**} 示例： `Device(config-pmap-c)#` **`queue-limit cos 7 percent 50`** `Device(config-pmap-c)#`	（可选）为尾部丢弃指定队列最大门限值： • *packets*——默认配置数据包，输入 1 至 2000000 之间的数值 • **cos**——为每个 CoS 值输入参数 • **dscp**——为每个 DSCP 值输入参数 • **percent**——为门限值输入百分比 有关这条命令及其用法的更多信息，用户可以参考： 配置队列限制（CLI）
步骤 12	**service-policy** *policy-map name* 示例： `Device(config-pmap-c)#` **`service-policy test_2000`** `Device(config-pmap-c)#`	（可选）配置 QoS 服务策略
步骤 13	**set** {**cos** | **dscp** | **ip** | **precedence** | **qos-group** | **wlan**}	（可选）设置 QoS 值。用户可以配置的 QoS 值包括以下这些：

	示例：
	`Device(config-pmap-c)# set cos 7`
	`Device(config-pmap-c)#`
		• **cos**——设置 IEEE 802.1Q/ISL 类别的服务/用户优先级
		• **dscp**——设置 IPv4 和 IPv4 数据包中的 DSCP
		• **ip**——设置 IP 特定的值
		• **precedence**——设置 IPv4 和 IPv6 数据包中的优先级值
		• **qos-group**——设置 QoS 组
		• **wlan**——设置 WLAN 用户优先级

步骤 14	**shape average** {*target _bit_rate* | **percent**}
	
	示例：
	`Device(config-pmap-c) #shape average percent 50`
	`Device(config-pmap-c) #`
		（可选）设置流量整形特性。命令参数包括以下这些：
		• *target_bit_rate*——目标比特速率
		• **percent**——为承诺信息速率设置接口带宽的百分比
		有关这条命令及其用法的更多信息，用户可以参考：
		配置整形特性（CLI）

步骤 15	**end**
	
	示例：
	`Device(config-pmap-c) #end`
	`Device(config-pmap-c) #`
		返回特权 EXEC 模式

接下来做什么？

配置接口。

配置基于类别的数据包标记（CLI）

接下来的步骤解释了如何在用户交换机上配置下列基于类别的数据包标记特性：

- CoS 值
- DSCP 值
- IP 值
- 优先级值
- QoS 组值

- WLAN 值

在开始前

在开始这部分配置之前，用户应该已经创建了 class-map 和 policy-map。

总步骤

1. configure terminal

2. policy-map *policy name*

3. class *class name*

4. set cos {*cos value* | **cos table** *table-map name* | **dscp table** *table-map name* | **precedence table** *table-map name* | **qos-group table** *table-map name* | **wlan user-priority table** *table-map name*}

5. set dscp {*dscp value* | **default** | **dscp table** *table-map name* | **ef** | **precedence table** *table-map name* | **qos-group table** *table-map name* | **wlan user-priority table** *table-map name*}

6. set ip {**dscp** | **precedence**}

7. set precedence {*precedence value* | **cos table** *table-map name* | **dscp table** *table-map name* | **precedence table** *table-map name* | **qos-group table** *table-map name*}

8. set qos-group {*qos-group value* | **dscp table** *table-map name* | **precedence table** *table-map name*}

~~**9. set wlan user-priority** {*wlan user-priority value* | **cos table** *table-map name* | **dscp table** *table-map name* | **qos-group table** *table-map name* | **wlan table** *table-map name*}~~

10. end

11. show policy-map

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**policy-map** *policy name* **示例：** `Device(config)# policy-map policy1`	进入 policy-map 配置模式。 创建或修改一个 policy-map，用户可以把它关联到一个或多个接口，用来指定服务策略

	`Device(config-pmap)#`	
步骤 3	**class** *class name* **示例：** `Device(config-pmap)#` **`class class1`** `Device(config-pmap-c)#`	进入策略 class-map 配置模式。指定用户想要创建或更改的类别的名称。 用户可以在策略 class-map 配置模式中指定以下选项： • **admit——**允许请求呼叫准入控制（CAC） • **bandwidth**——带宽配置选项 • **exit**——离开 QoS 类别行为控制模式 • **no**——反向执行命令或恢复默认值 • **police**——限速器配置选项 • **priority**——为这个类别严格指定调度优先级配置选项 • **queue-buffers**——队列缓存配置选项 • **queue-limit**——为加权尾部丢弃（WTD）设置队列最大门限值的配置选项 • **service-policy**——配置 QoS 服务策略 • **set**——使用以下选项来设置 QoS 值： · CoS 值 · DSCP 值 · 优先级值 · QoS 组值 · WLAN 值 • **shape——**流量整形配置选项 **注释：** 这个步骤描述了用户可以使用的 **set** 命令选项。其他命令选项

		（**admit**、**bandwidth** 等）会在这个指导的其他部分进行描述。尽管这个任务中列出了所有可用的 **set** 命令，但每个类别中只能配置一个 **set** 命令

步骤 4	**set cos** {*cos value* | **cos table** *table-map name* | **dscp table** *table-map name* | **precedence table** *table-map name* | **qos-group table** *table-map name* | **wlan user-priority table** *table-map name*}
	
	示例：
	`Device(config-pmap)# set cos 5`
	`Device(config-pmap)#`
		（可选）为出站数据包设置特定的 IEEE 802.1Q 二层 CoS 值。取值范围是 0 至 7.
		用户还可以使用 **set cos** 命令来设置以下值：
		• **cos table**——基于 table-map 设置 CoS 值
		• **dscp table**——基于 table-map 设置代码点值
		• **precedence table** — — 基于 table-map 设置代码点值
		• **qos-group table** — — 基于 table-map 把 QoS 组设置为 CoS 值
		• ~~**wlan user-priority table**——基于 table-map 把 WLAN 用户优先级设置为 CoS 值~~

步骤 5	**set dscp** {*dscp value* | **default** | **dscp table** *table-map name* | **ef** | **precedence table** *table-map name* | **qos-group table** *table-map name* | **wlan user-priority table** *table-map name*}
	
	示例：
	`Device(config-pmap)# set dscp af11`
	`Device(config-pmap)#`
		（可选）设置 DSCP 值。
		除了能够设置具体的 DSCP 值之外，用户还能使用 **set dscp** 命令来设置以下参数：
		• **default**——使用默认 DSCP 值匹配数据包（000000）
		• **dscp table**——基于 table-map 把 DSCP 值设置为数据包 DSCP 值
		• **ef**——使用 EF DSCP 值（101110）来匹配数据包
		• **precedence table** — — 基于 table-map 把优先级值设置为数据

		包 DSCP 值 • **qos-group table** ——基于 table-map 把 QoS 组设置为数据包 DSCP 值 • ~~**wlan-user-priority table**——基于 table-map 把 WLAN 用户优先级设置为数据包 DSCP 值~~
步骤 6	**set ip** {**dscp** | **precedence**} **示例：** Device(config-pmap)# **set ip dscp c3** Device(config-pmap)#	（可选）设置 IP 特定值。这些值可以是 IPDSCP 值，也可以是 IP 优先级值。用户可以使用 **set ip dscp** 命令来设置以下值： • *dscp value*——设置具体的 DSCP 值 • **default**——使用默认 DSCP 值匹配数据包（000000） • **dscp table**——基于 table-map 把 DSCP 值设置为数据包 DSCP 值 • **ef**——使用 EF DSCP 值（101110）来匹配数据包 • **precedence table** ——基于 table-map 把优先级值设置为数据包 DSCP 值 • **qos-group table** ——基于 table-map 把 QoS 组设置为数据包 DSCP 值 • ~~**wlan-user-priority table**——基于 table-map 把 WLAN 用户优先级设置为数据包 DSCP 值~~ 用户可以使用 **set ip precedence** 命令来设置以下值： • *precedence value*——设置优先级值（取值为 0 至 7） • **cos table**——基于 table-map 把二

		层 CoS 设置为优先级值 • **dscp table**——基于 table-map 把 DSCP 值设置为优先级值 • **precedence table** —— 基于 table-map 把优先级设置为优先级值 • **qos-group table** —— 基于 table-map 把 QoS 组设置为优先级值

步骤 7	**set precedence** {*precedence value* | **cos table** *table-map name* | **dscp table** *table-map name* | **precedence table** *table-map name* | **qos-group table** *table-map name*}
	
	示例：
	`Device(config-pmap)# set precedence 5`
	`Device(config-pmap)#`
		（可选）设置 IPv4 和 IPv6 数据包中的优先级值。
		用户可以使用 **set precedence** 命令设置以下值：
		• *precedence value*——设置优先级值（取值为 0 至 7）
		• **cos table**——基于 table-map 把二层 CoS 设置为优先级值
		• **dscp table**——基于 table-map 把 DSCP 值设置为优先级值
		• **precedence table** —— 基于 table-map 把优先级设置为优先级值
		• **qos-group table** —— 基于 table-map 把 QoS 组设置为优先级值

步骤 8	**set qos-group** {*qos-group value* | **dscp table** *table-map name* | **precedence table** *table-map name*} **示例：** `Device(config-pmap)# set qos-group 10`	（可选）设置 QoS 组值。用户可以使用这条命令设置以下值： • *qos-group value*——设置 1 至 31 之间的数值 • **dscp table**——基于 table-map 把 DSCP 值设置为代码点值 • **precedence table** —— 基于

	Device(config-pmap)#	table-map 把优先级设置为代码点值
~~步骤 9~~	~~**set wlan user-priority** {*wlan user-priority value* | **cos table** *table-map name* | **dscp table** *table-map name* | **qos-group table** *table-map name* | **wlan table** *table-map name*}~~	
步骤 10	**end** **示例：** Device(config-pmap)# **end** Device#	返回特权 EXEC 模式
步骤 11	**show policy-map** **示例：** Device# **show policy-map**	（可选）显示为所有服务策略配置的所有类别策略配置信息

接下来做什么？

使用 **service-policy** 命令把流量策略关联到一个接口。

为语音和视频配置 class-map（CLI）

用户可以按照以下步骤，来为语音和视频流量配置 class-map。

总步骤

1. configure terminal

2. class-map *class-map-name*

3. match dscp *dscp-value-for-voice*

4. end

5. configure terminal

6. class-map *class-map-name*

7. match dscp *dscp-value-for-video*

8. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**class-map** *class-map-name* **示例：** `Device(config)# class-map voice`	创建一个 class-map
步骤 3	**match dscp** *dscp-value-for-voice* **示例：** `Device(config-cmap)# match dscp 46`	匹配 IPv4 和 IPv6 数据包中的 DSCP 值。把这个值设置为 6
步骤 4	**end** **示例：** `Device(config-cmap)# end`	返回特权 EXEC 模式。或者用户也可以使用 **Ctrl-Z** 退出全局配置模式
步骤 5	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 6	**class-map** *class-map-name* **示例：** `Device(config)# class-map video`	创建一个 class-map
步骤 7	**match dscp** *dscp-value-for-video* **示例：** `Device(config-cmap)# match`	匹配 IPv4 和 IPv6 数据包中的 DSCP 值。把这个值设置为 34

	dscp 34	
步骤 8	**end** **示例：** `Device(config-cmap)# end`	返回特权 EXEC 模式。或者用户也可以使用 **Ctrl-Z** 退出全局配置模式

把流量策略关联到一个接口（CLI）

在创建了流量类别和流量策略后，用户必须使用接口配置命令 **service-policy**，把流量策略关联到接口上，并且指定这个策略应该执行的方向（针对进入接口的数据包，还是针对离开接口的数据包）。

在开始前

用户在能够把流量策略关联到接口前，必须先创建流量类别和流量策略。

总步骤

1. configure terminal

2. interface *type*

3. service-policy {**input** *policy-map* | **output** *policy-map* }

4. end

5. show policy map

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 2	**interface** *type*
	
	示例：
	`Device(config)# interface GigabitEthernet1/0/1`
	`Device(config-if)#`
		进入接口配置模式并对接口进行配置。用户可以在接口配置模式中设置的命令参数如下所示：
		• **Auto Template**——auto-template 接口
		• **Capwap**——CAPWAP 隧道接口
		• **GigabitEhternet**——千兆以太网

		IEEE 802 • **GroupVI**——组虚拟接口 • **Internal Interface**——内部接口 • **Loopback**——环回接口 • **Null**——空接口 • **Port-Channel**——EtherChannel 接口 • **TenGigabitEthernet**——万兆以太网 • **Tunnel**——隧道接口 • **Vlan**——Inspur VLAN • **Range**——接口范围
步骤 3	**service-policy** {**input** *policy-map* | **output** *policy-map*} **示例：** Device(config-if)# **service-policy output policy_map_01** Device(config-if)#	在接口的入方向上或出方向上关联一个 policy-map。这个 policy-map 会被用作这个接口的服务策略。 在示例中，流量策略会评估所有离开接口的流量
步骤 4	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 5	**show policy map** **示例：** `Device# show policy map`	（可选）显示制定接口上策略的状态统计信息

接下来做什么？

继续在接口上应用其他流量策略，并且指定策略应该在哪个方向上执行。

在物理端口上使用 policy-map 实现分类、限速和标记流量（CLI）

用户可以在物理端口上配置非层级式的 policy-map，以此指定要对哪类流量类别实施 QoS 行为。行为包括标记和限速。

在开始前

在开始这部分介绍的配置步骤前，用户应该已经使用 policy-map 确定了网络流量的分类、限速和标记。

总步骤

class-map {*class-map name* | **match-any** }

3. match access-group { *access list index* | *access list name* }

4. policy-map *policy-map-name*

5. class {*class-map-name* | **class-default**}

6. set {**cos** | **dscp** | **ip** | **precedence** | **qos-group** | **wlan user-priority**}

7. police {*target_bit_rate* | **cir** | **rate** }

8. exit

9. exit

10. interface *interface-id*

11. service-policy input *policy-map-name*

12. end

13. show policy-map [*policy-map-name* [**class** *class-map-name*]]

14. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**class-map** {*class-map name* | **match-any**}	进入 class-map 配置模式。 • 创建一个 class-map，用来把数据包匹配到用户定义的类别中

	示例： Device(config)# **class-map ipclass1** Device(config-cmap)# **exit** Device(config)#	• 如果用户指定了 **match-any**，那么流量必须至少与其中一个条件相匹配，这样这个流量才会被分类到这个流量类别中。这是默认设置
步骤 3	**match access-group** {*access list index* | *access list name*} **示例：** Device(config-cmap)# **match access-group 100** Device(config-cmap)# **exit** Device(config)#	指定匹配这个 class-map 的分类条件。 用户可以在这条命令中配置以下参数： • **access-group**——匹配 access-group • **class-map**——匹配另一个 class-map • **cos**——匹配一个 CoS 值 • **dscp**——匹配一个 DSCP 值 • **ip**——匹配指定 IP 值 • **non-client-nrt**——匹配非客户端 NRT • **precedence**——匹配 IPv4 和 IPv6 数据包中的优先级 • **qos-group**——匹配一个 QoS 组 • **vlan**——匹配一个 VLAN
步骤 4	**policy-map***policy-map-name* **示例：** Device(config)# **policy-map flowit** Device(config-pmap)#	通过输入 policy-map 的名称创建一个 policy-map，并进入 policy-map 配置模式。 默认设备中没有定义 policy-map
步骤 5	**class** {*class-map-name* | **class-default**} **示例：** Device(config-pmap)# **class ipclass1** Device(config-pmap-c)#	定义流量分类，并进入 policy-map 类别配置模式。 默认设备中没有定义 policy-map 和 class-map。 如果用户已经通过全局配置命令 **class-map** 定义了流量类别，就可以直接在这个命令中指定 *class-map-name*。

		class-default 流量类别是预定义的，用户可以把它添加到任何策略中。这个流量类别总是会被放在 policy-map 末尾。在 **class-default** 类别中有隐含的 **match any** 语句，所有没有与其他流量类别相匹配的数据包都会匹配 **class-default**

步骤 6	**set** {**cos** | **dscp** | **ip** | **precedence** | **qos-group** | ~~**wlan user-priority**~~}
	
	示例：
	`Device(config-pmap-c)# set dscp 45`
	`Device(config-pmap-c)#`
		（可选）设置 QoS 值。用户可以设置的 QoS 参数如下所示：
		• **cos**——设置 IEEE 802.1Q/ISL 类别的服务/用户优先级
		• **dscp**——设置 IPv4 和 IPv4 数据包中的 DSCP
		• **ip**——设置 IP 特定的值
		• **precedence**——设置 IPv4 和 IPv6 数据包中的优先级值
		• **qos-group**——设置 QoS 组
		• ~~**wlan**——设置 WLAN 用户优先级~~
		在示例中，**set dscp** 命令通过在数据包中设置一个新的 DSCP 值，来分类 IP 流量

步骤 7	**police** {*target_bit_rate* | **cir** | **rate**}
	
	示例：
	`Device(config-pmap-c)#`
	`police 100000 conform-action transmit exceed-action drop`
	`Device(config-pmap-c)#`
		（可选）配置限速器：
		• *target_bit_rate*——输入每秒比特率，输入 8000 至 10000000000 之间的数值
		• **cir**——承诺信息速率
		• **rate**——指定限速速率，为层级式策略使用 PCR，或为单级别 ATM 4.0 限速器策略使用 SCR
		在示例中，**police** 命令把限速器添加到类别中，超出了用户设置的目标比特率（100000）的流量都会被丢弃

步骤 8	**exit**	返回 policy-map 配置模式

	示例： `Device(config-pmap-c)# exit`	
步骤 9	**exit** **示例：** `Device(config-pmap)# exit`	返回全局配置模式
步骤 10	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 2/0/1`	指定要应用 policy-map 的接口并进入接口配置模式。 有效的接口包括物理端口
步骤 11	**service-policy input** *policy-map-name* **示例：** `Device(config-if)# service-policy input flowit`	指定 policy-map 名称，并把它应用到入向端口上。一个入向端口上只支持一个 policy-map
步骤 12	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 13	**show policy-map** [*policy-map-name* [**class** *class-map-name*]] **示例：** `Device# show policy-map`	（可选）确认用户的配置
步骤 14	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

接下来做什么？

如果需要的话，用户可以在 QoS 配置中使用 policy-map，在 SVI 接口上配置分类、限速和标记。

在 SVI 上使用 policy-map 实现分类、限速和标记（CLI）

在开始前

在开始这部分介绍的配置步骤前，用户应该已经使用 policy-map 确定了网络流量的分类、限速和标记。

总步骤

1. configure terminal

2. class-map {*class-map name* | **match-any** }

3. match vlan *vlan number*

4. policy-map *policy-map-name*

5. description *description*

6. class {*class-map-name* | **class-default**}

7. set {**cos** | **dscp** | **ip** | **precedence** | **qos-group** | **wlan user-priority**}

8. police {*target_bit_rate* | **cir** | **rate**}

9. exit

10. exit

11. interface *interface-id*

12. service-policy input *policy-map-name*

13. end

14. show policy-map [*policy-map-name* [**class** *class-map-name*]]

15. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 2	**class-map** {*class-map name* | **match-any**}	进入 class-map 配置模式。 • 创建一个 class-map，用来把数据

	示例： `Device(config)# class-map class_vlan100`	包匹配到用户定义的类别中 • 如果用户指定了 **match-any**，那么流量必须至少与其中一个条件相匹配，这样这个流量才会被分类到这个流量类别中。这是默认设置

步骤 3	**match vlan** *vlan number*
	
	示例：
	`Device(config-cmap)# match vlan 100`
	`Device(config-cmap)# exit`
	`Device(config)#`
		指定匹配到这个 class-map 的 VLAN

步骤 4	**policy-map** *policy-map-name*
	
	示例：
	`Device(config)# policy-map policy_vlan100`
	`Device(config-pmap)#`
		通过输入 policy-map 的名称创建一个 policy-map，并进入 policy-map 配置模式。
		默认设备中没有定义 policy-map

步骤 5	**description** *description* **示例：** `Device(config-pmap)# description vlan 100`	（可选）为 policy-map 输入一个描述信息

步骤 6	**class** {*class-map-name* | **class-default**}
	
	示例：
	`Device(config-pmap)# class class_vlan100`
	`Device(config-pmap-c)#`
		定义流量分类，并进入 policy-map 类别配置模式。
		默认设备中没有定义 policy-map 和 class-map。
		如果用户已经通过全局配置命令 **class-map** 定义了流量类别，就可以直接在这个命令中指定 *class-map-name*。
		class-default 流量类别是预定义的，用户可以把它添加到任何策略中。这个流

		量类别总是会被放在 policy-map 末尾。在 **class-default** 类别中有隐含的 **match any** 语句，所有没有与其他流量类别相匹配的数据包都会匹配 **class-default**

步骤 7	**set** {**cos** | **dscp** | **ip** | **precedence** | **qos-group** | ~~**wlan user-priority**~~}
	
	示例：
	`Device(config-pmap-c)# set dscp af23`
	`Device(config-pmap-c)#`
		（可选）设置 QoS 值。用户可以设置的 QoS 参数如下所示：
		• **cos**——设置 IEEE 802.1Q/ISL 类别的服务/用户优先级
		• **dscp**——设置 IPv4 和 IPv4 数据包中的 DSCP
		• **ip**——设置 IP 特定的值
		• **precedence**——设置 IPv4 和 IPv6 数据包中的优先级值
		• **qos-group**——设置 QoS 组
		• ~~**wlan**——设置 WLAN 用户优先级~~
		在示例中，**set dscp** 命令通过匹配 DSCP 值 AF23（010010），来分类 IP 流量

步骤 8	**police** {*target_bit_rate* | **cir** | **rate**}
	
	示例：
	`Device(config-pmap-c)#`
	`police 200000 conform-action transmit exceed-action drop`
	`Device(config-pmap-c)#`
		（可选）配置限速器：
		• *target_bit_rate*——输入每秒比特率，输入 8000 至 10000000000 之间的数值
		• **cir**——承诺信息速率
		• **rate**——指定限速速率，为层级式策略使用 PCR，或为单级别 ATM 4.0 限速器策略使用 SCR
		在示例中，**police** 命令把限速器添加到类别中，超出了用户设置的目标比特率（200000）的流量都会被丢弃

步骤 9	**exit** **示例：** `Device(config-pmap-c)# exit`	返回 policy-map 配置模式

步骤 10	**exit** **示例：** `Device(config-pmap)# exit`	返回全局配置模式
步骤 11	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/3`	指定要应用 policy-map 的接口并进入接口配置模式。 有效的接口包括物理端口
步骤 12	**service-policy input** *policy-map-name* **示例：** `Device(config-if)# service-policy input policy_vlan100`	指定 policy-map 名称，并把它应用到入向端口上。一个入向端口上只支持一个 policy-map
步骤 13	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 14	**show policy-map** [*policy-map-name* [**class** *class-map-name*]] **示例：** `Device# show policy-map`	（可选）确认用户的配置
步骤 15	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 table-map（CLI）

table-map 是标记特性所使用的配置格式，还使用表格提供了一个字段到另一个字段的映射

和转换。举例来说，用户可以使用 table-map 把二层 CoS 设置映射和转换为三层优先级值。

注释： 多个策略可以同时调用一个 table-map，或者一个策略中可以多次调用一个 table-map。

总步骤

1. configure terminal

2. table-map *name* {**default** {*default value* | **copy** | **ignore**} | **exit** | **map** {**from** *from value* **to** *to value* } | **no**}

3. map from *value* **to** *value*

4. exit

5. exit

6. show table-map

7. configure terminal

8. policy-map

9. class class-default

10. set cos dscp table *table map name*

11. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 2	**table-map** *name* {**default** {*default value*	**copy**	**ignore**}	**exit**	**map** {**from** *from value* **to** *to value* }	**no**}
						
						示例：
						`Device(config)# table-map table01`
						`Device(config-tablemap)#`
							创建一个 table-map 并进入 table-map 配置模式。在 table-map 配置模式中，用户可以输入以下命令：
							• **default**——配置 table-map 的默认值，或者为 table-map 中没有的值设置默认行为：复制或忽略
							• **exit**——退出 table-map 配置模式
							• **map**——在 table-map 中把 *from* 值影射为 *to* 值
							• **no**——反向执行的命令或者恢复

		命令默认值
步骤 3	**map from** *value* **to** *value* **示例：** Device(config-tablemap)# **map from 0 to 2** Device(config-tablemap)# **map from 1 to 4** Device(config-tablemap)# **map from 24 to 3** Device(config-tablemap)# **map from 40 to 6** Device(config-tablemap)# **default 0** Device(config-tablemap)#	在这一步骤中，数据包中的 DSCP 值 0 会被标记为 CoS 值 2，DSCP 值 1 会被标记为 CoS 值 4，DSCP 值 24 会被标记为 CoS 值 3，DSCP 值 40 会被标记为 CoS 值 6，其他值标记为 CoS 值 0。 **注释：** 这个示例使用 policy-map 类别配置命令 **set** 把 CoS 值映射为 DSCP 值
步骤 4	**exit** **示例：** Device(config-tablemap)# **exit** Device(config)#	返回全局配置模式
步骤 5	**exit** **示例：** Device(config) **exit** Device#	返回特权 EXEC 模式
步骤 6	**show table-map** **示例：** Device# **show table-map** Table Map table01 from 0 to 2	显示 table-map 的配置

	from 1 to 4 from 24 to 3 from 40 to 6 default 0	
步骤 7	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式
步骤 8	**policy-map** **示例：** Device(config)# **policy-map table-policy** Device(config-pmap)#	为这个 table-map 配置 policy-map
步骤 9	**class class-default** **示例：** Device(config-pmap)# **class class-default** Device(config-pmap-c)#	把类别匹配到系统默认
步骤 10	**set cos dscp table** *table map name* **示例：** Device(config-pmap-c)# **set cos dscp table table01** Device(config-pmap-c)#	如果用户把这个策略应用在入站端口，那么这个端口上就启用了 DSCP 信任，并且会根据指定 table-map 来进行标记
步骤 11	**end** **示例：** Device(config-pmap-c)# **end** Device#	返回特权 EXEC 模式

接下来做什么？

为用户网络中的 QoS 策略配置其他 policy-map。在创建了 policy-map 后，使用 **service-policy**

命令把一个或多个流量策略关联到接口上。

配置信任特性

配置 QoS 特性和功能

配置呼叫准入控制（CLI）

以下示例展示了如何在设备上为呼叫准入控制（CAC），配置基于类别的无条件数据包标记特性。

总步骤

1. configure terminal

2. class-map *class name*

3. match dscp *dscp value*

4. exit

5. class-map *class name*

6. match dscp *dscp value*

7. exit

8. table-map *name*

9. default copy

10. exit

11. table-map *name*

12. default copy

13. exit

14. policy-map *policy name*

15. class *class-map-name*

16. priority level *level_value*

17. police [*target_bit_rate* | **cir** | **rate**]

18. admit cac wmm-tspec

19. rate *value*

21. exit

22. exit

23. class *class name*

24. priority level *level_value*

25. police [*target_bit_rate* | **cir** | **rate**]

26. admit cac wmm-tspec

27. rate *value*

29. exit

30. exit

31. policy-map *policy name*

32. class *class-map-name*

33. set dscp dscp table *table_map_name*

35. shape average {*target bit rate* | **percent** *percentage*}

36. queue-buffers {**ratio** *ratio value*}

37. service-policy *policy_map_name*

38. end

39. show policy-map

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 2	**class-map** *class name*
	
	示例：
	`Device(config)# class-map voice`
	`Device(config-cmap)#`
		进入策略 class-map 配置模式。指定用户想要创建或更改的类别的名称。策略 class-map 配置模式中的命令选项如下所示：
		• *word*——class-map 名称
		• **class-default**——系统默认类别，匹配所有未分类的数据包

步骤 3	**match dscp** *dscp value*	（可选）匹配 IPv4 和 IPv6 数据

	示例： `Device(config-cmap)# match dscp 46`	包中的 DSCP 值

步骤 4	**exit**
	
	示例：
	`Device(config-cmap)# exit`
	`Device(config)#`
		返回全局配置模式

步骤 5	**class-map** *class name*
	
	示例：
	`Device(config)# class-map video`
	`Device(config-cmap)#`
		进入策略 class-map 配置模式。指定用户想要创建或更改的类别的名称。策略 class-map 配置模式中的命令选项如下所示：
		• *word*——class-map 名称
		• **class-default**——系统默认类别，匹配所有未分类的数据包

步骤 6	**match dscp** *dscp value* **示例：** `Device(config-cmap)# match dscp 34`	（可选）匹配 IPv4 和 IPv6 数据包中的 DSCP 值

步骤 7	**exit**
	
	示例：
	`Device(config-cmap)# exit`
	`Device(config)#`
		返回全局配置模式

步骤 8	**table-map** *name*
	
	示例：
	`Device(config)# table-map dscp2dscp`
	`Device(config-tablemap)#`
		创建一个 table-map 并进入 table-map 配置模式

步骤 9	**default copy**	针对 table-map 中没有的值设置默认行为：复制。

	示例： Device(config-tablemap)# **default copy**	**注释：** 这是默认选项。用户也可以设置 DSCP 值到 DSCP 值的映射
步骤 10	**exit** **示例：** Device(config-tablemap)# **exit** Device(config)#	返回全局配置模式
步骤 11	**table-map** *name* **示例：** Device(config)# **table-map dscp2up** Device(config-tablemap)#	创建一个 table-map 并进入 table-map 配置模式
步骤 12	**default copy** **示例：** Device(config-tablemap)# **default copy**	针对 table-map 中没有的值设置默认行为：复制。 **注释：** 这是默认选项。用户也可以设置 DSCP 值到 UP 值的映射
步骤 13	**exit** **示例：** Device(config-tablemap)# **exit** Device(config)#	返回全局配置模式
步骤 14	**policy-map** *policy name* **示例：** Device(config)# **policy-map ssid_child_cac** Device(config-pmap)#	进入 policy-map 配置模式。 创建或修改一个 policy-map，用户可以把这个 policy-map 关联到一个或多个接口来指定一个服务策略
步骤 15	**class** *class-map-name* **示例：**	定义一个接口级别的流量分类，并进入 policy-map 类别配置模式

	Device(config-pmap)# **class voice**	
步骤 16	**priority level** *level_value* **示例：** Device(config-pmap-c)# **priority level 1**	使用 **priority** 命令为类别分配严格的调度优先级。 **注释：** 优先级等级 1 比优先级等级 2 更重要。QoS 会首先处理优先级等级 1 预留的带宽，因此它的延迟最低。优先级等级 1 和 2 都会预留带宽
步骤 17	**police** [*target_bit_rate* | **cir** | **rate**] **示例：** Device(config-pmap-c)# **police cir 10m**	（可选）配置限速器： • *target_bit_rate*——输入每秒比特率，输入 8000 至 10000000000 之间的数值 • **cir**——承诺信息速率 • **rate**——指定限速速率，为层级式策略使用 PCR，或为单级别 ATM 4.0 限速器策略使用 SCR
步骤 18	**admit cac wmm-tspec**	为这个 policy-map 配置呼叫准入控制
步骤 19	**rate** *value* **示例：** Device(config-pmap-admit-cac-wmm)# **rate 5000**	配置目标比特速率（千比特每秒）。输入 8 至 10000000 之间的数值
~~**步骤 20**~~	~~**wlan-up** *value*~~ ~~**示例：**~~ ~~Device(config-pmap-admit-cac-wmm)# **wlan-up 6 7**~~	~~配置 WLAN UP 值。输入 0 至 7 之间的数值~~
步骤 21	**exit** **示例：**	返回 policy-map 类别配置模式

	Device(config-pmap-admit-cac-wmm)# **exit** Device(config-pmap-c)#	
步骤 22	**exit** **示例：** Device(config-pmap-c)# **exit** Device(config-pmap)#	返回 policy-map 配置模式
步骤 23	**class-map** *class name* **示例：** Device(config)# **class-map video** Device(config-cmap)#	进入策略 class-map 配置模式。指定用户想要创建或更改的类别的名称。策略 class-map 配置模式中的命令选项如下所示： • *word*——class-map 名称 • **class-default**——系统默认类别，匹配所有未分类的数据包
步骤 24	**priority level** *level_value* **示例：** Device(config-pmap-c)# **priority level 2**	使用 **priority** 命令为类别分配严格的调度优先级。 **注释：** 优先级等级 1 比优先级等级 2 更重要。QoS 会首先处理优先级等级 1 预留的带宽，因此它的延迟最低。优先级等级 1 和 2 都会预留带宽
步骤 25	**police** [*target_bit_rate* | **cir** | **rate**] **示例：** Device(config-pmap-c)# **police cir 20m**	（可选）配置限速器： • *target_bit_rate*——输入每秒比特率，输入 8000 至 10000000000 之间的数值 • **cir**——承诺信息速率 • **rate**——指定限速速率，为层级式策略使用 PCR，或为单级别 ATM 4.0 限速器策略使用 SCR

步骤 26	**admit cac wmm-tspec**
	
	示例：
	`Device(config-pmap-c)# admit cac wmm-tspec`
	`Device(config-pmap-admit-cac-wmm)#`
		为这个 policy map 配置呼叫准入控制。
		注释：这条命令只为无线 QoS 配置 CAC

步骤 27	**rate** *value* **示例：** `Device(config-pmap-admit-cac-wmm)# rate 5000`	配置目标比特速率（千比特每秒）。输入 8 至 10000000 之间的数值
步骤 28	**wlan-up** *value* **示例：** `Device(config-pmap-admit-cac-wmm)# wlan-up 4 5`	配置 WLAN UP 值。输入 0 至 7 之间的数值

步骤 29	**exit**
	
	示例：
	`Device(config-pmap-admit-cac-wmm)# exit`
	`Device(config-pmap)#`
		返回 policy-map 配置模式

步骤 30	**exit**
	
	示例：
	`Device(config-pmap)# exit`
	`Device(config)#`
		返回全局配置模式

步骤 31	**policy-map** *policy name*
	
	示例：
	`Device(config)# policy-map ssid_cac`
	`Device(config-pmap)#`
		进入 policy-map 配置模式。
		创建或修改一个 policy-map，用户可以把这个 policy-map 关联到一个或多个接口来指定一个服务策略

步骤 32	**class-map** *class name* **示例：** `Device(config)# class-map default`	定义接口级别的流量分类并进入 policy-map 配置模式。 这个示例把 class-map 设置为默认
步骤 33	**set dscp dscp table** *table_map_name* **示例：** `Device(config-pmap-c)# set dscp dscp table dscp2dscp`	（可选）设置 QoS 值。这个示例使用 **set dscp dscp table** 命令创建一个 table-map 并设定它的值
~~**步骤 34**~~	~~**set wlan user-priority dscp table** *table_map_name*~~ ~~**示例：**~~ ~~`Device(config-pmap-c)# set wlan user-priority dscp table dscp2up`~~	~~（可选）设置 QoS 值。这个示例使用命令 **set wlan user-prority dscp table** 设置了 WLAN 用户优先级~~
步骤 35	**shape average** {*target bit rate* | **percent** *percentage*} **示例：** `Device(config-pmap-c)# shape average 100000000`	配置平均整形速率。用户可以通过设置目标比特速率（比特每秒）来设置平均整形速率，也可以通过设置接口带宽的承诺信息速率（CIR）百分比来设置平均整形速率
步骤 36	**queue-buffers** {**ratio** *ratio value*} **示例：** `Device(config-pmap-c)# queue-buffers ratio 0`	为队列配置相应的缓存大小。 **注释：** 一个策略中配置的所有缓存之和必须小于或等于 100%。未分配的缓存会被平均分到所有剩余队列中
步骤 37	**service-policy** *policy_map_name* **示例：** `Device(config-pmap-c)# service-policy ssid_child_cac`	为服务策略指定 policy-map
步骤 38	**end**	返回特权 EXEC 模式

	示例：
	`Device(config-pmap-c)# end`
	`Device#`	

步骤 39	**show policy-map** **示例：** `Device# show policy-map`	（可选）显示为所有服务策略配置的所有类别策略配置信息

接下来做什么？

为用户网络中的 QoS 策略配置其他 policy-map。在创建了 policy-map 后，使用 **service-policy** 命令把一个或多个流量策略关联到接口上。

更多有关 CAC 的信息，用户可以参考 *System Management Configuration Guide, Inspur INOS (Inspur 6650 Switches)*。

配置带宽（CLI）

下面这个流程解释了如何在用户交换机上配置带宽。

在开始前

在开始这部分介绍的配置步骤前，用户应该已经为带宽创建了 class-map。

总步骤

1. configure terminal

2. policy-map *policy name*

3. class *class name*

4. bandwidth {*Kb/s* | **percent** *percentage* | **remaining** { **ratio** *ratio* }}

5. end

6. show policy-map

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 2	**policy-map** *policy name* **示例：** `Device(config)#` **`policy-map policy_bandwidth01`** `Device(config-pmap)#`	进入 policy-map 配置模式。 创建或修改一个 policy-map，用户可以把它关联到一个或多个接口，用来指定服务策略
步骤 3	**class** *class name* **示例：** `Device(config-pmap)#` **`class class_bandwidth01`** `Device(config-cmap-c)#`	进入策略 class-map 配置模式。指定用户想要创建或更改的类别的名称。策略 class-map 配置模式中的命令选项如下所示： • *word*——class-map 名称 • **class-default**——系统默认类别，匹配所有未分类的数据包
步骤 4	**bandwidth** {*Kb/s* | **percent** *percentage* | **remaining** { **ratio** *ratio* }} **示例：** `Device(config-pmap-c)#` **`bandwidth 200000`** `Device(config-pmap-c)#`	为 policy-map 配置带宽。参数如下所示： • *Kb/s*——千比特每秒，输入 20000 至 10000000 之间的数值 • **percent**——根据百分比为某个类别分配最小带宽。如果其他队列没有用到全部端口带宽的话，这条队列可以超额订阅带宽。总和不能超过 100%，如果小于 100%的话，剩余带宽会被平均分配到所有带宽队列中 • **remaining**——为某个类别分配最小带宽。如果其他队列没有用到全部端口带宽的话，这条队列可以超额订阅带宽。总和不能超过 100%。如果策略中的某条队列上使用了 **priority** 命令，那么用户最好也是用这条命令。用户也可以为每条队列分配速率而不是百分比；这样每条队列就会获得特定的加权，加权值

		与这些比率相关联。比率的取值范围是 0 至 100。在这个示例中，为策略分配的总带宽比率可以超过 100。 **注释：** 用户不能在一个 policy-map 中混用带宽类型。举例来说，用户不能在一个 policy-map 中同时使用带宽百分比和千比特每秒来配置带宽

步骤 5	**end**
	
	示例：
	`Device(config-pmap-c)# end`
	`Device#`
		返回特权 EXEC 模式

步骤 6	**show policy-map** **示例：** `Device# show policy-map`	（可选）显示为所有服务策略配置的所有类别策略配置信息

接下来做什么？

为用户网络中的 QoS 策略配置其他 policy-map。在创建了 policy-map 后，使用 **service-policy** 命令把一个或多个流量策略关联到接口上。

配置限速（CLI）

下面这个示例解释了如何在用户交换机上配置限速特性。

在开始前

在开始这部分介绍的配置步骤前，用户应该已经为限速创建了 class-map。

总步骤

1. configure terminal

2. policy-map *policy name*

3. class *class name*

4. police {*target_bit_rate* [*burst bytes* | **bc** | **conform-action** | **pir**] | **cir** {*target_bit_rate* | **percent** *percentage*} | **rate** {*target_bit_rate* | **percent** *percentage*} **conform-action transmit exceed-action** {**drop** [**violate action**] | **set-cos-transmit** | **set-dscp-transmit** | **set-prec-transmit** |

transmit [**violate action**] }}

5. end

6. show policy-map

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 2	**policy-map** *policy name*
	
	示例：
	`Device(config)# policy-map policy_police01`
	`Device(config-pmap)#`
		进入 policy-map 配置模式。
		创建或修改一个 policy-map，用户可以把它关联到一个或多个接口，用来指定服务策略

步骤 3	**class** *class name*
	
	示例：
	`Device(config-pmap)# class class_police01`
	`Device(config-cmap-c)#`
		进入策略 class-map 配置模式。指定用户想要创建或更改的类别的名称。策略 class-map 配置模式中的命令选项如下所示：
		• *word*——class-map 名称
		• **class-default**——系统默认类别，匹配所有未分类的数据包

步骤 4	**police** {*target_bit_rate* [*burst bytes* | **bc** | **conform-action** | **pir**] | **cir** {*target_bit_rate* | **percent** *percentage*} | **rate** {*target_bit_rate* | **percent** *percentage*} **conform-action transmit exceed-action** {**drop** [**violate action**] | **set-cos-transmit** | **set-dscp-transmit** | **set-prec-transmit** | **transmit** [**violate action**] }}	用户可以在这条命令中配置以下 **police** 子命令： • *target_bit_rate*——比特每秒（取值为 8000 至 10000000000） · *burst bytes*——输入 1000 至 512000000 之间的数值 · **bc**——合格突发 · **conform-action**——当速率低于合格突发时的行为 · **pir**——最高信息速率

	示例：
	`Device(config-pmap-c)# police 8000 conform-action transmit exceed-action drop`
	`Device(config-pmap-c)#`
		• **cir**——承诺信息速率
		· *target_bit_rate*——比特每秒（取值为 8000 至 10000000000）
		· **percent**——接口带宽 CIR 的百分比
		• **rate**——指定限速速率，为层级式策略指定 PCR，或者为单等级 ATM 4.0 限速器策略指定 SCR。
		· *target_bit_rate*——比特每秒（取值为 8000 至 10000000000）
		· **percent**——接口带宽 CIR 的百分比
		用户可以配置以下 **police conform-actiontransmit exceed-action** 子命令选项：
		• **drop**——丢弃数据包
		• **set-cos-transmit**——设置 CoS 值并发送
		• **set-dscp-transmit**——设置 DSCP 值并发送
		• **set-prec-transmit**——重写数据包的优先级并发送
		• **transmit**——传输数据包
		注释： 基于限速器的降低优先级行为只支持使用 table-map。在交换机中，每个标记字段只能有一个降低优先级 table-map

步骤 5	**end** **示例：**	返回特权 EXEC 模式

	Device(config-pmap-c)# **end** Device#	
步骤 6	**show policy-map** **示例：** Device# **show policy-map**	（可选）显示为所有服务策略配置的所有类别策略配置信息

接下来做什么？

为用户网络中的 QoS 策略配置其他 policy-map。在创建了 policy-map 后，使用 **service-policy** 命令把一个或多个流量策略关联到接口上。

配置优先级（CLI）

以下示例解释了如何在用户交换机上配置优先级特性。

用户可以为指定队列分配优先级。用户可以指定两个优先级等级（1 和 2）。

注释： 支持语音和视频的队列应该分配优先级等级 1。

在开始前

在开始这部分介绍的配置步骤前，用户应该已经为优先级创建了 class-map。

总步骤

1. configure terminal

2. policy-map *policy name*

3. class *class name*

4. priority [*Kb/s* [*burst_in_bytes*] | **level** *level_value* [*Kb/s* [*burst_in_bytes*] | **percent** *percentage* [*burst_in_bytes*]] | **percent** *percentage* [*burst_in_bytes*]]

5. end

6. show policy-map

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式
步骤 2	**policy-map** *policy name*	进入 policy-map 配置模式。

	示例： `Device(config)#` **`policy-map policy_priority01`** `Device(config-pmap)#`	创建或修改一个 policy-map，用户可以把它关联到一个或多个接口，用来指定服务策略
步骤 3	**class** *class name* **示例：** `Device(config-pmap)#` **`class class_priority01`** `Device(config-cmap-c)#`	进入策略 class-map 配置模式。指定用户想要创建或更改的类别的名称。策略 class-map 配置模式中的命令选项如下所示： • *word*——class-map 名称 • **class-default**——系统默认类别，匹配所有未分类的数据包
步骤 4	**priority** [*Kb/s* [*burst_in_bytes*] | **level** *level_value* [*Kb/s* [*burst_in_bytes*] | **percent** *percentage* [*burst_in_bytes*]] | **percent** *percentage* [*burst_in_bytes*]] **示例：** `Device(config-pmap-c)#` **`priority level 1`** `Device(config-pmap-c)#`	（可选）用户可以使用 **priority** 命令为这个类别严格指定调度优先级配置选项。命令中包含的选项如下所示： • *Kb/s*——千比特每秒（取值为 1 至 2000000） · *burst_in_bytes*——指定突发字节（取值为 32 至 2000000） • **level** *level_value*——指定多等级（1-2）优先级队列 · *Kb/s*——千比特每秒（取值为 1 至 2000000） · *burst_in_bytes*——指定突发字节（取值为 32 至 2000000） · **percent**——总带宽的百分比 · *burst_in_bytes*——指定突发字节（取值为 32 至 2000000） • **percent**——总带宽的百分比 · 指定突发字节（取值为 32 至

		2000000） **注释：** 优先级等级 1 比优先级等级 2 更重要。QoS 会首先处理优先级等级 1 预留的带宽，因此它的延迟最低。优先级等级 1 和 2 都会预留带宽

步骤 5	**end**
	
	示例：
	`Device(config-pmap-c)# end`
	`Device#`
		返回特权 EXEC 模式

步骤 6	**show policy-map** **示例：** `Device# show policy-map`	（可选）显示为所有服务策略配置的所有类别策略配置信息

接下来做什么？

为用户网络中的 QoS 策略配置其他 policy-map。在创建了 policy-map 后，使用 **service-policy** 命令把一个或多个流量策略关联到接口上。

配置队列和整形

配置出向队列特征

根据用户网络的复杂性和 QoS 解决方案，用户可能需要执行这一部分的所有步骤。用户需要对以下这些特征做出判断：

- 哪些数据包由 DSCP 值、CoS 值或 QoS 组值映射到每条队列和门限值 ID？
- 要对这些队列应用的丢弃百分比门限值是多少，以及需要为流量类别保留的最大内存是多少？
- 要为队列分配多少固定缓存空间？
- 是否要限制端口上的带宽速率？
- 出向队列能够接收服务的频率是多少，以及应该使用哪种技术（整形、共享或两者都使用）？

注释： 用户可以只在交换机上配置出向队列。

配置队列缓存（CLI）

用户可以为队列分配缓存。如果用户没有分配缓存的话，所有缓存会平均分给所有队列。用

户可以使用队列缓存比率（queue-buffer ratio），以特定的比率来分配缓存。默认 DTS（动态门限值与缩放）特性是对所有队列启用的，这些属于软缓存。

注释： 有线端口上支持队列缓存比率（queue-buffer ratio）配置，但队列缓存比率（queue-buffer ratio）不能和队列限制（queue-limit）一起配置。

在开始前

执行这个配置步骤有以下先决条件：

- 在开始这部分介绍的配置步骤前，用户应该已经为队列缓存创建了 class-map；
- 在配置队列缓存前，用户必须已经在 policy-map 中配置了带宽、整形或优先级。

总步骤

1. configure terminal

2. policy-map *policy name*

3. class *class name*

4. bandwidth {*Kb/s* | **percent** *percentage* | **remaining** { **ratio** *ratio value* }}

5. queue-buffers {**ratio** *ratio value*}

6. end

7. show policy-map

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 2	**policy-map** *policy name*
	
	示例：
	`Device(config)# policy-map policy_queuebuffer01`
	`Device(config-pmap)#`
		进入 policy-map 配置模式。
		创建或修改一个 policy-map，用户可以把它关联到一个或多个接口，用来指定服务策略

步骤 3	**class** *class name* **示例：** `Device(config-pmap)# class`	进入策略 class-map 配置模式。指定用户想要创建或更改的类别的名称。策略 class-map 配置模式中的命令选项如下所示：

	class_queuebuffer01 Device(config-cmap-c)#	• *word*——class-map 名称 • **class-default**——系统默认类别，匹配所有未分类的数据包
步骤 4	**bandwidth** {*Kb/s* | **percent** *percentage* | **remaining** { **ratio** *ratio value* }} **示例：** Device(config-pmap-c)# **bandwidth percent 80** Device(config-pmap-c)#	为这个 policy-map 配置带宽。用户可以使用的命令参数如下所示： • *Kb/s*——千比特每秒，输入 20000 至 10000000 之间的数值 • **percent**——根据百分比为某个类别分配最小带宽。如果其他队列没有用到全部端口带宽的话，这条队列可以超额订阅带宽。总和不能超过 100%，如果小于 100%的话，剩余带宽会被平均分配到所有带宽队列中 • **remaining**——为某个类别分配最小带宽。如果其他队列没有用到全部端口带宽的话，这条队列可以超额订阅带宽。总和不能超过 100%。如果策略中的某条队列上使用了 **priority** 命令，那么用户最好也是用这条命令。用户也可以为每条队列分配速率而不是百分比；这样每条队列就会获得特定的加权，加权值与这些比率相关联。比率的取值范围是 0 至 100。在这个示例中，为策略分配的总带宽比率可以超过 100。 **注释：** 用户不能在一个 policy-map 中混用带宽类型
步骤 5	**queue-buffers** {**ratio** *ratio value*} **示例：**	为队列配置适当的缓存大小。 **注释：** 一个策略中配置的缓存总大小必须小于或等于 100%。未分配的缓存会

	Device(config-pmap-c)# **queue-buffers ratio 10** Device(config-pmap-c)#	平均分布给所有剩余队列

步骤 6	**end**
	
	示例：
	`Device(config-pmap-c)# end`
	`Device#`
		返回特权 EXEC 模式

步骤 7	**show policy-map** **示例：** `Device# show policy-map`	（可选）显示为所有服务策略配置的所有类别策略配置信息

接下来做什么？

为用户网络中的 QoS 策略配置其他 policy-map。在创建了 policy-map 后，使用 **service-policy** 命令把一个或多个流量策略关联到接口上。

配置队列限制（CLI）

用户可以使用队列限制来配置加权尾部丢弃（WTD）。WTD 能够为每条队列配置多个门限值。每个服务类别都根据不同的门限值来丢弃数据包，以此提供 QoS 差分服务。在交换机上，每条队列有 3 个用户可以配置的门限值类别——0、1、2。因此每条队列中每个数据包的队列/丢弃决策是由数据包的门限值类别分配结果决定的，这个分配结果是由数据帧头部的 DSCP、CoS 或 QoS 组字段决定的。

WTD 还使用软限制，因此用户能够把队列限制配置为 400%（最大值为普通池中保留缓存的 4 倍）。这个软限制能够防止超越普通池，并且不会影响其他特性。

注释：用户只能在有线端口的出向队列上配置队列限制。

在开始前

执行这个配置步骤有以下先决条件：

- 在开始这部分介绍的配置步骤前，用户应该已经为队列限制创建了 class-map；
- 在配置队列限制前，用户必须已经在 policy-map 中配置了带宽、整形或优先级。

总步骤

1. configure terminal

2. policy-map *policy name*

3. class *class name*

4. bandwidth {*Kb/s* | **percent** *percentage* | **remaining** { **ratio** *ratio value* }}

5. queue-limit {*packets* **packets** | **cos** {*cos value* { *maximum threshold value* | **percent** *percentage* } | **values** {*cos value* | **percent** *percentage* } } | **dscp** {*dscp value* {*maximum threshold value* | **percent** *percentage*} | *match packet* {*maximum threshold value* | **percent** *percentage*} | **default** {*maximum threshold value* | **percent** *percentage*} | **ef** {*maximum threshold value* | **percent** *percentage*} | **dscp values** *dscp value*} | **percent** *percentage* }}

6. end

7. show policy-map

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 2	**policy-map** *policy name*
	
	示例：
	`Device(config)# policy-map policy_queuelimit01`
	`Device(config-pmap)#`
		进入 policy-map 配置模式。
		创建或修改一个 policy-map，用户可以把它关联到一个或多个接口，用来指定服务策略

步骤 3	**class** *class name*
	
	示例：
	`Device(config-pmap)# class class_queuelimit01`
	`Device(config-cmap-c)#`
		进入策略 class-map 配置模式。指定用户想要创建或更改的类别的名称。策略 class-map 配置模式中的命令选项如下所示：
		• *word*——class-map 名称
		• **class-default**——系统默认类别，匹配所有未分类的数据包

步骤 4	**bandwidth** {*Kb/s* | **percent** *percentage* | **remaining** { **ratio** *ratio value* }} **示例：** Device(config-pmap-c)# **bandwidth 500000**	为这个 policy-map 配置带宽。用户可以使用的命令参数如下所示： • *Kb/s*——千比特每秒，输入 20000 至 10000000 之间的数值 • **percent**——根据百分比为某个类别分配最小带宽。如果其他队列没

	Device(config-pmap-c)#	有用到全部端口带宽的话，这条队列可以超额订阅带宽。总和不能超过 100%，如果小于 100%的话，剩余带宽会被平均分配到所有带宽队列中 • **remaining**——为某个类别分配最小带宽。如果其他队列没有用到全部端口带宽的话，这条队列可以超额订阅带宽。总和不能超过 100%。如果策略中的某条队列上使用了 **priority** 命令，那么用户最好也是用这条命令。用户也可以为每条队列分配速率而不是百分比；这样每条队列就会获得特定的加权，加权值与这些比率相关联。比率的取值范围是 0 至 100。在这个示例中，为策略分配的总带宽比率可以超过 100。 **注释：** 用户不能在一个 policy-map 中混用带宽类型
步骤 5	**queue-limit** {*packets* **packets** | **cos** {*cos value* { *maximum threshold value* | **percent** *percentage* } | **values** {*cos value* | **percent** *percentage* } } | **dscp** {*dscp value* {*maximum threshold value* | **percent** *percentage*} | *match packet* {*maximum threshold value* | **percent** *percentage*} | **default** {*maximum threshold value* | **percent** *percentage*} | **ef** {*maximum threshold value* | **percent** *percentage*} | **dscp values** *dscp value*} | **percent** *percentage* }}	设置队列限制门限值的百分比值。 在每条队列中都有三个门限值（0、1、2），每个门限值都有一个默认值。用户可以使用这条命令来修改默认值，或者队列限制门限值的其他设置。举例来说，如果 DSCP 值为 3、4 和 5 的数据包会被发送到一个配置中的指定队列，用户就可以使用这条命令来社为这 3 个 DSCP 值设置门限值百分比。有关队列限制门限值的更多信息，用户可以参考加权尾部丢弃。 **注释：** 交换机不支持使用绝对的队列

	示例： `Device(config-pmap-c)#` **`queue-limit dscp 3 percent 20`** `Device(config-pmap-c)#` **`queue-limit dscp 4 percent 30`** `Device(config-pmap-c)#` **`queue-limit dscp 5 percent 40`**	限制百分比，只支持 DSCP 或 CoS 队列限制百分比
步骤 6	**end** **示例：** `Device(config-pmap-c)#` **`end`** `Device#`	返回特权 EXEC 模式
步骤 7	**show policy-map** **示例：** `Device#` **`show policy-map`**	（可选）显示为所有服务策略配置的所有类别策略配置信息

接下来做什么？

为用户网络中的 QoS 策略配置其他 policy-map。在创建了 policy-map 后，使用 **service-policy** 命令把一个或多个流量策略关联到接口上。

配置整形特性（CLI）

用户可以使用 **shape** 命令来为指定类别配置整形（最大带宽）特性。队列的带宽会被限制为用户配置的值，即使端口还有更多带宽可用。用户可以用平均百分比来配置整形特性，也可以用单位为比特每秒的平均值来配置整形特性。

在开始前

在开始这部分介绍的配置步骤前，用户应该已经为整形特性创建了 class-map。

总步骤

1. configure terminal

2. policy-map *policy name*

3. class *class name*

4. shape average {*target bit rate* | **percent** *percentage*}

5. end

6. show policy-map

具体步骤

步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 2	**policy-map** *policy name*
	
	示例：
	`Device(config)# policy-map policy_shaping01`
	`Device(config-pmap)#`
		进入 policy-map 配置模式。
		创建或修改一个 policy-map，用户可以把它关联到一个或多个接口，用来指定服务策略

步骤 3	**class** *class name*
	
	示例：
	`Device(config-pmap)# class class_shaping01`
	`Device(config-cmap-c)#`
		进入策略 class-map 配置模式。指定用户想要创建或更改的类别的名称。策略 class-map 配置模式中的命令选项如下所示：
		• *word*——class-map 名称
		• **class-default**——系统默认类别，匹配所有未分类的数据包

步骤 4	**shape average** {*target bit rate* | **percent** *percentage*}
	
	示例：
	`Device(config-pmap-c)# shape average percent 50`
	`Device(config-pmap-c)#`
		配置平均整形速率。用户可以用目标比特速率（比特每秒）来配置平均整形速率，或者使用接口带宽承诺信息速率（CIR）百分比来配置平均整形速率。
		注释： 对于出向 class-default SSID 策略来说，用户必须在配置了平均整形速率后，把队列缓存比率配置为 0

步骤 5	**end**
	
	示例：
	`Device(config-pmap-c)# end`
	`Device#`
		返回特权 EXEC 模式

步骤 6	**show policy-map**	（可选）显示为所有服务策略配置的所有类别策略配置信息

	示例： `Device# show policy-map`	

接下来做什么？

为用户网络中的 QoS 策略配置其他 policy-map。在创建了 policy-map 后，使用 **service-policy** 命令把一个或多个流量策略关联到接口上。

监控 QoS

用户可以使用以下命令来监控交换机上的 QoS。

表 99：监控 QoS

命令	描述
show class-map [*class_map_name*]	显示用户配置的所有 class-map 列表
show class-map type control subscriber {**all** | *name* } **show class-map type control subscriber detail**	显示控制 class-map 及其状态统计信息。 • **all——**显示所有 class-map 的信息 • *name*——显示配置的 class-map
show policy-map [*policy_map_name*]	显示用户配置的所有 policy-map 列表。用户可以使用的命令参数如下所示： • **poicy-map 名称** • **接口** • **会话**
show policy-map interface { **Auto-template** | **Capwap** | **GigabitEthernet** |**GroupVI** | **InternalInterface** |**Loopback** |**Lspvif** | | **Null** | **Port-channel** | **TenGigabitEthernet** | **Tunnel** | **Vlan** | **brief** | **class** | **input** | **output** }	显示交换机上配置的所有策略运行时的情况和状态统计信息。用户可以使用的命令参数如下所示： • **Auto Template**——auto-template 接口 • **Capwap**——CAPWAP 隧道接口 • **GigabitEhternet**——千兆以太网 IEEE 802.3z • **GroupVI**——组虚拟接口 • **Internal Interface**——内部接口 • **Loopback**——环回接口 • **Lspvif**——LSP 虚拟接口

	• **Null**——空接口 • **Port-Channel**——EtherChannel 接口 • **TenGigabitEthernet**——万兆以太网 • **Tunnel**——隧道接口 • **Vlan**——Inspur VLAN • **brief**——policy-map 的简要描述信息 • **class**——每个类别的状态统计信息 • **input**——输入策略 • **output**——输出策略
show policy-map session [**input** | **output** | **uid** *UUID*]	描述隧道 QoS 策略。用户可以使用的命令参数如下所示： • **input**——输入策略 • **output**——输出策略 • **uid**——基于 SSS 唯一识别符的策略
show policy-map type control subscriber { **all** | **name** }	显示 QoS policy-map 类型
show table-map	显示所有 table-map 及其配置
show ap name *ap_name* **service-policy**	显示 AP 上配置的所有策略

QoS 的配置示例

示例：使用访问控制列表实现分类

这个示例展示了如何使用访问控制列表（ACL），为 QoS 实现数据包分类：

```
Device# configure terminal
Device(config)# access-list 101 permit ip host 12.4.1.1 host 15.2.1.1
Device(config)# class-map acl-101
Device(config-cmap)# description match on access-list 101
Device(config-cmap)# match access-group 101
Device(config-cmap)#
```

在使用 ACL 创建了一个 class-map 后，用户需要为这个类别创建一个 policy-map，并把这个

policy-map 应用到接口上。

示例：服务类别（CoS）二层分类

这个示例展示了如何使用服务类别（CoS）二层分类特性，为 QoS 实现数据包分类：

```
Device# configure terminal
Device(config)# class-map cos
Device(config-cmap)# match cos ?
<0-7> Enter up to 4 class-of-service values separated by white-spaces
Device(config-cmap)# match cos 3 4 5
Device(config-cmap)#
```

在使用 CoS 二层分类特性创建了一个 class-map 后，用户需要为这个类别创建一个 policy-map，并把这个 policy-map 应用到接口上。

示例：服务类别（CoS）DSCP 分类

这个示例展示了如何使用服务类别（CoS）DSCP 分类特性，为 QoS 实现数据包分类：

```
Device# configure terminal
Device(config)# class-map dscp
Device(config-cmap)# match dscp af21 af22 af23
Device(config-cmap)#
```

在使用 DSCP 分类特性创建了一个 class-map 后，用户需要为这个类别创建一个 policy-map，并把这个 policy-map 应用到接口上。

示例：VLAN ID 二层分类

这个示例展示了如何使用 VLAN ID 二层分类特性，为 QoS 实现数据包分类：

```
Device# configure terminal
Device(config)# class-map vlan-120
Device(config-cmap)# match vlan ?
<1-4095> VLAN id
Device(config-cmap)# match vlan 120
Device(config-cmap)#
```

在使用 VLAN 二层分类特性创建了一个 class-map 后，用户需要为这个类别创建一个 policy-map，并把这个 policy-map 应用到接口上。

示例：使用 DSCP 值或优先级值进行分类

这个示例展示了如何使用 DSCP 值或优先级值来实现数据包分类：

```
Device# configure terminal
Device(config)# class-map prec2
Device(config-cmap)# description matching precedence 2 packets
Device(config-cmap)# match ip precedence 2
Device(config-cmap)# exit
Device(config)# class-map ef
Device(config-cmap)# description EF traffic
Device(config-cmap)# match ip dscp ef
Device(config-cmap)#
```

在使用 DSCP 值或优先级值创建了一个 class-map 后，用户需要为这个类别创建一个 policy-map，并把这个 policy-map 应用到接口上。

示例：层级式分类

这个示例展示了层级式分类，用户创建了名为 parent 的类别，它匹配另一个名为 child 的类别。名为 child 的类别基于 IP 优先级 2 进行匹配。

```
Device# configure terminal
Device(config)# class-map child
Device(config-cmap)# match ip precedence 2
Device(config-cmap)# exit
Device(config)# class-map parent
Device(config-cmap)# match class child
Device(config-cmap)#
```

在使用创建了 class-map parent 后，用户需要为这个类别创建一个 policy-map，并把这个 policy-map 应用到接口上。

示例：层级式策略的配置

这个示例展示了使用层级式策略来配置 QoS 的示例：

```
Device# configure terminal
Device(config)# class-map c1
Device(config-cmap)# match dscp 30
Device(config-cmap)# exit
Device(config)# class-map c2
Device(config-cmap)# match precedence 4
Device(config-cmap)# exit
Device(config)# class-map c3
Device(config-cmap)# exit
Device(config)# policy-map child
Device(config-pmap)# class c1
Device(config-pmap-c)# priority level 1
Device(config-pmap-c)# police rate percent 20 conform-action transmit
exceed action drop
Device(config-pmap-c-police)# exit
Device(config-pmap-c)# exit
Device(config-pmap)# class c2
Device(config-pmap-c)# bandwidth 20000
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# bandwidth 20000
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# policy-map parent
Device(config-pmap)# class class-default
Device(config-pmap-c)# shape average 1000000
Device(config-pmap-c)# service-policy child
Device(config-pmap-c)# end
```

这个示例展示了使用 table-map 来配置层级式策略：

```
Device(config)# table-map dscp2dscp
Device(config-tablemap)# default copy
Device(config)# table-map dscp2up
Device(config-tablemap)# map from 46 to 6
Device(config-tablemap)# map from 34 to 5
Device(config-tablemap)# default copy
Device(config)# policy-map ssid_child_policy
Device(config-pmap)# class voice
Device(config-pmap-c)# priority level 1
Device(config-pmap-c)# police 15000000
Device(config-pmap)# class video
Device(config-pmap-c)# priority level 2
Device(config-pmap-c)# police 10000000
Device(config)# policy-map ssid_policy
Device(config-pmap)# class class-default
Device(config-pmap-c)# shape average 30000000
Device(config-pmap-c)# queue-buffer ratio 0
Device(config-pmap-c)# set dscp dscp table dscp2dscp
Device(config-pmap-c)# service-policy ssid_child_policy
```

示例：为语音和视频进行分类

这个示例描述了如何使用设备指定信息，为语音和视频流中的数据包进行分类。

在这个示例中，语音和视频流量从端点 A 进入设备的 GigabitEthernet1/0/1 接口，分别携带优先级值 5 和 6。除此之外，语音和视频还从端点 B 进入设备的 GigabitEthernet1/0/2 接口，分别携带 DSCP 值 EF 和 AF11。

假设从这两个接口收到的所有数据包都要发往上行链路接口，那么要求用户为语音流量实施 100 Mbit/s 限速，为视频流量实施 150 Mbit/s 限速。

为了按照上述需求进行分类，用户创建了一个类别来匹配从 GigabitEthernet1/0/1 入站的语音数据包，命名为 voice-interface-1，匹配优先级值 5。类似的，创建另一个类别来匹配从 GigabitEthernet1/0/2 入站的语音数据包，命名为 voice-interface-2。这两个类别分别关联到两个不同的策略中，名为 input-interface-1 的策略关联到 GigabitEthernet1/0/1，名为 input-interface-2 的策略关联到 GigabitEthernet1/0/2。用户把这个类别的行为定义为标记 QoS

组值为 10。为了在出站接口匹配 QoS 组值为 10 的数据包，用户创建了名为 voice 的类别，并在其中匹配 QoS 组值 10。这个类别关联到另一个名为 output-interface 的策略中，这个策略关联到上行链路接口上。视频也是使用类似的方法进行处理的，只不过使用 QoS 组值 20。这个示例展示了使用上述设备指定信息来进行数据包分类：

```
Device(config)#
Device(config)# class-map voice-interface-
1 Device(config-cmap)# match ip precedence
5 Device(config-cmap)# exit
Device(config)# class-map video-interface-
1 Device(config-cmap)# match ip precedence
6 Device(config-cmap)# exit
Device(config)# class-map voice-interface-
2 Device(config-cmap)# match ip dscp ef
Device(config-cmap)# exit
Device(config)# class-map video-interface-
2 Device(config-cmap)# match ip dscp af11
Device(config-cmap)# exit
Device(config)# policy-map input-interface-
1 Device(config-pmap)# class voiceinterface-
1 Device(config-pmap-c)# set qosgroup
10 Device(config-pmap-c)# exit
Device(config-pmap)# class video-interface-1
Device(config-pmap-c)# set qos-group 20
Device(config-pmap-c)# policy-map input-interface-2
Device(config-pmap)# class voice-interface-2
Device(config-pmap-c)# set qos-group 10
Device(config-pmap-c)# class video-interface-2
Device(config-pmap-c)# set qos-group 20
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# class-map voice
Device(config-cmap)# match qos-group 10
Device(config-cmap)# exit
```

```
Device(config)# class-map video
Device(config-cmap)# match qos-group 20
Device(config)# policy-map output-interface
Device(config-pmap)# class voice
Device(config-pmap-c)# police 256000 conform-action transmit
exceed-action drop
Device(config-pmap-c-police)# exit
Device(config-pmap-c)# exit
Device(config-pmap)# class video
Device(config-pmap-c)# police 1024000 conform-action transmit
exceed-action drop
Device(config-pmap-c-police)# exit
Device(config-pmap-c)# exit
```

示例：配置下游 SSID 策略

要想配置下游 BSSID 策略，用户必须首先使用优先级等级队列来配置端口子系策略。

策略类型	示例
用户定义的端口子系策略	policy-map port_child_policy class voice priority level 1 20000 class video priority level 2 10000 class non-client-nrt-class bandwidth remaining ratio 10 class class-default bandwidth remaining ratio 15
出向 BSSID 策略	policy-map bssid-policer queue-buffer ratio 0 class class-default shape average 30000000 set dscp dscp table dscp2dscp

	set wlan user-priority dscp table dscp2up service-policy ssid_child_qos
SSID 子系 QoS 策略	Policy Map ssid-child_qos Class voice priority level 1 police cir 5m admit cac wmm-tspec UP 6,7 / tells WCM allow ‘voice’ TSPEC\SIP snoop for this ssid rate 4000 / must be police rate value is in kbps) Class video priority level 2 police cir 60000

示例：入向 SSID 策略

这个示例展示了入向 SSID 层级式策略：

入向 SSID 策略类型	**示例**
入向 SSID 层级式策略	policy-map ssid-child-policy class voice //match dscp 46 police 3m class video //match dscp 34 police 4m policy-map ssid-in-policy class class-default set dscp wlan user-priority table up2dscp service-policy ssid-child-policy
	policy-map ssid_in_policy class dscp-40

	set cos 1 police 10m class up-1 set dscp 34 police 12m class dscp-10 set dscp 20 police 15m class class-default set dscp wlan user-priority table up2dscp police 50m

示例：客户端策略

客户端策略类型	示例/详情
默认出向子系策略	任何入站流量都包含用户优先级 0。 **注释：** 只有在启用了 ACM 的 WMM 客户端上，才默认启用客户端策略。 用户可以使用命令 **show ap dot11 5ghz network** 来确认 ACM 是否已启用。要想启用 ACM，用户可以使用命令 **ap dot11 5ghz cac voice acm**。 Policy-map client-def-down class class-default ~~set wlan user-priority 0~~
基于 AAA 和 TCLAS 的客户端策略	Policy Map client2-down[AAA+ TCLAS pol example] Class voice\\match dscp police <> set <> Class class-default

	set <> Class voice1|| voice2 [match acls] police <> class voice1 set <> class voice2 set <>
出方向上语音和视频流量的客户端策略	Policy Map client3-down class voice \\match dscp, cos police X class video police Y class class-default police Z
使用限速的，入方向上语音和视频流量的客户端策略	Policy Map client1-up class voice \\match dscp, up, cos police X class video police Y class class-default police Z
基于 DSCP 的语音和视频客户端策略	Policy Map client2-up class voice \\match dscp, up, cos set dscp <> class video set dscp <> class class-default set dscp <>
使用标记和限速的客户端入向策略	policy-map client_in_policy class dscp-48 //match dscp 48 set cos 3 police 2m

	class up-4 //match wlan user-priority 4 set dscp 10 police 3m class acl //match acl set cos 2 police 5m class class-default set dscp 20 police 15m
层级式客户端入向策略	policy-map client-child-policy class voice //match dscp 46 set dscp 40 police 5m class video //match dscp 34 set dscp 30 police 7m policy-map client-in-policy class class-default police 15m service-policy client-child-policy

示例：平均速率整形特性的配置

这个示例展示了如何配置平均速率整形特性：

```
Device# configure terminal
Device(config)# class-map prec1
Device(config-cmap)# description matching precedence 1 packets
Device(config-cmap)# match ip precedence 1
Device(config-cmap)# end
Device# configure terminal
Device(config)# class-map prec2
```

```
Device(config-cmap)# description matching precedence 2 packets
Device(config-cmap)# match ip precedence 2
Device(config-cmap)# exit
Device(config)# policy-map shaper
Device(config-pmap)# class prec1
Device(config-pmap-c)# shape average 512000
Device(config-pmap-c)# exit
Device(config-pmap)# policy-map shaper
Device(config-pmap)# class prec2
Device(config-pmap-c)# shape average 512000
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# shape average 1024000
```

在配置了 class-map、policy-map 和整形平均特性后，用户需要把 policy-map 应用在接口上。

示例：队列限制的配置

这个示例展示了如何基于 DSCP 值和百分比，来配置队列限制策略：

```
Device# configure terminal
Device#(config)# policy-map port-queue
Device#(config-pmap)# class dscp-1-2-3
Device#(config-pmap-c)# bandwidth percent 20
Device#(config-pmap-c)# queue-limit dscp 1 percent 80
Device#(config-pmap-c)# queue-limit dscp 2 percent 90
Device#(config-pmap-c)# queue-limit dscp 3 percent 100
Device#(config-pmap-c)# exit
Device#(config-pmap)# class dscp-4-5-6
Device#(config-pmap-c)# bandwidth percent 20
Device#(config-pmap-c)# queue-limit dscp 4 percent 20
Device#(config-pmap-c)# queue-limit dscp 5 percent 30
Device#(config-pmap-c)# queue-limit dscp 6 percent 20
Device#(config-pmap-c)# exit
Device#(config-pmap)# class dscp-7-8-9
```

```
Device#(config-pmap-c)# bandwidth percent 20
Device#(config-pmap-c)# queue-limit dscp 7 percent 20
Device#(config-pmap-c)# queue-limit dscp 8 percent 30
Device#(config-pmap-c)# queue-limit dscp 9 percent 20
Device#(config-pmap-c)# exit
Device#(config-pmap)# class dscp-10-11-12
Device#(config-pmap-c)# bandwidth percent 20
Device#(config-pmap-c)# queue-limit dscp 10 percent 20
Device#(config-pmap-c)# queue-limit dscp 11 percent 30
Device#(config-pmap-c)# queue-limit dscp 12 percent 20
Device#(config-pmap-c)# exit
Device#(config-pmap)# class dscp-13-14-15
Device#(config-pmap-c)# bandwidth percent 10
Device#(config-pmap-c)# queue-limit dscp 13 percent 20
Device#(config-pmap-c)# queue-limit dscp 14 percent 30
Device#(config-pmap-c)# queue-limit dscp 15 percent 20
Device#(config-pmap-c)# end
Device#
```

在完成上述 policy-map 队列限制配置后，用户可以把这个 policy-map 应用到接口上。

示例：队列缓存的配置

这个示例展示了如何配置队列缓存策略并把它应用到接口上：

```
Device# configure terminal
Device(config)# policy-map policy1001
Device(config-pmap)# class class1001
Device(config-pmap-c)# bandwidth remaining ratio 10
Device(config-pmap-c)# queue-buffer ratio ?
<0-100> Queue-buffers ratio limit
Device(config-pmap-c)# queue-buffer ratio 20
Device(config-pmap-c)# end
Device# configure terminal
Device(config)# interface gigabitEthernet2/0/3
```

```
Device(config-if)# service-policy output policy1001
Device(config-if)# end
```

示例：限速行为的配置

下面这个示例展示了可以与限速器关联的各种限速行为。这些行为是通过对合格、超出和违反限速规定的数据包指定不同的配置完成的。用户可以对超出和违反流量分析描述规则的数据包灵活地丢弃、标记和传输，或传输。

举例来说，在一个普通部署环境中，企业客户的限速流量离开自己的网络，去往服务提供商网络，并根据不同的 DSCP 值标记为合格（Conforming）、超出（Exceeding）和违反（Violating）数据包。服务提供商会在拥塞时选择丢弃被标记为超出和违反 DSCP 值的数据包，但会在带宽可用时选择传输这些数据包。

注释： 设备可以对二层字段中的 CoS 字段进行标记，也可以对三层字段中的优先级和 DSCP 字段进行标记。

用户可以使用这样一个有用的特性：把多个行为与一个时间相关联。举例来说，用户可以为所有合格数据包设置优先级比特和 CoS 字段。然后限速特性可以提供一个子模式来配置行为。

限速行为的配置示例如下所示：

```
Device# configure terminal
Device(config)# policy-map police
Device(config-pmap)# class class-default
Device(config-pmap-c)# police cir 1000000 pir 2000000
Device(config-pmap-c-police)# conform-action transmit
Device(config-pmap-c-police)# exceed-action set-dscp-transmit dscp
table exceed-markdown-table
Device(config-pmap-c-police)# violate-action set-dscp-transmit dscp
table
violate-markdown-table
Device(config-pmap-c-police)# end
```

在这个示例中，exceed-markdown-table 和 violate-markdown-table 都是 table-map。

基于线速器的降低优先级标记行为只支持使用 table-map。设备中每个标记字段只能使用一个降低优先级的标记 table-map。

示例：限速器 VLAN 配置

下面这个示例展示了一个 VLAN 限速器的配置。在配置最后，用户在接口上应用了 VLAN policy-map 来实施 QoS 行为。

```
Device# configure terminal
Device(config)# class-map vlan100
Device(config-cmap)# match vlan 100
Device(config-cmap)# exit
Device(config)# policy-map vlan100
Device(config-pmap)# policy-map class vlan100
Device(config-pmap-c)# police 100000 bc conform-action transmit exceed-action drop
Device(config-pmap-c-police)# end
Device# configure terminal
Device(config)# interface gigabitEthernet1/0/5
Device(config-if)# service-policy input vlan100
```

示例：限速单元

这个示例展示了能够为 QoS 提供支持的各种限速单元。限速单元是令牌桶工作的基础。

设备支持的限速单元如下所示：

- 以比特每秒为单位指定 CIR 和 PIR。以字节为单位指定突发参数。这是默认模式；当用户没有指定单元时就会使用这个单元。用户也可以使用百分比来配置 CIR 和 PIR，这时突发参数必须以毫秒为单位进行配置；
- 以数据包每秒为单位指定 CIR 和 PIR。这时突然参数必须也配置为数据包。

以下示例展示了以比特每秒为单位的限速器配置：

```
Device(config)# policy-map bps-policer
Device(config-pmap)# class class-default
Device(config-pmap-c) # police rate 256000 bps burst 1000 bytes conform-action transmit exceed-action drop
```

以下示例展示了以数据包每秒为单位的限速器配置。在这个配置中，用户配置了双速三色限速器，评估单元是数据包。突发和最高突发也都是以数据包为单位指定的。

```
Device(config)# policy-map pps-policer
Device(config-pmap)# class class-default
Device(config-pmap-c)# police rate 5000 pps burst 100 packets
peak-rate 10000 pps peak-burst 200 packets conform-action transmit
exceed-action drop violate-action drop
```

示例：单速双色限速特性的配置

以下示例展示了如何配置单速双色限速器：

```
Device(config)# class-map match-any prec1
Device(config-cmap)# match ip precedence 1
Device(config-cmap)# exit
Device(config)# policy-map policer
Device(config-pmap)# class prec1
Device(config-pmap-c)# police cir 256000 conform-action transmit
exceed-action drop
Device(config-pmap-c-police)# exit
Device(config-pmap-c)#
```

示例：双速三色限速特性的配置

以下示例展示了如何配置双色三色限速器：

```
Device# configure terminal
Device(config)# policy-Map dual-rate-3color-policer
Device(config-pmap)# class class-default
Device(config-pmap-c)# police cir 64000 bc 2000 pir 128000 be 2000
Device(config-pmap-c-police)# conform-action transmit
Device(config-pmap-c-police)# exceed-action set-dscp-transmit dscp
table exceed-markdown-table
Device(config-pmap-c-police)# violate-action set-dscp-transmit dscp
table violate-markdown-table
Device(config-pmap-c-police)# exit
Device(config-pmap-c)#
```

在这个示例中，exceed-markdown-table 和 violate-markdown-table 都是 table-map。

注释： 基于线速器的降低优先级标记行为只支持使用 table-map。设备中每个标记字段只能使用一个降低优先级的标记 table-map。

示例：table-map 标记特性的配置

以下步骤和示例展示了如何在 QoS 配置中使用 table-map 标记特性：

1. 定义一个 table-map；

   使用命令 **table-map** 来定义一个 table-map，并指定数值的映射关系。这个表并不知道它会使用的策略或类别。table-map 中的默认命令指示出当数据包中携带的数值不匹配“From”字段时，就复制用户在“To”字段中配置的值。举例来说，用户创建了名为 table-map1 的 table-map。映射关系中定义了把从 0 到 1、从 2 到 3 的映射，同时把默认值设置为 4。

   ```
   Device(config)# table-map table-map1
   Device(config-tablemap)# map from 0 to 1
   Device(config-tablemap)# map from 2 to 3
   Device(config-tablemap)# default 4
   Device(config-tablemap)# exit
   ```

2. 定义 policy-map 并在其中使用 table-map；

   在这个示例中，QoS 会根据表 table-map1 中指定的映射关系，把入站数据包中的 CoS 值影射为 DSCP 值。举例来说，如果入站数据包的 DSCP 值为 0，那么数据包中的 CoS 值就会被设置为 1。如果用户没有在这里指定 table-map 名称，那么默认行为就是把“From”字段（本例中是 DSCP）的值复制到“To”字段（本例中是 CoS）。但是用户要知道 CoS 是 3 比特字段，DSCP 是 6 比特字段，也就是说 CoS 是复制了 DSCP 中的前 3 个比特。

   ```
   Device(config)# policy map policy1
   Device(config-pmap)# class class-default
   Device(config-pmap-c)# set cos dscp table table-map1
   Device(config-pmap-c)# exit
   ```

3. 把策略关联到一个接口。

   ```
   Device(config)# interface GigabitEthernet1/0/1
   Device(config-if)# service-policy output policy1
   Device(config-if)# exit
   ```

示例：保留 CoS 标记的 table-map 配置

以下示例展示了如何在 QoS 配置中，使用 table-map 来保留接口上的 CoS 标记。
用户在接口的入方向上启用了 cos-rust-policy 策略（配置在示例中），以此来保留从接口进入 CoS 标记。如果用户没有启用这个策略的话，那么默认只会信任 DSCP 值。如果一个纯二层数据包到达了接口，那么当入向端口上没有为 CoS 配置相应策略的话，它的 CoS 值会被重写为 0。

```
Device# configure terminal
Device(config)# table-map cos2cos
Device(config-tablemap)# default copy
Device(config-tablemap)# exit
Device(config)# policy map cos-trust-policy
Device(config-pmap)# class class-default
Device(config-pmap-c)# set cos cos table cos2cos
Device(config-pmap-c)# exit
Device(config)# interface GigabitEthernet1/0/2
Device(config-if)# service-policy input cos-trust-policy
Device(config-if)# exit
```

接下来做什么？

再次查看 Auto-QoS 文档，来确定用户是否可以在 QoS 配置中使用这些自动功能。

Auto-QoS 的其他参考资料

相关文档

相关主题	文档名称
本章中命令的完整语法和用法信息	*QoS Command Reference (Inspur 6650 Switches)* *Inspur INOS Quality of Service Solutions Command Reference*
呼叫准入控制（CAC）	*System Management Configuration Guide (Inspur 6650 Switches)*

	System Management Command Reference (Inspur 6650 Switches)
组播整形和限速	*IP Multicast Routing Configuration Guide (Inspur 6650 Switches)*
应用可见性和控制	*System Management Configuration Guide (Inspur 6650 Switches)* *System Management Command Reference (Inspur 6650 Switches)*
应用可见性和控制	*System Management Configuration Guide (Inspur 6650 Switches)* *System Management Command Reference (Inspur 6650 Switches)*

错误消息解码器

描述	**链接**
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

标准和 RFC

标准/RFC	**标题**
无	-

技术助手

描述	**链接**
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要	http://www.icntnetworks.com

用户在 icntnetworks.com 上注册用户 ID 和密码。	

Auto-QoS 的特性历史与信息

版本	变更
Inspur INOS 11.3.1	引入该特性

第 12 部分 路由

配置双向转发检测

双向转发检测

这个文档描述了如何启用双向转发检测（BFD）协议。BFD 是一项检测协议，用来为所有媒介类型、封装、技术和路由协议提供快速转发路径失效检测。

除了快速转发路径失效检测之外，BFD 为网络管理员提供了一致的失效检测方法。由于网络管理员可以使用 BFD 以单一速率来检测转发路径失效，而不是为不同的路由协议 Hello 机制使用不同的速率，因此网络分析描述和规划工作变得简单了，再收敛时间也是一致且可预测的。

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置双向转发检测的先决条件

- 所有参与的交换机上必须都启用 Inspur 快速转发特性和 IP 路由功能；
- 在部署 BFD 之前，交换机上必须配置一种能够支持 BFD 的 IP 路由协议。用户应该为网络中正在使用的路由协议实施快速收敛特性。用户可以查看交换机使用的 Inspur INOS 软件版本相对应的 IP 路由文档，来查看如何配置快速收敛特性。有关 Inspur INOS 软件中支持的 BFD 路由协议，用户可以参考配置双向转发检测的限制条件一节。

配置双向转发检测的限制条件

- BFD 只能对直连邻居起作用。BFD 邻居必须不能超过 1 跳的距离。不支持多跳配置；
- 不是所有平台和接口都支持 BFD。要想确认某个平台或接口是否支持 BFD，并获得更为详细的平台和硬件限制条件，用户可以按照具体的软件版本查看 Inspur IONS 软件版本注释；
- QoS 策略中无法匹配 BFD 数据包，因为这是交换机自己生成的数据包；
- 用户可以使用命令 **class class-default** 来匹配 BFD 数据包。因此用户必须确保网络中有足够的带宽，防止因为超额订阅而丢弃 BFD 数据包；
- Inspur INOS 中部支持 BFD HA（高可用性）特性。

双向转发检测的相关信息

BFD 的工作原理

BFD 提供了一种低开销和短周期的方法，在两台邻接交换机之间检测转发路径中的失效情况，其中包括接口、数据链路和转发平面。

用户可以在接口级别和路由协议级别启用 BFD 这种检测协议。Inspur 支持 BFD 异步模式，也就是在两个系统之间发送 BFD 控制数据包，来激活并维护两台交换机之间的 BFD 邻居会话。因此，为了创建 BFD 会话，用户必须在两个系统（或两个 BFD 对等体）上都配置 BFD。

一旦用户在接口上启用了 BFD，以及在相应路由协议的路由级别启用了 BFD，BFD 对等体之间就会创建 BFD 会话并协商 BFD 计时器，并且 BFD 对等体之间会以协商出的时间间隔，向对方发送 BFD 控制数据包。

邻居关系

BFD 提供了快速 BFD 对等体失效检测时间，它提供的方法独立于所有媒介类型、封装、技术和路由协议，比如 BGP、EIGRP、IS-IS 和 OSPF。BFD 通过向本地路由器中的路由协议发送快速失效检测通知，触发路由表重计算过程，以此减少了网络收敛总时间。下面这幅图中展示了一个简单的示例网络，其中两台交换机上运行 OSPF 和 BFD。当 OSPF 发现邻居后（1），它会向本地 BFD 进程发送请求，初始化一个去往 OSPF 邻居交换机的 BFD 邻居会话（2）。与 OSPF 邻居交换机之间的 BFD 邻居会话已建立（3）。

Switch A	交换机 A
OSPF neighbors	OSPF 邻居
BFD neighbors	BFD 邻居
Switch B	交换机 B

下图展示了当网络中发生故障时发生的事件（1）。与 OSPF 邻居交换机之间的 BFD 邻居会话断开了（2）。BFD 通知本地 OSPF 进程：BFD 邻居已经不可达（3）。本地 OSPF 进程断开 OSPF 邻居关系（4）。如果网络中有替换路径可用，交换机会马上使用这条链路进行收敛。

Switch A	交换机 A
OSPF neighbors	OSPF 邻居
BFD neighbors	BFD 邻居
Switch B	交换机 B

路由协议需要为它的每个邻居都向 BFD 进行注册。一旦邻居注册后，BFD 就会为这个邻居初始化一个会话，如果这个会话当前不存在的话。

OSPF 会在发生以下事件时向 BFD 进行注册：

- 邻居的有限状态机（FSM）转换到 Full 状态；
- 同时启用了 OSPF BFD 和 BFD。

在广播接口上，OSPF 只会与指定路由器（DR）和备份指定路由器（BDR）建立 BFD 会话，两台 DROTHER 状态的交换机上不会建立 BFD 会话。

BFD 失效检测

一旦 BFD 会话建立起来，并且计时器协商完成后，BFD 对等体之间就会发送 BFD 控制数据包，这种数据包的工作方式与 IGP 中用来检测存活性的 Hello 协议相同，只不过 BFD 控制数据包的速率更快。用户应该关注以下信息：

- BFD 是一项转发路径失效检测协议。BFD 能够检测失效情况，但路由协议必须采取一些行为来绕过失效的对等体；
- 在 Inspur IONS 11.3.1 中，Inspur 设备能够支持 BFD 版本 0，也就是说设备可以为多个客户端协议使用一个 BFD 会话。举例来说，如果网络的部署是通过相同的链路与相同的

对等体之间运行 OSPF 和 EIGRP，那么只需要建立一个 BFD 会话，这个 BFD 会话会共享两个路由协议的会话信息。

BFD 版本之间的互操作性

所有 BFD 会话默认都会使用版本 1 进行建立，并且都能够与版本 0 进行互操作。系统会自动执行 BFD 版本检测，邻居之间的 BFD 会话会运行两者之间都支持的最高版本。举例来说，如果一个 BFD 邻居运行 BFD 版本 0，另一个 BFD 邻居运行版本 1，那么会话就会运行 BFD 版本 0。命令 **show bfd neighbors** [**details**]的输出信息中会确认 BFD 邻居所运行的 BFD 版本。用户可以在以下示例中查看 BFD 版本检测的示例：EIGRP 网络中的 BFD 配置，默认启用 Echo 模式。

BFD 会话限制

在 Inspur INOS 11.3.1 中，用户能够创建的 BFD 会话数量增加到了 100。

为非广播媒介接口提供 BFD

在 Inspur INOS 11.3.1 中，路由端口、SVI 接口和三层 PortChannel 端口上能够支持 BFD 特性。用户必须在接口上配置 **bfd interval** 才能初始化 BFD 监控功能。

为状态化切换的无间断转发功能提供 BFD

通常当网络设备重启时，这台设备的所有路由对等体都会检测到这台设备失效了，然后又启动了。这种状态的过度会导致路由翻动，会影响多个路由域。由路由功能重启导致的路由翻动会带来路由不稳定性，这对于整个网络性能都是有害的。无间断转发（NSF）会帮助抑制设备中由状态化切换（SSO）带来的路由翻动，由此减少网络的不稳定性。

NSF 能够在切换后，根据恢复的路由协议信息，按照已知路由来转发数据包。通过使用 NSF，对等体联网设备不会经历路由翻动。在切换期间，备用 RP 会从失效的主用 RP 那里接管控制权，并通过智能线卡或双转发处理器来转发数据流量。NSF 工作原理的关键之处在于：在切换期间，线卡和转发处理器是保持工作状态的，并且它能够保留在活跃的 RP 上保留当前的转发信息库（FIB）。

在支持双 RP 的设备中，SSO 会与把其中一个 RP 设置为活跃处理器，把另一个 RP 设置为备用处理器，并且在它们之间同步信息。当活跃 RP 失效时、当这它从联网设备上移除时，或者当用户手动移除进行维护时，就会发生从活跃 RP 到备用 RP 的切换过程。

为状态化切换提供 BFD 支持

BFD 协议能够在邻接的转发引擎之间的路径上提供短期失效检测。在使用双 RP 路由器或交换机（来提供冗余）的网络部署中，路由器拥有平滑重启（Graceful Restart）机制，能够在活跃 RP 和备用 RP 之间进行切换时，对转发状态提供保护。

双 RP 环境中的切换时间取决于硬件性能，也就是用来检测通信故障的硬件。当 RP 上运行了 BFD 时，有些平台无法在 BFD 协议超时前，检测到切换事件；这些平台被称为慢切换平

台。

为静态路由提供 BFD

与动态路由协议（比如 OSPF 和 BGP）不同，静态路由不具备对等体发现手段。因此在配置 BFD 时，网关可达性完全取决于 BFD 到指定邻居的会话状态。除非 BFD 会话是启用（Up）的，否则这条静态路由所使用的网关就会被 BFD 认为是不可达的，因此这条路由也不会被放入到相应的路由信息库（RIB）中。

要想成功建立 BFD 会话，用户必须在对等体设备的接口上配置 BFD，并且对等体设备上必须有使用 BFD 邻居地址进行注册的 BFD 客户端。当一个接口用于动态路由协议时，用户可以通过在每个 BFD 邻居上配置路由协议实例，来满足后一个要求。当一个接口只用于静态路由时，用户必须通过在对等体设备上配置静态路由来满足这一需求。

如果当 BFD 会话为启用（Up）状态时，用户移除了对等体设备上的 BFD 配置，BFD 会话的更新状态并不会影响 IPv4 静态路由。这会导致静态路由仍保留在 RIB 中。唯一的解决方案是移除 IPv4 静态 BFD 邻居的配置，这样一来静态路由就不会再追踪 BFD 会话的状态。而且，如果用户在变更串行接口的封装类型时，把它修改为 BFD 不支持的封装类型，这个接口上的 BFD 会话状态就会变为失效（Down）状态。解决方案是手动关闭这个接口，更换一种 BFD 可以支持的封装类型，之后再重新配置 BFD。

IPv4 静态客户端可以使用单条 BFD 会话，通过指定接口追踪下一跳的可达性。用户可以为多个由 BFD 追踪的静态路由分配一个 BFD 组。每个组必须有一个主动的静态 BFD 配置，有一个或多个被动的 BFD 配置，其中指定的惊天路由可以由 BFD 进行追踪。不在组中的条目是 BFD 追踪的静态路由，这些静态路由并没有被分配到 BFD 组中。BFD 组必须符合静态 BFD 配置，静态 BFD 配置可以是不同 VRF 中的一部分。事实上，被动静态 BFD 配置并不一定要与主动配置同属于一个 VRF 中。

对于每个 BFD 组来说，可以只有一个主动的静态 BFD 会话。用户可以通过添加静态 BFD 配置，以及使用这个 BFD 配置的相应静态路由，来配置主动 BFD 会话。只有当用户在 BFD 组中配置了主动的静态 BFD 配置，以及使用这个静态 BFD 配置的静态路由，这个组中的 BFD 会话才会被创建。当用户从 BFD 组中移除了主动的静态 BFD 配置，或者主动的静态路由，所有被动的静态路由就会从 RIB 中撤回。事实上，所有被动的静态路由都是失效（Inactive）状态，直到这个组中配置了主动的静态 BFD 配置，以及配置了使用这个主动 BFD 会话进行追踪的静态路由。

类似的，对于每个 BFD 组来说，可以有一个或多个被动的静态 BFD 配置，以及由这些 BFD 进行追踪的相应静态路由。被动的静态会话路由只会在主动的 BFD 会话状态为可达时才会生效。尽管这个组的主动 BFD 会话状态为可达，也只有当相应的接口状态为启用（Up）时，被动的静态路由才会被添加到 RIB 中。当用户从组中移除了被动的 BFD 会话时，它不会影响

已有的主动 BFD 会话（如果有的话），也不会影响 BFD 组可达性状态。

使用 BFD 进行故障检测的优势

在用户部署任何特性时，都要考虑所有相关问题，并且一定要权衡利与弊。

与 BFD 关系最密切的问题是常规 EIGRP、IS-IS 和 OSPF 的部署中都使用为 EIGRP、IS-IS 和 OSPF 路由协议修改的失效检测机制。

如果用户把 EIGRP Hello 和保持（Hold）计时器设置为具体的分钟值，EIGRP 的失效检测速率就会是 1 至 2 秒的范围内。

如果用户为 IS-IS 或 OSPF 使用快速 Hello，这些内部网关协议（IGP）协议就会把它们的失效检测机制降低为至少 1 秒钟。

鉴于路由协议能够使用缩减了的计时器机制，部署 BFD 拥有以下好处：

- 尽管 EIGRP、IS-IS 和 OSPF 计时器都可以把失效检测时间最小化到 1 至 2 秒钟之内，但 BFD 可以提供小于 1 秒钟的失效检测；
- 由于 BFD 没有与任意一个具体的路由协议相绑定，因此它可以作为通用且一致的失效检测机制，用在 EIGRP、IS-IS 和 OSPF 环境中；
- 由于 BFD 中的一些部分可以分布到数据平面中，因此它可以比 EIGRP、IS-IS 和 OSPF（缩减的）计时器消耗更少的 CPU 资源，这些路由协议的计时器总是运行在控制平面。

如何配置双向转发检测

在接口上配置 BFD 会话参数

要想在接口上配置 BFD，用户需要在接口上设置 BFD 会话基线参数。用户需要在每个希望与 BFD 邻居运行 BFD 会话的接口上，重复以下步骤来设置 BFD 会话基线参数。

总步骤

1. enable

2. configure terminal

3. 配置以下命令之一：

• **ip address** *ipv4-address mask*

• **ipv6 address** *ipv6-address*/*mask*

4. bfd interval *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*

5. end

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式 • 在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 3	执行以下步骤之一：
	• **ip address** *ipv4-address mask*
	• **ipv6 address** *ipv6-address*/*mask*
	
	示例：
	为接口配置 IPv4 地址：
	`Device(config-if)# ip address 10.201.201.1 255.255.255.0`
	为接口配置 IPv6 接口：
	`Device(config-if)# ipv6 address 2001:db8:1:1::1/32`
		为接口配置 IP 地址

步骤 4	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* **示例：** `Device(config-if)# bfd interval 100 min_rx 100 multiplier 3`	在接口上启用 BFD。 如果在子接口上配置了 BFD **interval** 配置，那么当子接口被移除时，BFD 的配置也会被移除。 在以下事件发生时，不会移除 BFD **interval** 配置： • 接口的 IPv4 地址被移除 • 接口的 IPv6 地址被移除 • 接口上的 IPv6 被禁用

		• 接口被关闭 • 在全局禁用 IPv4 CEF，或在接口禁用 IPv4 CEF • 在全局禁用 IPv6 CEF，或在接口禁用 IPv6 CEF
步骤 5	**end** **示例：** `Device(config-if)# end`	退出接口配置模式并返回特权 EXEC 模式

为动态路由协议配置 BFD

为 eBGP 配置 BFD

这部分描述了如何配置 BFD 来为 BGP 提供支持，使 BGP 成为向 BFD 注册的协议，并从 BFD 接收转发路径检测失效消息。

在开始前

所有参与的交换机上必须运行 eBGP。

用户必须在所有希望与 BFD 邻居建立 BFD 会话的接口上配置 BFD 会话基线参数。用户可以从在接口上配置 BFD 会话参数一节获得更多信息。

注释： 命令 **show bfd neighbors details** 中展示了用户配置的间隔。

总步骤

1. enable

2. configure terminal

3. router bgp *as-tag*

4. neighbor *ip-address* **fall-over bfd**

5. end

6. show bfd neighbors [**details**]

7. show ip bgp neighbor

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式 • 在提示时输入密码

	示例： `Device> enable`	
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**router bgp** *as-tag* **示例：** `Device(config)# router bgp tag1`	指定 BGP 进程并进入路由器配置模式
步骤 4	**neighbor** *ip-address* **fall-over bfd** **示例：** `Device(config-router)# neighbor 172.16.10.2 fall-over bfd`	启用 BFD 来支持故障切换
步骤 5	**end** **示例：** `Device(config-router)# end`	退出路由器配置模式并返回特权 EXEC 模式
步骤 6	**show bfd neighbors** [**details**] **示例：** `Device# show bfd neighbors detail`	（可选）确认 BFD 邻居是否为主动状态，并且显示已注册 BFD 的路由信息
步骤 7	**show ip bgp neighbor** **示例：** `Device# show ip bgp neighbor`	（可选）显示邻居的 BGP 和 TCP 连接信息

为 EIGRP 配置 BFD

这一部分描述了如何为 EIGRP 配置 BFD，使 EIGRP 成为向 BFD 注册的协议，并从 BFD 接收转

发路径检测失效消息。用户可以使用两种方法来为 EIGRP 启用 BFD：

- 用户可以在路由器配置模式中，使用命令 **bfd all-interfaces** 为所有运行 EIGRP 的接口都启用 BFD；
- 用户可以在路由器配置模式中，使用命令 **bfd interface** *type number* 为一部分接口启用 BFD。

在开始前

所有参与的交换机上必须运行 EIGRP。

用户必须在所有希望与 BFD 邻居建立 BFD 会话的接口上配置 BFD 会话基线参数。用户可以从在接口上配置 BFD 会话参数一节获得更多信息。

注释： 命令 **show bfd neighbors details** 中展示了用户配置的间隔。

总步骤

1. enable

2. configure terminal

3. router eigrp *as-number*

4. 配置以下命令之一：

• bfd all-interfaces

• bfd interface *type number*

5. end

6. show bfd neighbors [**details**]

7. show ip eigrp interfaces [*type number*] [*as-number*] [**detail**]

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式 • 在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**router eigrp** *as-number*	配置 EIGRP 路由进程并进入路由器配置模式

	示例： `Device(config)# router eigrp 123`	

步骤 4	配置以下命令之一：
	• **bfd all-interfaces**
	• **bfd interface** *type number*
	
	示例：
	`Device(config-router)# bfd all-interfaces`
	示例：
	`Device(config-router)# bfd interface GigabitFastEthernet 1/0/1`
		在与 EIGRP 路由进程相关联的所有接口上全局启用 BFD
		或者
		在与 EIGRP 路由进程相关联的一个或多个接口上，以接口为基础启用 BFD

步骤 5	**end** **示例：** `Device(config-router)# end`	退出路由器配置模式并返回特权 EXEC 模式
步骤 6	**show bfd neighbors** [**details**] **示例：** `Device# show bfd neighbors detail`	（可选）确认 BFD 邻居是否为主动状态，并且显示已注册 BFD 的路由信息
步骤 7	**show ip eigrp interfaces** [*type number*] [*as-number*] [**detail**] **示例：** `Device# show ip eigrp interfaces detail`	（可选）显示某个接口上是否针对 EIGRP 启用了 BFD

为 IS-IS 配置 BFD

这一部分描述了如何为 IS-IS 配置 BFD，使 IS-IS 成为向 BFD 注册的协议，并从 BFD 接收转发路径检测失效消息。用户可以使用两种方法来为 IS-IS 启用 BFD：

- 用户可以在路由器配置模式中，使用命令 **bfd all-interfaces** 为所有运行 IS-IS 的接口都启

用 BFD。然后用户可以使用接口配置模式的命令 **isis bfd disable** 来为这些接口中的一个或多个禁用 BFD；

- 用户可以在接口配置模式中，使用命令 **isis bfd** 为运行 IS-IS 的一部分接口启用 BFD。

要想为 IS-IS 配置 BFD，用户可以执行以下步骤：

先决条件：

所有参与的交换机上必须运行 IS-IS

用户必须在所有希望与 BFD 邻居建立 BFD 会话的接口上配置 BFD 会话基线参数。用户可以从在接口上配置 BFD 会话参数一节获得更多信息。

注释： 命令 **show bfd neighbors details** 中展示了用户配置的间隔。命令的输出中不会显示变更的间隔，因为用户在为硬件中运行的 BFD 会话配置发送（Tx）和接收（Rx）间隔时，间隔不是 50 毫秒的倍数。

在所有接口上为 IS-IS 配置 BFD

用户可以按照这部分展示的步骤，在所有支持 IPv4 路由的 IS-IS 接口上配置 BFD。

总步骤

1. enable

2. configure terminal

3. router isis *area-tag*

4. bfd all-interfaces

5. exit

6. interface *type number*

7. ip router isis [*tag*]

8. isis bfd [**disable**]

9. end

10. show bfd neighbors [**details**]

11. show clns interface

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式 • 在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device# configure terminal`	
步骤 3	**router isis** *area-tag* **示例：** `Device(config)# router isis tag1`	指定 IS-IS 进程并进入路由器配置模式
步骤 4	**bfd all-interfaces** **示例：** `Device(config-router)# bfd all-interfaces`	在所有关联了 IS-IS 路由进程的接口上全局启用 BFD
步骤 5	**exit** **示例：** `Device(config-router)# exit`	（可选）退出路由其配置模式并返回全局配置模式
步骤 6	**interface** *type number* **示例：** `Device(config)# interface fastethernet 6/0`	（可选）进入接口配置模式
步骤 7	**ip router isis** [*tag*] **示例：** `Device(config-if)# ip router isis tag1`	（可选）在接口上启用 IPv4 路由
步骤 8	**isis bfd** [**disable**] **示例：** `Device(config-if)# isis bfd`	（可选）以接口为基础，在一个或多个关联了 IS-IS 路由进程的接口上启用或禁用 BFD。 **注释：** 只有用户已经在与 IS-IS 相关联的所有接口上使用命令 **bfd**

		all-interfaces 启用了 BFD 时，才应该使用 **disable** 关键字
步骤 9	**end** **示例：** `Device(config-if)# end`	退出接口配置模式并返回特权 EXEC 模式
步骤 10	**show bfd neighbors** [**details**] **示例：** `Device# show bfd neighbors detail`	（可选）确认 BFD 邻居是否为主动状态，并且显示已注册 BFD 的路由信息
步骤 11	**show clns interface** **示例：** `Device# show clns interface`	（可选）显示关联了 IS-IS 的接口上是否启用了 BFD

在一个或多个接口上为 IS-IS 配置 BFD

用户可以使用以下步骤，在一个或多个 IS-IS 接口上配置 BFD。

总步骤

1. enable

2. configure terminal

3. interface *type number*

4. ip router isis [*tag*]

5. isis bfd [disable]

6. end

7. show bfd neighbors [**details**]

8. show clns interface

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式 • 在提示时输入密码

步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *type number* **示例：** `Device(config)# interface fastethernet 6/0`	进入接口配置模式
步骤 4	**ip router isis** [*tag*] **示例：** `Device(config-if)# ip router isis tag1`	（可选）在接口上启用 IPv4 路由
步骤 5	**isis bfd** [**disable**] **示例：** `Device(config-if)# isis bfd`	（可选）以接口为基础，在一个或多个关联了 IS-IS 路由进程的接口上启用或禁用 BFD。 **注释：** 只有用户已经在与 IS-IS 相关联的所有接口上使用命令 **bfd all-interfaces** 启用了 BFD 时，才应该使用 **disable** 关键字
步骤 6	**end** **示例：** `Device(config-if)# end`	退出接口配置模式并返回特权 EXEC 模式
步骤 7	**show bfd neighbors** [**details**] **示例：** `Device# show bfd neighbors detail`	（可选）确认 BFD 邻居是否为主动状态，并且显示已注册 BFD 的路由信息
步骤 8	**show clns interface**	（可选）显示关联了 IS-IS 的接口上是否启用了 BFD

	示例： `Device# show clns interface`	

为 OSPF 配置 BFD

这一部分描述了如何为 OSPF 配置 BFD，使 OSPF 成为向 BFD 注册的协议，并从 BFD 接收转发路径检测失效消息。用户可以在所有关联到 OSPF 的接口上全局配置 BFD，也可以在一个或多个接口上有选择地启用 BFD。

用户可以使用以下两种方法来为 OSPF 启用 BFD：

- 用户可以在路由器配置模式中，使用命令 **bfd all-interfaces** 为所有运行 OSPF 的接口都启用 BFD。然后用户可以使用接口配置模式的命令 **ip ospf bfd** [**disable**]来为这些接口中的一个或多个禁用 BFD；
- 用户可以在接口配置模式中，使用命令 **ip ospf bfd** 为运行 OSPF 的一部分接口启用 BFD。

要想为 OSPF 配置 BFD，用户可以执行以下步骤：

在所有接口上为 OSPF 配置 BFD

用户可以按照这部分展示的步骤，在所有 OSPF 接口配置 BFD。

如果用户不想在所有 OSPF 接口上都配置 BFD，或者希望有选择地在一个或多个接口上配置 BFD，可以通过在一个或多个 OSPF 接口上配置 BFD 一节中查看更多信息。

在开始前

所有参与的交换机上必须运行 OSPF。

用户必须在所有希望与 BFD 邻居建立 BFD 会话的接口上配置 BFD 会话基线参数。用户可以从在接口上配置 BFD 会话参数一节获得更多信息。

总步骤

1. enable

2. configure terminal

3. router ospf *process-id*

4. bfd all-interfaces

5. exit

6. interface *type number*

7. ip ospf bfd [**disable**]

8. end

9. show bfd neighbors [**details**]

10. show ip ospf

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式 • 在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**router ospf** *process-id* **示例：** `Device(config)# router ospf 4`	指定 OSPF 进程并进入路由器配置模式
步骤 4	**bfd all-interfaces** **示例：** `Device(config-router)# bfd all-interfaces`	在所有关联了 OSPF 路由进程的接口上全局启用 BFD
步骤 5	**exit** **示例：** `Device(config-router)# exit`	（可选）退出路由其配置模式并返回全局配置模式。只有当用户希望执行步骤 7 来禁用一个或多个接口，才需要使用这条命令
步骤 6	**interface** *type number* **示例：** `Device(config)# interface fastethernet 6/0`	（可选）进入接口配置模式。只有当用户希望执行步骤 7 来禁用一个或多个接口，才需要使用这条命令
步骤 7	**ip ospf bfd** [**disable**] **示例：** `Device(config-if)# ip ospf bfd disable`	（可选）以接口为基础，在一个或多个关联了 OSPF 路由进程的接口上启用或禁用 BFD。 **注释：** 只有用户已经在与 OSPF 相关联的所有接口上使用命令 **bfd**

		all-interfaces 启用了 BFD 时，才应该使用 **disable** 关键字
步骤 8	**end** **示例：** `Device(config-if)# end`	退出接口配置模式并返回特权 EXEC 模式
步骤 9	**show bfd neighbors** [**details**] **示例：** `Device# show bfd neighbors detail`	（可选）确认 BFD 邻居是否为主动状态，并且显示已注册 BFD 的路由信息
步骤 10	**show ip ospf** **示例：** `Device# show ip ospf`	（可选）显示信息来确认是否为 OSPF 启用了 BFD

在一个或多个接口上为 OSPF 配置 BFD

用户可以使用以下步骤，在一个或多个 OSPF 接口上配置 BFD。

在开始前

所有参与的交换机上必须运行 OSPF。

用户必须在所有希望与 BFD 邻居建立 BFD 会话的接口上配置 BFD 会话基线参数。用户可以从在接口上配置 BFD 会话参数一节获得更多信息。

总步骤

1. enable

2. configure terminal

3. interface *type number*

4. ip ospf bfd [**disable**]

5. end

6. show bfd neighbors [**details**]

7. show ip ospf

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式

	示例： `Device> enable`	• 在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *type number* **示例：** `Device(config)# interface fastethernet 6/0`	（可选）进入接口配置模式
步骤 4	**ip ospf bfd** [**disable**] **示例：** `Device(config-if)# ip ospf bfd`	（可选）以接口为基础，在一个或多个关联了 OSPF 路由进程的接口上启用或禁用 BFD。 **注释：** 只有用户已经在与 OSPF 相关联的所有接口上使用命令 **bfd all-interfaces** 启用了 BFD 时，才应该使用 **disable** 关键字
步骤 5	**end** **示例：** `Device(config-if)# end`	退出接口配置模式并返回特权 EXEC 模式
步骤 6	**show bfd neighbors** [**details**] **示例：** `Device# show bfd neighbors detail`	（可选）确认 BFD 邻居是否为主动状态，并且显示已注册 BFD 的路由信息
步骤 7	**show ip ospf** **示例：** `Device# show ip ospf`	（可选）显示信息来确认是否为 OSPF 启用了 BFD

为 HSRP 配置 BFD

用户可以使用这部分介绍的步骤为热备份路由器协议（HSRP）启用 BFD。用户需要在每个希望与 HSRP 对等体建立 BFD 会话的接口上重复这个配置步骤。

HSRP 默认支持 BFD。如果用户手动为 HSRP 禁用了 BFD，可以在路由器级别，为所有接口全局重新启用 BFD，也可以在接口级别为每个接口启用 BFD。

在开始前

- 所有参与的交换机上必须运行 HSRP；
- 必须启用 Inspur 快速转发。

总步骤

1. enable

2. configure terminal

3. ip cef [**distributed**]

4. interface *type number*

5. ip address *ip-address mask*

6. standby [*group-number*] **ip** [*ip-address* [**secondary**]]

7. standby bfd

8. exit

9. standby bfd all-interfaces

10. exit

11. show standby neighbors

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式 • 在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip cef** [**distributed**]	启用 Inspur 快速转发或分布式 Inspur 快速转发

	示例： `Device(config)# ip cef`	
步骤 4	**interface** *type number* **示例：** `Device(config)# interface fastethernet 6/0`	进入接口配置模式
步骤 5	**ip address** *ip-address mask* **示例：** `Device(config-if)# ip address 10.1.0.22 255.255.0.0`	为接口配置 IP 地址
步骤 6	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **示例：** `Device(config-if)# standby 1 ip 10.0.0.11`	激活 HSRP
步骤 7	**standby bfd** **示例：** `Device(config-if)# standby bfd`	（可选）在接口上为 HSRP 启用 BFD
步骤 8	**exit** **示例：** `Device(config-if)# exit`	退出接口配置模式
步骤 9	**standby bfd all-interfaces** **示例：** `Device(config)# standby bfd`	（可选）在所有接口上为 HSRP 启用 BFD

	all-interfaces	
步骤 10	**exit** **示例：** Device(config)# exit	退出全局配置模式
步骤 11	**show standby neighbors** **示例：** Device# show standby neighbors	（可选）显示有关 BFD 支持的 HSRP 信息

为静态路由配置 BFD

用户可以按照这部分描述的步骤为静态路由配置 BFD。用户需要为每个 BFD 邻居重复相同的配置步骤。用户可以在“示例：为静态路由配置 BFD”部分查看更多信息。

总步骤

1. enable

2. configure terminal

3. interface *type number*

4. 配置以下命令之一：

• **ip address** *ipv4-address mask*

• **ipv6 address** *ipv6-address*/*mask*

5. bfd interval *milliseconds* **mix_rx** *milliseconds* **multiplier** *interval-multiplier*

6. exit

7. ip route static bfd *interface-type interface-number ip-address* [**group** *group-name* [**passive**]]

8. ip route [**vrf** *vrf-name*] *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**dhcp**] [*distance*] [**name** *next-hop-name*] [**permanent** | **track** *number*] [**tag** *tag*]

9. exit

10. show ip static route

11. show ip static route bfd

12. exit

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式 • 在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *type number* **示例：** `Device(config)# interface serial 2/0`	配置一个接口并进入接口配置模式

步骤 4	执行以下步骤之一：
	• **ip address** *ipv4-address mask*
	• **ipv6 address** *ipv6-address*/*mask*
	
	示例：
	为接口配置 IPv4 地址：
	`Device(config-if)# ip address 10.201.201.1 255.255.255.0`
	为接口配置 IPv6 接口：
	`Device(config-if)# ipv6 address 2001:db8:1:1::1/32`
		为接口配置 IP 地址

步骤 5	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* **示例：** `Device(config-if)# bfd`	在接口上启用 BFD。 如果在子接口上配置了 BFD **interval** 配置，那么当子接口被移除时，BFD 的配置也会被移除。 在以下事件发生时，不会移除 BFD **interval** 配置：

	`interval 500 min_rx 500 multiplier 5`	• 接口的 IPv4 地址被移除 • 接口的 IPv6 地址被移除 • 接口上的 IPv6 被禁用 • 接口被关闭 • 在全局禁用 IPv4 CEF，或在接口禁用 IPv4 CEF • 在全局禁用 IPv6 CEF，或在接口禁用 IPv6 CEF
步骤 6	**exit** **示例：** `Device(config-if)# exit`	退出接口配置模式并返回全局配置模式
步骤 7	**ip route static bfd** *interface-type interface-number ip-address* [**group** *group-name* [**passive**]] **示例：** `Device(config)# ip route static bfd TenGigabitEthernet1/0/1 10.10.10.2 group group1 passive`	指定静态路由的 BFD 邻居。 • *interface-type*、*interface-number* 和 *ip-address* 变量都是必需的，因为 BFD 只支持直连邻居
步骤 8	**ip route** [**vrf** *vrf-name*] *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**dhcp**] [*distance*] [**name** *next-hop-name*] [**permanent** | **track** *number*] [**tag** *tag*] **示例：** `Device(config)# ip route 10.0.0.0 255.0.0.0`	指定一个静态路由的 BFD 邻居
步骤 9	**exit**	退出全局配置模式并返回特权 EXEC 模

	示例： `Device(config)# exit`	式
步骤 10	**show ip static route** **示例：** `Device# show ip static route`	（可选）显示静态路由数据库信息
步骤 11	**show ip static route bfd** **示例：** `Device# show ip static route bfd`	（可选）从用户配置的 BFD 组和非组条目中显示有关静态 BFD 配置的信息
步骤 12	**exit** **示例：** `Device# exit`	退出特权 EXEC 模式并返回用户 EXEC 模式

配置 BFD Echo 模式

BFD Echo 模式是默认就启用的，但用户可以禁用该模式，这个模式在每个方向上是独立运行的。

BFD Echo 模式是与异步 BFD 一起工作的。Echo 数据包是由转发引擎发送的，并沿着相同的路径转发回来以便执行检测——对端的 BFD 会话并不参与 Echo 数据包的实际转发。Echo 功能和转发引擎负责这个检测过程；在两个 BFD 邻居之间需要发送的 BFD 控制数据包数量减少了。除此之外，由于转发引擎在测试远端（邻居）系统的转发路径同时，并不把远端系统牵扯进来，因此减少了数据包经历的延迟变量，从而提高了失效检测时间，实现了比 BFD 版本 0 中为 BFD 会话使用 BFD 控制数据包更快的检测时间。

当两边系统上都运行 Echo 模式时（两个 BFD 邻居都运行 Echo 模式），Echo 模式会被描述为非不对称。

先决条件

所有参与的交换机上必须运行 BFD。

在使用 BFD Echo 模式之前，用户必须通过输入命令 **no ip redirects**，禁止交换机发送 Internet 控制消息协议（ICMP）重定向消息，这样做是为了避免出现高 CPU 利用率。

用户必须在所有希望与 BFD 邻居建立 BFD 会话的接口上配置 BFD 会话基线参数。用户可以从在接口上配置 BFD 会话参数一节获得更多信息。

限制条件

BFD Echo 模式无法与单播逆向路径转发（uRPF）配置协同工作。如果启用了 BFD Echo 模式和 uRPF 配置，那么会话会翻动。

禁用非不对称 BFD Echo 模式

这部分配置步骤介绍了如何禁用非不对称 BFD Echo 模式——交换机不会发送 Echo 数据包，也不会转发从邻居交换机那里收到的 BFD Echo 数据包。

用户需要在每台 BFD 路由器上执行这个配置步骤。

总步骤

1. enable

2. configure terminal

3. no bfd echo

4. end

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式 • 在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**no bfd echo** **示例：** `Router(config)# no bfd echo`	禁用 BFD Echo 模式。 • 使用 **no** 形式的命令来禁用 BFD Echo 模式
步骤 4	**end** **示例：** `Router(config)# end`	退出全局配置模式并返回特权 EXEC 模式

创建并配置 BFD 模版

用户可以配置一个 single-hop 模版来指定一系列 BFD 间隔值。BFD 间隔值是 BFD 模版中的一部分，模版中指定了适用于多个接口的参数。

注释： 配置 bfd-template 会禁用 Echo 模式。

配置 single-hop 模版

用户可以使用以下步骤来创建一个 BFD single-hop 模版并配置 BFD 间隔计时器。

总步骤

1. enable

2. configure terminal

3. bfd-template single-hop *template-name*

4. interval min-tx *milliseconds* **min-rx** *milliseconds* **multiplier** *multiplier-value*

5. end

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式 • 在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**bfd-template single-hop** *template-name* **示例：** `Device(config)# bfd-template single-hop bfdtemplate1`	创建一个 single-hop BFD 模版并进入 BFD 配置模式
步骤 4	**interval min-tx** *milliseconds* **min-rx** *milliseconds* **multiplier** *multiplier-value* **示例：**	配置发送和接收 BFD 数据包的间隔，指定连续的 BFD 控制数据包数量，在丢失这个数量的数据包后，BFD 会认为对等体不可达

	Device(bfd-config)# interval min-tx 120 min-rx 100 multiplier 3	
步骤 5	**end** **示例：** Device(bfd-config)# end	退出 BFD 配置模式并返回特权 EXEC 模式

BFD 的监控和排错

这一部分描述了因维护和排错需求，检索 BFD 信息的做法。用户可以按照需要来使用这个配置步骤中的命令，并且可以以任何顺序输入这些命令。

这一部分包含的 BFD 监控和排错信息适用于以下 Inspur 平台：

BFD 的监控和排错

要想在 Inspur 7600 系列路由器上对 BFD 进行监控和排错，用户可以执行以下一个或多个步骤。

总步骤

1. enable

2. show bfd neighbors [**details**]

3. debug bfd [**packet** | **event**]

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** Device> enable	进入特权 EXEC 模式 • 在提示时输入密码
步骤 2	**show bfd neighbors** [**details**] **示例：** Router# show bfd neighbors details	（可选）显示 BFD 邻接数据库。 • **details** 关键字显示了每个邻居的所有 BFD 协议参数和计时器
步骤 3	**debug bfd** [**packet**	**event**]	（可选）显示有关 BFD 数据包的调试信

	示例： `Router# debug bfd packet`	息

配置 MSDP

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置 MSDP 的相关信息

这部分描述了如何在交换机上配置组播源发现协议（MSDP）。MSDP 与多个协议无关组播稀疏模式（PIM-SM）域相连。

MSDP 在这个软件版本中并不是完全支持的，因为这个软件版本不支持组播边界网关协议（MBGP），而 MBGP 与 MSDP 的关系紧密。但如果不运行 MBGP 的话，用户也可以创建默认对等体来支持 MSDP。

注释： 要想使用这个特性，活跃的交换机上必须运行 IP Services 特性集。

MSDP 概述

MSDP 能够使不同域中的所有汇集点（RP）都知道某个组的组播源信息。每个 PIM-SM 域都使用自己的 RP，并且不会依赖于其他域中的 RP。一个 RP 可以在传输控制协议（TCP）上运行 MSDP，来发现其他域中的组播源。

一个 PIM-SM 域中的 RP 可以与另一个域中的启用了 MSDP 的设备建立 MSDP 对等体关系。对等体关系是建立在 TCP 连接上的，主要用于交换源列表，这些源会向组播组发送流量。RP 之间的 TCP 连接是通过低层路由系统建立的。接收方 RP 会使用源列表来建立源路径。这个拓扑的目的是为了让一个域中的 RP 能够发现其他域中的组播源。如果一个域中有对某个组播源感兴趣的接收方，组播数据会以普通的方式传递，使用 PIM-SM 环境中的源树建立机制。MSDP 也用于通告向某个组发送数据的源。这些通告必须由域 RP 生成。

MSDP 极度依赖于边界网关协议（BGP）或 MBGP，在域间进行操作。建议用户在自己域中的 RP 上运行 MSDP，这些 RP 负责把向全局组发送数据的源通告到 Internet。

MSDP 的工作原理

当一个源发出第一个组播数据包时，与这个源直连的首跳路由器（*指定路由器*或 RP）会向 RP 发送 PIM 注册消息。RP 会使用这个注册消息来注册一个活跃的源，并沿着本地域中的共享树向下转发组播数据包。如果用户配置了 MSDP，RP 也会向所有 MSDP 对等体转发活跃源（SA，Source-Active）消息。这个 SA 消息中包含了这个源的身份、这个源向哪个组发送数据，以及 RP 的地址或起源 ID（用作 RP 地址的接口 IP 地址），如果配置了的话。

每个 MSDP 对等体都会从起源 RP 那里接收并转发 SA 消息，来完成对等体反向路径泛洪（RPF，Reverse-Path Flooding）。MSDP 设备会查看 BGP 或 MBGP 路由表，来找到应该把起源 RP 发来的 SA 消息转发给哪个下一跳对等体。这个对等体称为 ***RPF 对等体***（反向路径转发对等体）。MSDP 设备会向除 RPF 对等体之外的所有 MSDP 对等体转发这个消息。当设备不支持 BGP 和 MBGP 时，如何配置 MSDP 对等体的更多信息，用户可以参考“配置默认的 MSDP 对等体”。

如果 MSDP 对等体从非 RPF 对等体那里收到了与起源 RP 发来的相同的 SA 消息，它会丢弃这个消息。否则它会把这个消息转发给所有 MSDP 对等体。

一个域中的 RP 会从 MSDP 对等体那里收到 SA 消息。如果 RP 希望加入这个 SA 消息中描述的组，并且(*, G)条目中有一个非空的出向接口列表，那么就表示这个域对这个组感兴趣，并且 RP 会发出一个去往源的(S, G)加入请求。在这个(S, G)加入消息到达源的 DR 后，会从源到远端域中的 RP 建立起一个源树的分支。组播流量现在可以从源，穿越源树转发到 RP 了，

并且会沿着远端域中的共享树转发到接收方。

下图展示了两个 MSDP 对等体之间的 MSDP 操作。PIM 使用 MSDP 作为向一个域中的 RP 进行源注册的标准机制。当用户配置了 MSDP 时，会按序发生以下事件。

图 89：在 RP 对等体之间运行 MSDP

RP + MSDP peer	RP + MSDP 对等体
MSDP peer（共 2 处）	MSDP 对等体
Peer RPF flooding	对等体 RPF 泛洪
Source	源
Multicast	组播
Register	注册
PIM sparse-mode domain	PIM 稀疏模式域
TCP connection BGP	TCP 连接 BGP
(S, G) Join	(S, G)加入
Receiver	接收方

默认情况下，交换机上不会缓存从 SA 消息中收到的源或组对。当交换机转发 MSDP SA 消息时，它不会把相应消息储存在内存中。因此如果一个成员在本地 RP 收到 SA 消息后，马上加入组，这个成员就需要等待，直到 RP 从下一个 SA 消息中收到有关这个源的信息。这种延迟称为加入延迟。

本地 RP 可以针对指定组发送 SA 请求，并马上获得有关这个组的所有活跃源。默认情况下，当一个新成员加入一个组并希望接收组播流量时，交换机并不会向它的 MSDP 对等体发送任何 SA 请求消息。新成员需要等待 RP 收到下一个周期性 SA 消息。

如果用户希望新的组成员从它连接的 PIM 稀疏模式域中学习活跃的组播源（也就是向指定组发送数据的源），用户可以配置交换机当新成员加入一个组时，向指定 MSDP 对等体发送 SA 请求消息。

使用 MSDP 的好处

使用 MSDP 有以下好处：

- 它打破了共享的组播分发树。用户可以把共享树设置为组播域本地的共享树。本地成员可以加入本地树，加入共享树的消息永远无需离开本地域；

- PIM 稀疏模式域可以只依赖于它们自己的 RP，降低了对其他域中 RP 的依赖。这种行为增强了安全性，因为用户可以防止自己的源被本地域之外的设备获知；
- 只包含有接收方的域可以接收数据，而无需在全局范围内通告组成员；
- 不再需要全局源组播路由表状态，节省了内存空间。

如何配置 MSDP

默认的 MSDP 配置

MSDP 没有启用，并且不存在默认的 MSDP 对等体。

配置默认的 MSDP 对等体

在开始前

用户需要配置一个 MSDP 对等体。

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip msdp default-peer** *ip-address* | *name* [**prefix-list** *list*] **示例：** `Router(config)# ip msdp default-peer 10.1.1.1`	定义一个默认的对等体，并让它接受所有 MSDP SA 消息。 • 在 *ip-address* | *name* 部分输入 MSDP 默认对等体的 IP 地址或域名系统（DNS）服务器名称； • （可选）在 **prefix-list** *list* 部分输入

	prefix-list site-a	列表名称，这个列表指定这个对等体只为列表中列出的前缀充当默认对等体。当用户在默认对等体上关联前缀列表时，可以拥有多个活跃的默认对等体。 当用户输入多个带有 **prefix-list** 关键字的命令 **ip msdp default-peer** 时，用户会同时使用所有的默认对等体，每个对等体是不同 RP 前缀的默认对等体。这种配置方式通常用在服务提供商云中，这个云连接了多个末节站点云。 当用户输入多个不带有 **prefix-list** 关键字的命令 **ip msdp default-peer** 时，会由一个活跃的对等体接受所有 SA 消息。如果这个对等体失效了，下一个配置的默认对等体会接受所有 SA 消息。这种配置方式通常用在末节站点中
步骤 4	**ip prefix-list** *name* [**description** *string*] | **seq** *number* {**permit** | **deny**} *network length* **示例：** Router(config)# **prefix-list site-a seq 3 permit 12 network length 128**	（可选）使用步骤 2 中指定的名称创建一个前缀列表。 • （可选）在 **description** *string* 部分输入针对这个前缀列表的描述信息，最多 80 个字符； • 在 **seq** *number* 部分输入这个条目的序号。配置范围是 1 至 4294967294； • 关键字 **deny** 拒绝匹配条件的前缀； • 关键字 **permit** 允许匹配条件的前缀 • 在 *network length* 部分指定要允许

		或拒绝的网络掩码，指定它的网络号和长度（以比特为单位）
步骤 5	**ip msdp description** {*peer-name* | *peer-address*} *text* **示例：** `Router(config)# ip msdp description peer-name site-b`	（可选）为指定对等体配置描述信息，便于在配置中或在 **show** 命令的输出信息中识别这个对等体。 默认情况下，MSDP 对等体上没有描述信息
步骤 6	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 7	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 8	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

缓存活跃源（Source-Active）状态

如果用户希望牺牲掉一些内存，来减少获得源信息的延迟，就可以配置设备来缓存 SA 消息。

用户可以按照以下步骤，让设备缓存源/组对：

用户可以按照以下步骤来配置源/组对的缓存：

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：**	进入特权 EXEC 模式。在提示时输入密码

	Device> **enable**	
步骤 2	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式
步骤 3	**ip msdp cache-sa-state** [**list** *access-list-number*] **示例：** Device(config)# **ip msdp cache-sa-state 100**	启用源/组对的缓存功能（创建一个 SA 状态）。访问列表中允许的源/组对会被缓存。 在 **list** *access-list-number* 部分指定 100 至 199 之间的编号。 **注释：** 全局配置命令 **ip msdp sa-reques** 可以替换这条命令，该命令会在一个组的新成员启动后，让设备向 MSDP 对等体发送 SA 请求消息
步骤 4	**access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* **示例：** Device(config)# **access-list 100 permit ip 171.69.0.0 0.0.255.255 224.2.0.0 0.0.255.255**	创建一个 IP 扩展访问列表，用户可以按需多次重复配置这条命令。 • 在 *access-list-number* 部分输入步骤 2 中创建的编号，取值范围是 100 至 199； • 关键字 **deny** 拒绝匹配条件的源/组对。关键字 **permit** 允许匹配条件的源/组对； • 在 *protocol* 部分输入 **ip** 作为协议名称； • 在 *source* 部分输入数据包来自的络号或主机； • 在 *source-wildcard* 部分以点分十进制格式输入应用在这个源上的通配符比特。把希望忽略的比特位置设置为 1； • 在 *destination* 部分输入数据包发往的网络号或主机；

		• 在 *destination-wildcard* 部分以点分十进制格式输入应用在这个目的上的通配符比特。把希望忽略的比特位置设置为 1。 要记得访问列表总是以隐含的拒绝所有语句结尾
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

从 MSDP 对等体请求源信息

如果用户希望一个组的新成员能够在它连接的 PIM 稀疏模式域中，学到向该组发送数据的活跃组播源信息，可以执行以下步骤，让设备在新成员加入组时，向指定 MSDP 对等体发送 SA 请求消息。对等体会以本地 SA 缓存中的信息进行回复。如果对等体上没有配置缓存，那么这条命令不会有任何效果。配置这个特性可以减少加入延迟，但会耗费内存空间。

用户可以按照以下步骤，让设备在新成员加入组并且希望接收组播流时，向指定 MSDP 对等体发送 SA 请求消息：

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密码

	示例： `Device> enable`	
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip msdp sa-request** {*ip-address* | *name*} **示例：** `Device(config)# ip msdp sa-request 171.69.1.1`	配置设备向指定 MSDP 对等体发送 SA 请求消息。 在 *ip-address* | *name* 部分输入 MSDP 对等体的 IP 地址或名称，当一个组中的新成员启用时，本地设备会向这个对等体发送 SA 请求消息。 如果用户希望设备向多个 MSDP 对等体发送 SA 请求消息，可以针对每个 MSDP 对等体重复配置这条命令
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

控制本地交换机生成的源信息

用户可以控制本地设备生成的组播源信息：

- 本地通告的源（来自于本地源）；

- 源的接收方信息（来自于请求者）。

重新分布源

源所注册的那个 RP 会生成 SA 消息。默认情况下，任何注册在一个 RP 上的源都会被通告。当源注册到 RP 上时，RP 中会设置 *A 标记*，表示这个源已经在一个 SA 中通告出去了，除非用户配置了过滤规则。

用户可以按照以下步骤，进一步限制要 RP 通告哪些注册的源：

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip msdp redistribute** [**list** *access-list-name*] [**asn** *aspath-access-list-number*] [**route-map** *map*] **示例：** `Device(config)# ip msdp redistribute list 21`	配置设备要在 SA 消息中通告组播路由表中的哪个(S, G)条目。 默认情况下，只有本地域中的源会被通告出去。 • （可选）**list** *access-list-name*——输入一个 IP 标准或扩展访问列表的名称或编号。标准访问列表编号的取值范围是 1 至 99，扩展访问列表编号的取值范围是 100 至 199。访问控制列表用来控制设备能够通告哪些本地源，以及这些源向哪些组发送数据； • （可选）**asn** *aspath-access-list-number*——输入

		IP 标准或扩展访问列表编号，范围是 1 至 99。访问列表编号必须与 **ip as-path access-list** 命令中配置的相同； • （可选）**route-map** *map*——输入 IP 标准或扩展访问列表编号，取值范围是 1 至 99。访问列表编号必须与 **ip as-path access-list** 命令中配置的相同。 设备会根据访问列表或自治系统路径访问列表中的规则，来通告(S, G)对

步骤 4	用户可以使用以下命令之一：
	• **access-list** *access-list-number* {**deny| permit**} *source* [*source-wildcard*]
	• ***access-list*** *access-list-number* {**deny| permit**} *protocol source source-wildcard destination destination-wildcard*
	
	示例：
	`Device(config)# access list 21 permit 194.1.22.0`
	或
	`Device(config)# access list 21 permit ip 194.1.22.0 1.1.1.1 194.3.44.0 1.1.1.1`
		创建一个 IP 标准访问列表，用户可以按需多次重复配置这条命令。
		或者
		创建一个 IP 扩展访问列表，用户可以按需多次重复配置这条命令。
		• *access-list-number*——输入步骤 2 中创建的列表编号。标准访问列表编号的取值范围是 1 至 99，扩展访问列表编号的取值范围是 100 至 199；
		• 关键字 **deny**——拒绝匹配条件的源/组对。关键字 **permit** 允许匹配条件的源/组对；
		• *protocol*——输入 **ip** 作为协议名称；
		• *source*——输入数据包来自的网络号或主机；
		• *source-wildcard*——以点分十进制格式输入应用在这个源上的通配符比特。把希望忽略的比特位置设置为 1；

		• *destination*——输入数据包发往的网络号或主机； • *destination-wildcard*——以点分十进制格式输入应用在这个目的上的通配符比特。把希望忽略的比特位置设置为 1。 要记得访问列表总是以隐含的拒绝所有语句结尾
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

过滤活跃源（Source-Active）请求消息

默认情况下，只有缓存了 SA 信息的设备才能够对 SA 请求做出响应。默认情况下，这种设备会对 MSDP 对等体发来的所有 SA 请求消息做出响应，并为其提供活跃源的 IP 地址。

但是，用户可以配置设备来忽略某个 MSDP 对等体发来的所有 SA 请求。也可以配置设备只响应某些 SA 请求消息，按照一个标准访问控制列表中描述的组向对等体进行响应。如果访问列表中允许这个组，设备就会接受这个 SA 请求。从这个对等体发来的有关其他组的类似请求消息都会被忽略。

要想恢复默认设置，用户可以使用全局配置命令 **no ip msdp filter-sa-request** {*ip-address*| *name*}。

用户可以按照以下步骤来配置其中一种选项：

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	用户可以使用以下命令之一： • **ip msdp filter-sa-request** {*ip-address*| *name*} • **ip msdp filter-sa-request** {*ip-address*| *name*} **list** *access-list-number* **示例：** `Device(config)# ip msdp filter sa-request 171.69.2.2`	过滤从指定 MSDP 对等体发来的所有 SA 请求消息。 或者 过滤从指定 MSDP 对等体发来的，有关某个组的 SA 请求消息，放行规则定义在标准访问类别中。访问列表中描述了组播组地址。访问列表编号的取值范围是 1 至 99
步骤 4	**access-list** *access-list-number* {**deny**| **permit**} *source* [*source-wildcard*] **示例：** `Device(config)# access-list 1 permit 192.4.22.0 0.0.0.255`	创建一个 IP 标准访问列表，用户可以按需多次重复配置这条命令。 • 在 *access-list-number* 部分输入 1 至 99 之间的编号； • 关键字 **deny** 拒绝匹配条件的源。关键字 **permit** 允许匹配条件的源； • 在 *source* 部分输入数据包来自的络号或主机； • 在 *source-wildcard* 部分以点分十进制格式输入应用在这个源上的通配符比特。把希望忽略的比特位置设置为 1。

		要记得访问列表总是以隐含的拒绝所有语句结尾
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

控制本地交换机转发的源信息

默认情况下，设备会转发它从所有 MSDP 对等体收到的所有 SA 消息。但用户可以使用过滤器或通过设置生存时间（TTL）值，来阻止设备向对等体转发 SA 消息。

使用过滤器

通过创建一个过滤器，用户可以执行以下行为：

- 过滤所有源/组对；
- 指定一个 IP 扩展访问列表，只放行特定的源/组对；
- 基于一个 route-map 中的匹配条件进行过滤。

用户可以按照以下步骤来应用一个过滤器：

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密码

	示例： `Device> enable`	
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 3	用户可以使用以下命令之一：
	• **ip msdp sa-filter out** {*ip-address* | *name*}
	• **ip msdp sa-filter out** {*ip-address* | *name*} **list** *access-list-number*
	• **ip msdp sa-filter out** {*ip-address* | *name*} **route-map** *map-tag*
	
	示例：
	`Device(config)# ip msdp sa-filter out switch.icntnetworks.com`
	或
	`Device(config)# ip msdp sa-filter out list 100`
	或
	`Device(config)# ip msdp sa-filter out switch.icntnetworks.com route-map 22`
		• 过滤所有去往指定 MSDP 对等体的所以 SA 消息；
		• 只向指定对等体发送 IP 扩展访问列表中放行的 SA 消息。扩展访问列表的编号范围是 100 至 199；
		如果用户同时使用了 **list** 和 **route-map** 关键字，那么出向 SA 消息中的任意(S, G)对必须满足所有条件；
		• 只为指定 MSDP 对等体发送匹配 route-map *map-tag* 中规则的 SA 消息。
		如果使用全部匹配条件，route-map 中的 **permit** 关键字放行路由，**deny** 关键字过滤路由

步骤 4	**access-list** *access-list-number* {**deny| permit**} *protocol source source-wildcard destination destination-wildcard* **示例：** `Device(config)# access list`	（可选）创建一个 IP 扩展访问列表，用户可以按需多次重复配置这条命令。 • 在 *access-list-number* 部分输入步骤 3 中指定的列表编号； • 关键字 **deny** 拒绝匹配条件的源/组对。关键字 **permit** 允许匹配条件

	100 permit ip 194.1.22.0 1.1.1.1 194.3.44.0 1.1.1.1	的源/组对； • 在 *protocol* 部分输入 **ip** 作为协议名称； • 在 *source* 部分输入数据包来自的网络号或主机； • 在 *source-wildcard* 部分以点分十进制格式输入应用在这个源上的通配符比特。把希望忽略的比特位置设置为 1； • 在 *destination* 部分输入数据包发往的网络号或主机； • 在 *destination-wildcard* 部分以点分十进制格式输入应用在这个目的上的通配符比特。把希望忽略的比特位置设置为 1。 要记得访问列表总是以隐含的拒绝所有语句结尾
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

使用 TTL 来限制 SA 消息中发送的组播数据

用户可以使用 TTL 值来控制第一个 SA 消息中为每个源封装的数据。只有 IP 头部的 TTL 值大于或等于 *ttl* 参数的组播数据包会被发送给指定 MSDP 对等体。举例来说，用户可以把内部流量限制为 TTL 8。如果用户希望其他组被通告到外部站点中，就必须以大于 8 的 TTL 来发送这些数据包。

用户可以按照以下步骤来建立 TTL 门限值：

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip msdp ttl-threshold** {*ip-address* | *name*} *ttl* **示例：** `Device(config)# ip msdp ttl-threshold switch.icntnetworks.com 0`	限制为指定 MSDP 对等体发送的第一个 SA 消息中封装哪些组播数据。 • 在 *ip-address* | *name* 部分输入 MSDP 对等体的 IP 地址或名称，用户要向这个对等体应用 TTL 限制； • 在 *ttl* 部分输入 TTL 值。默认值为 0，表示所有携带组播数据的数据包都会被转发给对等体，直到 TTL 耗尽为止。参数配置范围是 0 至 255
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config**	检查用户输入的信息

	示例： `Device# show running-config`	
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

控制本地交换机接收的源信息

默认情况下，设备会接收 MSDP RPF 对等体发送给它的所有 SA 消息。但用户可以通过过滤入站 SA 消息，来控制本地设备从 MSDP 对等体接收的源信息。换句话说，用户可以让设备不接受这些 SA 消息。

通过创建一个过滤器，用户可以执行以下行为：

- 过滤所有源/组对；
- 指定一个 IP 扩展访问列表，只放行特定的源/组对；
- 基于一个 route-map 中的匹配条件进行过滤。

用户可以按照以下步骤来应用一个过滤器：

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	用户可以使用以下命令之一： • **ip msdp sa-filter in** {*ip-address* | *name*} • **ip msdp sa-filter in** {*ip-address* |	• 过滤所有来自指定 MSDP 对等体的所以 SA 消息； • 只从指定对等体接收 IP 扩展访问列表中放行的 SA 消息。扩展访问

	name} **list** *access-list-number* • **ip msdp sa-filter in** {*ip-address* | *name*} **route-map** *map-tag* **示例：** `Device(config)#` **`ip msdp sa-filter in switch.icntnetworks.com`** 或 `Device(config)#` **`ip msdp sa-filter in list 100`** 或 `Device(config)#` **`ip msdp sa-filter in switch.icntnetworks.com route-map 22`**	列表的编号范围是 100 至 199； 如果用户同时使用了 **list** 和 **route-map** 关键字，那么入向 SA 消息中的任意(S, G)对必须满足所有条件； • 只从指定 MSDP 对等体接收匹配 route-map *map-tag* 中规则的 SA 消息。 如果使用全部匹配条件，route-map 中的 **permit** 关键字放行路由，**deny** 关键字过滤路由
步骤 4	**access-list** *access-list-number* {**deny| permit**} *protocol source source-wildcard destination destination-wildcard* **示例：** `Device(config)#` **`access list 100 permit ip 194.1.22.0 1.1.1.1 194.3.44.0 1.1.1.1`**	（可选）创建一个 IP 扩展访问列表，用户可以按需多次重复配置这条命令。 • 在 *access-list-number* 部分输入步骤 3 中指定的列表编号； • 关键字 **deny** 拒绝匹配条件的源/组对。关键字 **permit** 允许匹配条件的源/组对； • 在 *protocol* 部分输入 **ip** 作为协议名称； • 在 *source* 部分输入数据包来自的网络号或主机； • 在 *source-wildcard* 部分以点分十进制格式输入应用在这个源上的通配符比特。把希望忽略的比特位置设置为 1； • 在 *destination* 部分输入数据包发

		往的网络号或主机； • 在 *destination-wildcard* 部分以点分十进制格式输入应用在这个目的上的通配符比特。把希望忽略的比特位置设置为 1。 要记得访问列表总是以隐含的拒绝所有语句结尾
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 MSDP 全互联组

MSDP 全互联组（MSDP Mesh Group）是指一组 MSDP 设备之间两两全互联。从全互联组中任意对等体收到的 SA 消息，都不会转发给相同全互联组中的其他对等体。由此减少了泛洪的 SA 消息数量并简化了对等体 RPF 泛洪。当一个域中有多个 RP 时，用户可以使用全局配置命令 **ip msdp mesh-group**。这条命令尤其适用于跨越域发送 SA 消息的情况。用户可以在一台设备上配置多个全互联组（使用不同的名称）。

用户可以按照以下步骤来创建一个全互联组：

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密

	示例： `Device> enable`	码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip msdp mesh-group** *name* {*ip-address* | *name*} **示例：** `Device(config)# ip msdp mesh-group 2 switch.icntnetworks.com`	配置一个 MSDP 全互联组，并指定属于这个全互联组的 MSDP 对等体。 默认情况下，MSDP 对等体都不属于任何全互联组。 • 在 *name* 部分输入全互联组的名称； • 在 *ip-address* | *name* 部分输入 MSDP 对等体的 IP 地址或名称，这个对等体要成为全互联组中的成员。 用户可以重复配置这条命令，在组中添加多个 MSDP 对等体
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

关闭一个 MSDP 对等体

如果用户希望对同一个对等体配置多条 MSDP 命令，但不希望这个对等体变为活跃模式，就可以先关闭这个对等体、配置它，然后再启用它。当用户关闭一个对等体时，TCP 连接也会终结并且不会重新建立。用户也可以在关闭一个 MSDP 会话的同时保留这个对等体的配置信息。

用户可以按照以下步骤来关闭一个对等体：

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip msdp shutdown** {*peer-name* | *peer-address*} **示例：** `Device(config)# ip msdp shutdown switch.icntnetworks.com`	关闭一个 MSDP 会话，同时保留这个对等体的配置信息。 在 *peer-name* | *peer-address* 部分输入要关闭的 MSDP 对等体 IP 地址或名称
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息

步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

在 MSDP 中包含邻接的 PIM 密集模式区域

用户可以在设备上配置 MSDP，同时这台设备是 PIM 稀疏模式区域和 PIM 密集模式区域的边界。默认情况下，密集模式区域中的活跃源不会参与 MSDP。

注释： 不建议用户使用全局配置命令 **ip msdp border sa-address**。最好用稀疏模式域中的边界路由器作为代理，向 RP 注册密集模式域中源，并且稀疏模式域使用标准的 MSDP 过程来通告这些源。

全局配置命令 **ip msdp originator-id** 也指定了一个接口，并将其当作 RP 地址。如果用户同时配置了全局配置命令 **ip msdp border sa-address** 和 **ip msdp originator-id**，那么命令 **ip msdp originator-id** 中的地址会成为 RP 地址。

用户可以按照以下步骤，配置边界路由器向 MSDP 对等体为发送密集模式区域中的活跃源发送 SA 消息。

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip msdp border sa-address** *interface-id* **示例：** `Device(config)# ip msdp border sa-address 0/1`	配置位于密集模式区域和稀疏模式区域边界的交换机，让它发送密集模式区域中的活跃源 SA 消息。 在 *interface-id* 部分指定接口，这个接口的 IP 地址会在 SA 消息中用作 RP 地址。

		这个接口的 IP 地址会用作起源 ID，这是 SA 消息中的 RP 字段
步骤 4	**ip msdp redistribute** [**list** *access-list-name*] [**asn** *aspath-access-list-number*] [**route-map** *map*] **示例：** `Device(config)# ip msdp redistribute list 100`	配置在 SA 消息中通告组播路由表中的哪些(S, G)条目。 更多信息用户可以参考“重新分布源”
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 RP 地址之外的起源地址

用户可以通过更改起源 ID（Originator ID），让生成 SA 消息的 MSDP 设备在 SA 消息中，使用接口 IP 地址作为 RP 地址。用户可能在以下情况中希望更改起源 ID：

- 如果用户在一个 MSDP 全互联组中的多个设备上配置了一个逻辑 RP；
- 如果用户有一台设备位于 PIM 稀疏模式区域和 PIM 密集模式区域的边界上。如果设备作为密集模式区域的边界，也就是稀疏模式是外部区域的话，用户可能会希望外部网络能够知道密集模式中的源。由于这台设备并不是 RP，因此它没有能够用在 SA 消息中的 RP 地址。因此这条命令通过指定一个接口的 IP 地址，提供了 RP 地址。

如果用户同时配置了全局配置命令 **ip msdp border sa-address** 和 **ip msdp originator-id**，那么命令 **ip msdp originator-id** 中的地址会成为 RP 地址。

用户可以按照以下步骤，来为生成 SA 消息的 MSDP 设备指定一个接口的 IP 地址，作为 SA 消息中的 RP 地址：

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip msdp originator-id** *interface-id* **示例：** `Device(config)# ip msdp originator-id 0/1`	配置起源设备上的接口地址作为 SA 消息中的 RP 地址。 在 *interface-id* 部分指定本地设备上的接口
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

MSDP 的监控和维护

用户可以使用以下命令来监控 MSDP SA 消息、对等体、状态和对等体状态：

表 103：监控和维护 MSDP 的命令

命令	目的
debug ip msdp [*peer-address* | *name*] [**detail**] [**routes**]	显示 MSDP 活动的调试信息
debug ip msdp resets	显示 MSDP 对等体重置原因的调试信息
show ip msdp count [*autonomous-system-number*]	显示每个自治系统 SA 消息中生成的源和组数量。要想让这条命令显示出计数器，用户必须配置命令 **ip msdp cache-sa-state**
show ip msdp peer [*peer-address* | *name*]	显示有关一个 MSDP 对等体的详细信息
show ip msdp sa-cache [*group-address* | *source-address* | *group-name* | *source-name*] [*autonomous-system-number*]	显示从 MSDP 对等体学到的(S, G)状态
show ip msdp summary	显示 MSDP 对等体状态和 SA 消息计数器

用户可以使用以下命令来清除 MSDP 会话、状态统计信息和 SA 缓存条目：

表 104：清除 MSDP 连接、状态统计信息或 SA 缓存条目的命令

命令	目的
clear ip msdp peer *peer-address* | *name*	清除与指定 MSDP 对等体之间的 TCP 连接，重置所有 MSDP 消息计数器
clear ip msdp statistics [*peer-address* | *name*]	清除一个或所有 MSDP 对等体的状态统计计数器，而不重置会话
clear ip msdp sa-cache [*group-address* | *name*]	清除 SA 缓存条目中的所有条目、指定组的所有源，或指定源/组对的所有条目

配置 MSDP 的配置示例

配置默认的 MSDP 对等体：示例

这个示例展示了路由器 A 和路由器 C 上的部分配置。这些 ISP 都有多个客户（比如？中的客户）使用默认的对等体（不使用 BGP 或 MBGP）。在这种情况中，这些客户可能都有类似的配置。也就是说如果相应的前缀列表中放行了 SA 的话，它们都只会接受默认对等体的 SA。

路由器 A

```
Router(config)# ip msdp default-peer 10.1.1.1
Router(config)# ip msdp default-peer 10.1.1.1 prefix-list site-a
Router(config)# ip prefix-list site-b permit 10.0.0.0/1
```

路由器 C

```
Router(config)# ip msdp default-peer 10.1.1.1 prefix-list site-a
Router(config)# ip prefix-list site-b permit 10.0.0.0/1
```

配置活跃源状态：示例

这个示例展示了如何为 171.69.0.0/16 发往组 224.2.0.0/16 中的所有源启用缓存状态：

```
Device(config)# ip msdp cache-sa-state 100
Device(config)# access-list 100 permit ip 171.69.0.0 0.0.255.255
224.2.0.0 0.0.255.255
```

从一个 MSDP 对等体组请求源信息：示例

这个示例展示了如何配置交换机，让它向 MSDP 对等体 171.69.1.1 发送 SA 请求消息：

```
Device(config)# ip msdp sa-request 171.69.1.1
```

控制本地交换机生成的源信息：示例

这个示例展示了如何配置交换机，让它过滤从 MSDP 对等体 171.69.2.2 发来的 SA 请求消息。从网络 192.4.22.0 收到的 SA 请求消息需要与访问列表 1 进行匹配后才会被接受；其他 SA 请求都会被忽略。

```
Device(config)# ip msdp filter sa-request 171.69.2.2 list 1
```

```
Device(config)# access-list 1 permit 192.4.22.0 0.0.0.255
```

控制本地交换机转发的源信息：示例

这个示例展示了如何配置交换机，让它在向名为 *switch.icntnetworks.com* 的对等体转发 SA 消息时，只发送访问列表 100 中匹配的(S, G)对：

```
Device(config)# ip msdp peer switch.icntnetworks.com connect-source gigabitethernet1/0/1
Device(config)# ip msdp sa-filter out switch.icntnetworks.com list 100
Device(config)# access-list 100 permit ip 171.69.0.0 0.0.255.255 224.20 0 0.0.255.255
```

控制本地交换机接收的源信息：示例

这个示例展示了如何过滤从名为 *switch.icntnetworks.com* 的对等体收到的所有 SA 消息：

```
Device(config)# ip msdp peer switch.icntnetworks.com connect-source gigabitethernet1/0/1
Device(config)# ip msdp sa-filter in switch.icntnetworks.com
```

配置 IP 单播路由

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特

性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

IP 单播路由的相关信息

这一部分描述了如何在交换机上配置 IP 版本 4（IPv4）单播路由。

在网络中的其他路由设备看来，交换机堆栈就是一台路由设备。它能够使用基本的路由功能，比如静态路由。要想使用高级路由特性和其他路由协议，用户必须在单台交换机或在活跃交换机上启用 IP Services 特性集。

注释： 除了 IPv4 流量之外，用户也可以启用 IP 版本 6（IPv6）单播路由，并配置接口转发 IPv6 流量。

IP 路由的相关信息

在一些网络环境中，会有多个 VLAN 与每个网络或子网相关联。在一个 IP 网络中，每个子网都映射到一个 VLAN 中。配置 VLAN 有助于控制广播域的范围，把本地流量控制在本地范围内。但是位于不同 VLAN 中的设备如果没有三层设备（路由器）的帮助，无法进行通信，三层设备会对 VLAN 之间的流量进行路由，这称为 VLAN 间路由。用户可以配置一台或多台路由器来为指定目的 VLAN 提供路由功能。

下图展示了一个基本的路由拓扑。交换机 A 在 VLAN 10 中，交换机 B 在 VLAN 20 中。路由器通过两个接口分别连接这两个 VLAN。

图 90：路由拓扑示例

Host（共 3 处）	主机
Switch A	交换机 A
Switch B	交换机 B
Dot1q Trunks	802.1Q Trunk

当 VLAN 10 中的主机 A 需要与 VLAN 10 中的主机 B 进行通信时，它会向主机 B 发送数据包。交换机 A 会直接把这个数据包转发给主机 B，而无需把它转发给路由器。

当主机 A 需要向 VLAN 20 中的主机 C 发送数据包时，交换机 A 会把这个数据包转发给路由器，路由器会从自己连接在 VLAN 10 中的接口收到这个数据包。路由器会检查自己的路由表，找到正确的出站接口，然后通过自己连接 VLAN 20 的接口把数据包转发给交换机 B。交换机 B 在收到数据包后，会把它转发给主机 C。

路由的类型

路由器和三层交换机可以通过以下方式对数据包进行路由：

- 使用默认路由；
- 使用为指定流量设置的静态路由；
- 使用路由协议动态计算路由。

默认路由指的是对于那些去往路由器未知目的地的流量，路由器使用一个默认的出口或目的地进行路由。

静态单播路由功能会从预先确定的端口转发数据包，通过单条路径进入或离开一个网络。静态路由是安全的，并且只占用少量带宽；但它不能对网络中的变化自动做出响应，比如链路故障；因此有可能会出现不可达的目的地。随着网络的增长，维护静态路由变成了劳动密集型工作。

运行 LAN Base 特性集的交换机最多支持 16 条用户配置的静态路由，以及一条用于管理接口的默认路由。LAN Base 镜像只在 SVI 接口上支持静态路由。

动态路由协议是路由器用来为流量动态计算最优转发路由的协议。动态路由协议分为以下两种类型：

- 使用距离矢量协议的路由器会在路由表中维护网络资源的距离值，并且周期性地把这些表传递给它们的邻居。距离矢量协议使用一种或一系列度量值来计算最优路由。这些协议易于配置和使用；
- 使用链路状态协议的路由器会在一个复杂的数据库中维护网络拓扑信息，这些信息来自于路由器之间交换的链路状态通告（LSA）。LSA 是由网络中的事件触发的，这种触发机制提高了网络的收敛时间，或者对于网络变化的响应时间。链路状态协议可以对拓扑变化做出快速响应，但它比距离矢量协议需要更多的带宽和更多的资源。

  交换机上能够支持的距离矢量协议有路由信息协议（RIP），它使用单一距离度量参数（开销）来确定最优路径；以及边界网关协议（BGP），它使用路径矢量机制。交换机还支持链路状态协议开放式最短路径优先（OSPF）和增强型 IGRP（EIGRP），EIGRP 在传统内部网关路由协议（IGRP）的基础上添加了一些链路状态路由特性，增强了工作效率。

注释： 在一台交换机或一个交换机堆栈中，交换机上所支持的协议是由活跃交换机上运行

的软件决定的。如果活跃交换机运行 IP Base 特性集，那它就只支持默认路由、静态路由和 RIP。如果交换机运行 LAN Base 特性集，那么用户可以在 SVI 接口上最多配置 16 条静态路由。所有其他路由协议都需要 IP Services 特性集。

IP 路由和交换机堆栈

在网络中，交换机堆栈是以一台交换机的形式进行运作的，无论堆栈中的哪条交换机连接着路由对等体。

活跃交换机能够执行以下功能：

- 它能够初始化并配置路由协议；
- 它能够向其他路由器发送路由协议消息和更新；
- 它能够处理从其他对等体路由器那里接收到的路由协议消息和更新；
- 总的来说，它能够向所有堆栈成员维护和分布 dCEF（分布式 Inspur 快速转发）数据库。所有交换机上配置的路由都以堆栈为基础储存在这个数据库中；
- 活跃交换机的 MAC 地址作为整个堆栈的路由器 MAC 地址，所有外部设备都使用这个地址向堆栈发送 IP 数据包；
- 所有需要进行软件转发或处理的数据包都会由活跃交换机的 CPU 进行处理。

堆栈成员能够执行以下功能：

- 它们充当路由备份交换机，准备好在活跃交换机失效时，自己选举成为接管活跃交换机角色的交换机；
- 它们能够在硬件中部署路由。

如果活跃交换机失效了，堆栈会检测到活跃交换机的失效事件，并在堆栈成员中选举出一个作为新的活跃交换机。在这段期间网络可能会暂时中断，堆栈硬件会继续转发无需主动协议的数据包。

但是，即使在失效后交换机堆栈会维护硬件识别符，在活跃交换机重启前，路由器邻居上的路由协议还是可能会在短暂的中断期间翻动。比如 OSPF 和 EIGRP 之类的路由协议需要识别邻居的状态过渡。路由器会使用两个级别的无间断转发（NSF）来检测活跃交换机的切换，以便继续转发网络流量，以及从对等体设备那里恢复路由信息：

- 具有 NSF 感知功能的路由器能够忍受邻居路由器的失效。在邻居路由器重启后，具有 NSF 感知功能的路由器会按需提供自己的状态信息和路由邻接关系；
- 具有 NSF 功能的路由能够支持 NSF。当它们检测到活跃交换机改变时，它们会从具有 NSF 功能或具有 NSF 感知功能的邻居那里重建路由信息，并且不会等待活跃交换机重启。

交换机堆栈能够使用具有 NSF 的路由功能为 OSPF 和 EIGRP 提供服务。

在选举后，新的活跃交换机能够执行以下功能：

- 它会开始生成、接收和处理路由更新；
- 它会建立路由表、生成 CEF 数据库，并把这些信息分发给堆栈成员；
- 它会使用自己的 MAC 地址作为路由器 MAC 地址。为了向网络对等体告知这个新的 MAC 地址，它会周期性（5 分钟之内每隔几秒钟）发送无故 ARP 响应消息，来告知这个新的路由器 MAC 地址。

  注释： 如果用户在堆栈上配置了一致 MAC 地址特性，那么当活跃交换机发生改变时，堆栈 MAC 地址在配置的时间段内是不会发生变化的。如果前一个活跃交换机在这段期间内重新加入了堆栈，成为堆栈成员之一，那么堆栈 MAC 地址就会保留前一个活跃交换机的 MAC 地址。
- 它会通过向代理 ARP IP 地址发送 ARP 请求，并接收 ARP 响应，来确定每个代理 ARP 条目的可达性。对于每个可达的代理 ARP IP 地址，它会生成一个无故 ARP 响应消息，来告知新的路由器 MAC 地址。新的活跃交换机选举结束后，这个过程会持续 5 分钟。

  注释： 当活跃交换机上运行 IP Services 特性集时，堆栈可以运行所有支持的协议，其中包括开放式最短路径优先（OSPF）和增强型 IGRP（EIGRP）。如果活跃交换机失效了，新选举为活跃交换机的设备上运行 IP Base 或 LAN Base 特性集，那么这个堆栈中就不再能运行上述协议了。

  注意： 把交换机堆栈分割为两个或多个堆栈，可能会在网络中产生不良的行为。

如果交换机重启了，那么交换机上的所有端口也就都关闭了，用来进行路由的接口上的流量会丢失，哪怕部署了 NSF/SSO 功能。

无类路由

默认情况下，当用户在设备上配置路由功能时，无类路由行为都是启用的。在使用无类路由时，如果路由器收到去往一个网络中一个子网的数据包，并且路由器上没有配置默认路由，那么路由器会按照最优的超网路由来转发这个数据包。一个超网是由多个连续 C 类地址空间的构成的，用来模拟一个单个且更大的地址空间，设计它的初衷是为了缓解快速消耗的 B 类地址空间。

下图中启用了无类路由行为。当主机向 120.20.4.1 发送数据包时，路由器不会丢弃这个数据包，而是按照最优超网路由来转发这个数据包。如果用户禁用了无类路由行为，那么路由器在收到去往一个网络中的一个子网中设备的数据包时，如果路由器上没有网络默认路由，它就会丢弃这个数据包。

图 91：IP 无类路由

IP classless	IP 无类

在下图中，网络 128.20.0.0 中的路由器连接着子网 128.20.1.0、128.20.2.0 和 128.20.3.0。如果主机向 120.20.4.1 发送数据包，由于路由器上没有网络默认路由，因此它会丢弃这个数据包。

图 92：没有 IP 无类路由

Bit bucket	比特桶

为了防止设备使用最优超网路由来转发去往未知子网的数据包，用户可以禁用无类路由行为。

地址解析

用户可以使用地址解析，来控制接口对 IP 的处理行为。使用 IP 协议的设备可以同时拥有本地地址或 MAC 地址，这个地址在设备本地网段或 VLAN 中唯一标识了设备；以及网络地址，这个地址标识了设备所属的网络。

注释： 在交换机堆栈中，网络通信使用一个 MAC 地址和堆栈的 IP 地址。

本地地址或 MAC 地址是数据链路地址，因为它的信息包含在数据包头部中的数据链路层（二层）部分中，并且会由数据链路层（二层）设备读取。要想与一个以太网中的设备进行通信，软件必须学到设备的 MAC 地址。通过 IP 地址学习 MAC 地址的过程称为*地址解析*。通过 MAC 地址学习 IP 地址的过程称为*反向地址解析*。

设备可以使用以下方式来进行地址解析：

- 设备使用地址解析协议（ARP）把 IP 地址与 MAC 地址进行关联。通过入站数据包的 IP 地址，ARP 能够学到与之相关联的 MAC 地址，并把它们保存到关联了 IP 地址/MAC 地址的 ARP 缓存中，以便快速检索。之后这个 IP 数据报会被封装在链路层帧中，并发送到网络中。除以太网之外，IEEE 802 网络上的 IP 数据报和 ARP 请求/响应封装方式由子网访问协议（SNAP）来定义。
- 代理 ARP 帮助没有路由表的主机学习其他网络或子网上的主机 MAC 地址。如果设备（路由器）接收到一个主机的 ARP 请求，且请求的地址与这个 ARP 请求的发送方不连接在同一个路由器接口上，如果路由器上有通过其他借口去往这个主机的路由，那么它就会生成一个代理 ARP 包，并在其中提供自己的本地数据链路地址。发送 ARP 请求的主机接着会把自己的数据包发送给路由器，之后路由器负责把数据包转发给正确的主机。

设备还会使用反向地址解析协议（RARP），它的功能与 ARP 相同，只是 RARP 包请求的是 IP 地址，而不是本地 MAC 地址。要想使用 RARP，需要在路由器接口连接的网段中有一台 RARP 服务器。用户可以使用接口配置模式的命令 **ip rarp-server** *address*，来指明这个服务器。

更多有关 RARP 的信息，用户可以参考 *Inspur INOS Configuration Fundamentals Configuration*

Guide。

代理 ARP

代理 ARP 是学习其他路由相关信息最常用的方法，使没有路由信息的以太网主机能够域其他网络或子网中的主机进行通信。主机会假设所有主机都位于相同的本地以太网中，并且它们能够使用 ARP 学习这些主机的 MAC 地址。如果设备从主机接收到一个 ARP 请求，所请求的主机与发送方主机不在同一个网络中，设备会评估自己是否有去往这个主机的最优路由。如果有的话，它会在 ARP 响应包中发送自己的以太网 MAC 地址，发送请求的主机之后就会把数据包发送给设备，设备会继而把数据包转发给正确的主机。代理 ARP 把所有网络都当作本地网络，并为所有 IP 地址执行 ARP 请求。

ICMP 路由器发现协议

路由器发现协议使设备能够使用 ICMP 路由器发现协议（IRDP），动态学习去往其他网络的路由相关信息。IRDP 使主机能够定位路由器。当运行为客户端时，设备会生成路由器发现数据包。当运行为主机时，设备会接收路由器发现数据包。

设备还可以侦听路由信息协议（RIP）路由更新，并使用这些信息来判断路由器的位置。设备并不会真正储存路由设备发送的路由表；它仅仅追踪发送这些数据的系统。使用 IRDP 的好处是每台路由器上都可以指定一个优先级和时间，在这段时间内如果没有接收到更多数据包，就认为这台设备已经离线。

发现的每台设备都会成为默认路由器的候选者，当发现了一台拥有更高优先级的路由器时，或者当前的默认路由器被判断为离线后，或者由于重传过多而使 TCP 连接超时后，就会选举新的最高优先级路由器。

UDP 广播数据包和协议

用户数据报协议（UDP）与 TCP 一样是一项 IP 主机到主机层协议。UDP 在两个终端系统之间提供了低开销无连接的会话，它不会对接收到的数据报提供确认。网络主机很少会使用 UDP 广播来查找地址、配置和域名信息。如果网段中有一台这样的主机，但没有服务器的话，UDP 广播通常不会被转发。用户可以配置路由器接口，让它向协助地址转发特定的广播类别，来解决这一问题。用户可以在一个接口上使用多个协助地址。

用户可以通过指定 UDP 目的端口，来控制设备转发哪些 UDP 服务。用户可以指定多个 UDP

协议。用户也可以指定网络硬盘（ND）协议，老式无盘 Sun 工作站和网络安全协议 SDNS 会使用 ND 协议。

默认情况下，如果接口上定义了协助地址的话，UDP 和 ND 转发就是启用的。*Inspur INOS IP Command Reference, Volume 1 of 3: Addressing and Services* 中对接口配置命令 **ip forward-protocol** 的描述中，列出了用户没有指定 UDP 端口时，默认转发的端口。

广播数据包的处理

在配置了 IP 接口地址后，用户可以启用路由功能并配置一个或多个路由协议，或者也可以配置设备用来响应网络广播的方法。广播是去往一个物理网络上所有主机的数据包。设备支持以下两类广播：

- 定向广播数据包是发往一个指定网络或一系列网络的。定向广播地址中包含网络或子网字段；
- 泛洪广播数据包是发往每个网络的。

  注释： 用户也可以使用接口配置命令 **storm-control**，把二层接口上的广播、单播和组播流量限制在一个抑制级别上。

路由器提供了几种预防广播风暴的方法，限制广播扩展到本地线缆上。由于网桥（包括智能网桥）是二层设备，因此它们会把广播转发到所有网络中，这样会导致广播风暴。解决广播风暴的最好方法是在网络中使用单一的广播地址机制。在大多数现代 IP 部署环境中，用户可以设置用作广播地址的地址。很多部署方式（包括在一台设备上进行部署）都支持多种编址机制来转发广播消息。

IP 广播泛洪

用户可以允许广播在互联网络中进行泛洪，但要使用桥接 STP 创建的数据库以一种可控的方式进行泛洪。使用这个特性可以防止环路。要想支持这个特性，每个参与泛洪的接口上必须配置桥接功能。如果用户没有在接口上配置桥接功能的话，接口仍可以接收广播。但它永远不会转发自己接收到的广播，并且路由器永远不会使用这些接口来发送其他接口上接收到的广播。

使用 IP 协助地址机制向单个网络地址转发的数据包可以被泛洪。设备只会在每个网段上发送这个数据包的一个副本。

要想让数据包被泛洪，数据包必须满足以下条件（这些条件与使用 IP 协助地址转发数据包的条件相同）：

- 数据包必须是 MAC 级别的广播；
- 数据包必须是 IP 级别的广播；
- 数据包必须是 TFTP 包、DNS 包、Time 包、NetBIOS 包、ND 或 BOOTP 包，或者全局配置命令 **ip forward-protocol udp** 指定的 UDP 包；
- 数据包的生存时间（TTL）值必须至少为 2。

能够被泛洪的 UDP 数据报目的地址必须通过接口配置模式的命令 **ip broadcast-address**，指定在出站接口上。目的地址可以设置为任意地址。因此数据报在网络中传播时，目的地址可能会改变。源地址永远不变，TTL 值会递减。

当被泛洪的 UDP 数据报从一个接口发出后（目的地址可能会发生变化），数据报的处理方式域普通 IP 输出流量相同，因此也会遇到访问列表，如果出站接口上配置了访问列表的话。

在设备中，绝大多数数据包都是在硬件中转发的；多数数据包并不会穿越设备的 CPU。对于这些不去 CPU 的数据包来说，用户可以使用 Turbo-Flooding 加速基于生成树的 UDP 泛洪，速度能够提升 4、5 倍。配置了 ARP 封装的以太网接口上能够支持这个特性。

如何配置 IP 路由

默认情况下设备上是禁用 IP 路由功能的，用户必须启用 IP 路由功能，设备才能对数据包进行路由。更多有关 IP 路由配置的信息，用户可以参考 *Inspur INOS IP Configuration Guide*。

在下面这个流程中，指定接口必须是以下三层接口之一：

- 路由端口：物理端口，用户使用接口配置模式的命令 **no switchport** 将其配置为三层端口；
- 交换机虚拟接口（SVI）：VLAN 接口，用户使用全局配置模式的命令 **interface vlan** *vlan_id* 创建的接口，默认就是三层接口；
- 三层模式的 EtherChannel 端口隧道：PortChannel 逻辑接口，用户使用全局配置模式的命令 **interface port-channel** *port-channel-number* 创建的，把以太网接口绑定在隧道组中。更多相关信息，用户可以参考二层配置指南中的“配置三层 EtherChannel”一章。

  注释： 交换机不支持为单播路由流量使用隧道接口。

所有需要提供路由功能的三层接口上都必须配置 IP 地址。

注释： 三层交换机可以为每个路由端口和 SVI 接口分配 IP 地址。

用户可以配置的路由端口和 SVI 接口数量限制为 128，超出建议数值或使用过多特性可能会由于硬件的限制，而影响 CPU 利用率。

路由的配置由以下主要流程构成：

- 为了支持 VLAN 接口，用户需要在设备或交换机堆栈上创建并配置 VLAN，并把 VLAN 成员分配到二层接口。更多相关信息，用户可以参考 VLAN 配置指南中的“配置 VLAN”一章；
- 配置三层接口；
- 在交换机上启用 IP 路由功能；
- 为三层接口分配 IP 地址；
- 在交换机上启用指定路由协议；
- 配置路由协议参数（可选）。

如何配置 IP 编址

配置 IP 路由工作中的一个必需任务是为三层接口分配 IP 地址，以此启用接口并允许这个接口上的主机使用这个 IP 地址与之进行通信。接下来的这一部分描述了如何配置不同的 IP 编址特性。用户必须为接口分配 IP 地址；其他步骤是可选的。

- 默认的编址配置
- 向网络接口分配 IP 地址
- 配置地址解决方案的方法
- IP 路由功能禁用时的路由协助
- 配置广播数据包的处理
- 监控和维护 IP 编址

默认的 IP 编址配置

表 105：默认的编址配置

特性	**默认设置**
IP 地址	无定义
ARP	地址解析协议（ARP）缓存中没有永久条目。 封装：标准以太网类型 ARP 超时：14400 秒（4 小时）
IP 广播地址	255.255.255.255（全 1）
IP 无类路由	启用

IP 默认网关	禁用
IP 定向广播	禁用（所有 IP 定向广播都会被丢弃）
IP 域	域列表：没有定义域名 域查找：启用 域名：启用
IP 转发协议	如果定义了协助地址，或配置了用户数据报协议（UDP）泛洪，UDP 转发在默认端口上是启用的。 任意本地广播：禁用 生成树协议（STP）：禁用 Turbo-Flood：禁用
IP 协助地址	禁用
IP 主机	禁用
IRDP	禁用 启用后的默认设置： • 广播 IRDP 通告 • 通告之间的最大间隔：600 秒 • 通告之间的最小间隔：0.75 倍的最大间隔 • 优先级：0
IP 代理 ARP	启用
IP 路由	禁用
IP 全零子网	禁用

为网络接口分配 IP 地址

IP 地址标识了这个 IP 数据包能够被发送到哪里。有些 IP 地址为特殊用途保留，不能用在主机、子网或网络地址。RFC 1166“Internet Numbers”中包含了官方对 IP 地址的描述。

一个接口上可以有一个主用 IP 地址。掩码标识了 IP 地址中的网络号。当用户使用掩码对网络进行子网划分时，这个掩码被称为子网掩码。要想获得网络号，用户需要联系自己的 Internet 服务提供商。

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密

	示例： `Device> enable`	码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/1`	进入接口配置模式，并指定用户想要配置的三层接口
步骤 4	**no switchport** **示例：** `Device(config-if)# no switchport`	从二层配置模式移除接口（如果这是物理接口的话）
步骤 5	**ip address** *ip-address subnet-mask* **示例：** `Device(config-if)# ip address 10.1.5.1 255.255.255.0`	配置 IP 地址和 IP 子网掩码
步骤 6	**no shutdown** **示例：** `Device(config-if)# no shutdown`	启用物理接口
步骤 7	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式

步骤 8	**show ip route** **示例：** `Device# show ip route`	检查用户输入的信息
步骤 9	**show ip interface** [*interface-id*] **示例：** `Device# show ip interface gigabitethernet 1/0/1`	检查用户输入的信息
步骤 10	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 11	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

使用全零子网

强烈不建议用户在子网划分是使用全零作为子网地址，因为如果一个网络和一个子网有相同的地址时会产生问题。举例来说，如果网络 131.108.0.0 被子网划分为 255.255.255.0，子网 0 就是 131.108.0.0，它与网络地址是相同的。

用户可以使用全 1 的子网（131.108.255.0），虽然并不推荐这样做，但如果用户需要为 IP 地址部署全部子网空间的话，也可以在需要时启用全零子网。

用户可以使用全局配置模式的命令 **no ip subnet-zero** 来恢复默认设置，并禁用全零子网。

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码

步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip subnet-zero** **示例：** `Device(config)# ip subnet-zero`	为接口地址和路由更新启用全零子网
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

禁用无类路由

要想阻止设备使用最优超网路由来转发去往未知子网的数据包，用户可以禁用无类路由行为。

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码

步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**no ip classless** **示例：** `Device(config)#no ip classless`	禁用无类路由行为
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置地址解析协议

用户可以按照以下步骤来配置地址解析。

定义一个静态 ARP 缓存

ARP 和其他地址解析协议能够提供 IP 地址和 MAC 地址之间的动态映射。由于大多数主机都支持动态地址解析，因此用于通常不需要指定静态 ARP 缓存条目。如果必须定义一个静态 ARP 缓存条目的话，用户可以在全局进行配置，这样做会在 ARP 缓存中指定一个永久的条目，设备会用它把 IP 地址转换为 MAC 地址。可选的，用户也可以让设备在响应 ARP 请求时，

假装它就是拥有这个 IP 地址的设备。如果用户不希望 ARP 条目是永久的，也可以为这个 ARP 条目指定超时时间。

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**arp** *ip-address hardware-address type* **示例：** `Device(config)# ip 10.1.5.1 c2f3.220a.12f4 arpa`	在 ARP 缓存中把 IP 地址与 MAC 地址（硬件）关联在一起，并指定以下封装类型之一： • **arpa**——用于以太网接口的 ARP 封装 • **snap**——用于令牌环和 FDDI 接口的子网地址协议封装 • **sap**——HP 的 ARP 类型
步骤 4	**arp** *ip-address hardware-address type* [**alias**] **示例：** `Device(config)# ip 10.1.5.3 d7f3.220d.12f5 arpa alias`	（可选）让交换机在响应 ARP 请求时，假装自己就是拥有这个 IP 地址的设备
步骤 5	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/1`	进入接口配置模式，并指定用户想要配置的三层接口
步骤 6	**arp** *timeout seconds*	（可选）设置 ARP 缓存条目能够保存在缓存中的时间长度。默认值是 14400 秒

	示例： `Device(config-if)# arp 20000`	（4 小时），取值范围是 0 至 2147483 秒
步骤 7	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 8	**show ip interface** [*interface-id*] **示例：** `Device# show ip interface gigabitethernet 1/0/1`	检查所有接口或指定接口上的 ARP 类型和使用的超时时间
步骤 9	**show arp** **示例：** `Device# show arp`	查看 ARP 缓存的内容
步骤 10	**show ip arp** **示例：** `Device# show ip arp`	查看 ARP 缓存中的内容
步骤 11	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

设置 ARP 封装

默认情况下，IP 接口上启用的是以太网 ARP 封装（由关键字 **arpa** 表示）。用户可以按照网络需求，把封装方式更改为 SNAP。

要想禁用一种封装类型，用户需要使用接口配置模式的命令 **no arp arpa** 或 **no arp snap**。

具体步骤

	命令或操作	**目的**

步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/2`	进入接口配置模式，并指定用户想要配置的三层接口
步骤 4	**arp** {**arpa** | **snap**} **示例：** `Device(config-if)# arp arpa`	指定 ARP 封装方式： • **arpa**——地址解析协议 • **snap**——子网地址协议
步骤 5	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 6	**show ip interface** [*interface-id*] **示例：** `Device# show ip interface gigabitethernet 1/0/2`	检查所有接口或指定接口上的 ARP 封装配置
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

启用代理 ARP

默认情况下，设备使用代理 ARP 来帮助主机学习其他网络或子网上的主机 MAC 地址。

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/2`	进入接口配置模式，并指定用户想要配置的三层接口
步骤 4	**ip proxy-arp** **示例：** `Device(config-if)# ip proxy-arp`	在接口上启用代理 ARP
步骤 5	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 6	**show ip interface** [*interface-id*] **示例：** `Device# show ip interface gigabitethernet 1/0/2`	检查所有接口或指定接口上的 ARP 封装配置

步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

禁用 IP 路由时的路由协助

设备可以使用以下机制，在没有启用 IP 路由功能时学习去往其他网络的路由：

- 代理 ARP
- 默认网关
- ICMP 路由器发现协议（IRDP）

代理 ARP

代理 ARP 默认就是启用的。要想在禁用后再次启用，用户可以查看“启用代理 ARP”一节。只要其他路由器也支持代理 ARP 的话，代理 ARP 就可以正常工作。

默认网关

另一种定位路由的方法是定义默认路由器或默认网关。所有非本地的数据包都会发送给用户指定的路由器，然后这台路由器会按需对这个数据包进行路由，或者返回 IP 控制消息协议（ICMP）重定向消息，指定这个主机应该使用的本地路由器。设备会缓存重定向消息，并高效地转发每个数据包。使用这种方法的限制是设备无法检测默认路由器是否已经下线或者是否可用。

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device# configure terminal`	
步骤 3	**ip default-gateway** *ip-address* **示例：** `Device(config)# ip default gateway 10.1.5.1`	设置默认网关（路由器）
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show ip redirects** **示例：** `Device# show ip redirects`	显示默认网关路由器的地址来确认用户的设置
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

IMCP 路由器发现协议（IRDP）

在接口上启用 IRDP 路由的唯一必需工作是在接口上启用 IRDP 进程。当启用后，设备就会应用默认参数。

用户（可选的）可以更改这些参数。如果用户改变了 **maxadvertinterval** 值，**holdtime** 和 **minadvertinterval** 值也会改变，因此重要的是先改变 **maxadvertinterval** 值，然后在手动改变 **holdtime** 或 **minadvertinterval** 值。

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密码

	示例： `Device> enable`	
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/1`	进入接口配置模式，并指定用户想要配置的三层接口
步骤 4	**ip irdp** **示例：** `Device(config-if)# ip irdp`	在接口上启用 IRDP 进程
步骤 5	**ip irdp multicast** **示例：** `Device(config-if)# ip irdp multicast`	（可选）代替 IP 广播，向组播地址（224.0.0.1）发送 IRDP 通告。 **注释：** 这条命令用于兼容 Sun Microsystems Solaris，它需要 IRDP 包以组播的形式发出。很多部署中不接收这些组播；用户要在使用这条命令前确认终端主机的能力
步骤 6	**ip irdp holdtime** *seconds* **示例：** `Device(config-if)# ip irdp holdtime 1000`	（可选）设置有效的 IRDP 通告周期。默认为 **maxadvertinterval** 值的 3 倍。这个值必须大于 **maxadvertinterval**，但不能超过 900 秒。如果用户更改 **maxadvertinterval** 值，这个值也会发生变化
步骤 7	**ip irdp maxadvertinterval** *seconds* **示例：** `Device(config-if)# ip irdp`	（可选）设置通告之间的 IRDP 最大间隔。默认值为 600 秒

	maxadvertinterval 650	
步骤 8	**ip irdp minadvertinterval** *seconds* **示例：** Device(config-if)# ip irdp minadvertinterval 500	（可选）设置通告之间的最小间隔。默认值为 **maxadvertinterval** 值的 0.75 倍。如果用户更改 **maxadvertinterval** 值，这个值也会发生变化（**maxadvertinterval** 值的 0.75 倍）
步骤 9	**ip irdp preference** *number* **示例：** Device(config-if)# ip irdp preference 2	（可选）设置设备的 IRDP 优先级。可配置的取值范围是-231 至 231。默认值为 0。较高的值会增加路由器的优先级级别
步骤 10	**ip irdp address** *address* [*number*] **示例：** Device(config-if)# ip irdp address 10.1.10.10	（可选）为代理通告指定 IRDP 地址和优先级
步骤 11	**end** **示例：** Device(config-if)# **end**	返回特权 EXEC 模式
步骤 12	**show ip irdp** **示例：** Device# show ip irdp	通过显示 IRDP 值来确认用户的设置
步骤 13	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

配置广播数据包的处理

用户可以按照以下步骤启用这些机制：

- 启用定向广播到物理广播的转换
- 转发 UDP 广播包和协议
- 建立 IP 广播地址
- 泛洪 IP 广播

启用定向广播到物理广播的转换

默认情况下，IP 定向广播会被丢弃；设备不会转发这类数据包。丢弃 IP 定向广播可以让路由器不容易遭受拒绝服务攻击。

用户可以在接口上启用 IP 定向广播转发功能，使广播变为物理（MAC 层）广播。只有用户使用全局配置命令 **ip forward-protocol** 配置的协议才会被转发。

用户可以指定一个访问列表来控制转发哪些广播。当用户定义了一个访问列表后，只有访问列表中允许的 IP 地址可以从定向广播转换为物理广播。有关访问列表的更多信息，用户可以参考安全配置指南中的“使用 ACL 提供网络安全”一节。

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/2`	进入接口配置模式，并指定用户想要配置的三层接口
步骤 4	**ip directed-broadcast** [*access-list-number*]	在接口上启用定向广播到物理广播的转换。用户可以使用访问列表来控制转发哪些广播。当使用访问列表时，只有

	示例： Device(config-if)# ip directed-broadcast 103	访问列表汇总允许的 IP 数据包才会被转换
步骤 5	**exit** **示例：** Device(config-if)# exit	返回全局配置模式
步骤 6	**ip forward-protocol** {**udp** [*port*] | **nd** | **sdns**} **示例：** Device(config)# ip forward-protocol nd	指定在转发广播数据包时，路由器会转发的协议和端口。 • **udp**——转发 UDP 数据报 *port*——（可选）通过目的端口号控制转发的 UDP 服务 • **nd**——转发 ND 数据报 • **sdns**——转发 SDNS 数据报
步骤 7	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 8	**show ip interface** [*interface-id*] **示例：** Device# show ip interface gigabitethernet 1/0/2	检查所有接口或指定接口上的配置
步骤 9	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中
步骤 10	**copy running-config startup-config** **示例：** Device# **copy running-config**	（可选）把输入的命令保存到配置文件中

	startup-config	

转发 UDP 广播包和协议

如果用户在配置 UDP 广播转发时没有指定 UDP 端口的话，可以把路由器配置为 BOOTP 转发代理。BOOTP 数据包会承载 DHCP 信息。

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/1`	进入接口配置模式，并指定用户想要配置的三层接口
步骤 4	**ip helper-address** *address* **示例：** `Device(config-if)# ip helper address 10.1.10.1`	启用转发功能并指定转发 UDP 广播报的目的地址，其中包括 BOOTP
步骤 5	**exit** **示例：** `Device(config-if)# exit`	返回全局配置模式
步骤 6	**ip forward-protocol** {**udp** [*port*] | **nd** | **sdns**}	指定在转发广播包时，路由器转发的协议

	示例： `Device(config)# ip forward-protocol sdns`	
步骤 7	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 8	**show ip interface** [*interface-id*] **示例：** `Device# show ip interface gigabitethernet 1/0/1`	检查所有接口或指定接口上的配置
步骤 9	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中
步骤 10	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

建立 IP 广播地址

最常用的 IP 广播地址（和默认设置）是由全 1 构成的地址（255.255.255.255）。但用户可以配置设备生成任意形式的 IP 组播数据包。

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密码

	示例： `Device> enable`	
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/1`	进入接口配置模式，并指定用户想要配置的三层接口
步骤 4	**ip broadcast-address** *ip-address* **示例：** `Device(config-if)# ip broadcast-address 128.1.255.255`	输入与默认值不同的广播地址，比如 128.1.255.255
步骤 5	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 6	**show ip interface** [*interface-id*] **示例：** `Device# show ip interface gigabitethernet 1/0/1`	检查所有接口或指定接口上的配置
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

泛洪 IP 广播

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip forward-protocol spanning-tree** **示例：** `Device(config)# ip forward-protocol spanning-tree`	使用桥接生成树数据库来泛洪 UDP 数据报
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中
步骤 7	**configure terminal**	进入全局配置模式

	示例： `Device# configure terminal`	
步骤 8	**ip forward-protocol turbo-flood** **示例：** `Device(config)# ip forward-protocol turbo-flood`	使用生成树数据库加速 UDP 数据报的泛洪
步骤 9	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 10	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 11	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

监控和维护 IP 编址

当某个缓存、表或数据库中的内容变得不可用或有可能不可用时，用户可以使用特权 EXEC 命令 **clear** 来移除所有内容。下面这个表格中列出了用来清除内容的命令。

表 106：清除缓存、表和数据库的命令

命令	目的
clear arp-cache	清除 IP ARP 缓存和快速交换缓存
clear host {*name* | ***}**	从主机和地址缓存中移除一个或所有条目
clear ip route {**network** [*mask*] | ***}**	从 IP 路由表中移除一个或更多路由

用户可以查看指定的状态统计信息，比如 IP 路由表、缓存和数据库中的内容、节点的可达性，以及数据包穿越网络所使用的路由路径。下面这个表格中列出了显示 IP 状态统计信息的特权 EXEC 命令。

表 107：显示 IP 状态统计信息的特权 EXEC 命令

命令	目标
show arp	显示 ARP 表中的条目
show hosts	显示默认域名、查找服务的方式、域名服务器主机，以及缓存的主机名和地址列表
show ip aliases	显示 IP 地址与 TCP 端口的映射（别名）
show ip arp	显示 IP ARP 缓存
show ip interface [*interface-id*]	显示接口的 IP 状态
show ip irdp	显示 IRDP 值
show ip masks *address*	显示网络地址使用的掩码，以及使用每个掩码的子网数量
show ip redirects	显示默认网关的地址
show ip route [*address* [*mask*]] | [*protocol*]	显示当前的路由表状态
show ip route summary	以汇总的形式显示当前路由表的状态

如何配置 IP 单播路由

启用 IP 单播路由

默认情况下，设备处于二层交换模式，IP 路由功能是禁用的。要想使用设备的三层功能，用户必须启用 IP 路由功能。

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device# configure terminal`	
步骤 3	**ip routing** **示例：** `Device(config)# ip routing`	启用 IP 路由功能
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

启用 IP 路由的示例

这个示例展示了如何启用 IP 路由功能，并使用 RIP 作为路由协议：

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# ip routing
Device(config)# router rip
Device(config-router)# network 10.0.0.0
Device(config-router)# end
```

接下来做什么？

用户现在可以为使用的路由协议设置参数了，可选的路由协议包括以下这些：

- RIP

- OSPF
- EIGRP
- BGP
- 单播反向路径转发
- 协议无关特性（可选）

RIP 的相关信息

路由信息协议（RIP）是一项内部网关协议（IGP），适用于小型网络。它是一向距离矢量路由协议，使用携带广播用户数据报协议（UDP）数据的数据包来交换路由信息。这个协议记录在 RFC 1058 文档中。有关 RIP 的更多信息，用户可以参考 Cisco Press 出版的 *IP Routing Fundamentals*。

注释： IP Base 镜像支持 RIP。

在使用 RIP 时，设备会每隔 30 秒发送路由信息更新（通告）。如果路由器 180 秒或更长时间没有从另一台路由器那里接收到路由更新，它就会把由那台路由器进行转发的路由标记为不可用。如果在此之后的 240 秒内仍没有收到更新，路由器会从路由表中移除所有未更新的路由。

RIP 使用跳数来评估不同的路由。跳数是一条路由中经历的路由器数量。直连网络的跳数为 0；跳数为 16 的网络被认为是不可达的。这个小范围（0 至 15）使 RIP 无法适用于大型网络。如果路由器上有默认网络路径，RIP 会通告一条把路由器连接到伪网络 1.1.1.1 的路由。网络 0.0.0.0 不存在；RIP 使用这个网络来实施默认路由特性。如果从 RIP 学到了默认路由，或者如果路由器有网关，并且 RIP 配置了默认度量值，设备就会通告默认网络。RIP 会向指定网络中的接口发送更新。如果没有指定接口的网络，设备就不会通告任何 RIP 更新。

汇总地址和水平分割

连接到广播类型 IP 网络的路由器在使用距离矢量路由协议时，通常会使用水平分割机制来减少可能出现的环路。水平分割会禁止路由器把一条路由信息从它收到这条路由信息的接口再次通告出去。这个特性通常优化了多台路由器之间的通信，尤其是当链路出现故障时。

如何配置 RIP

默认的 RIP 配置

表 108：默认的 RIP 配置

特性	默认设置
自动汇总	启用
生成默认信息	禁用
IP RIP 认证 密钥链	无认证 认证模式：明文
IP RIP 触发机制	禁用
IP 水平分割	取决于媒介类型
邻居	未定义
网络	未指定
偏移列表	禁用
输出延迟	0 毫秒
基本计时器	• 更新（Update）：30 秒 • 失效（Invalid）：180 秒 • 保留（Hold-Down）：180 秒 • 冲刷（Flush）：240 秒
有效更新源	启用
版本	接收 RIP 版本 1 和 2 数据包；发送版本 1 数据包

配置基本的 RIP 参数

要想配置 RIP，用户需要为网络启用 RIP 路由功能，并有选择地配置其他参数。在设备上，用户需要配置网络号才能使设备接受 RIP 配置命令。

具体步骤

	命令或操作	目的
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密码

	示例： `Device> enable`	
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip routing** **示例：** `Device(config)# ip routing`	启用 IP 路由功能（只适用于 IP 路由已禁用的设备）
步骤 4	**router rip** **示例：** `Device(config)# router rip`	启用 RIP 路由进程，并进入路由器配置模式
步骤 5	**network** *network number* **示例：** `Device(config)# network 12`	把一个网络关联到 RIP 路由进程。用户可以指定多个 **network** 命令。RIP 只会在连接这些网络的接口上发送和接收路由更新。 **注释：** 用户必须配置一个网络号，才能使 RIP 命令生效
步骤 6	**neighbor** *ip-address* **示例：** `Device(config)# neighbor 10.2.5.1`	（可选）定义一个邻居路由器，本地设备要与其交换路由信息。这一步可以使路由器向非广播网络发送 RIP 更新（正常情况下是广播协议）
步骤 7	**offset-list** [*access-list number* | *name*] {**in** | **out**} *offset* [*type number*] **示例：** `Device(config)# offset-list 103 in 10`	（可选）应用一个偏移列表，增加从 RIP 学到的路由的入向和出向度量值。用户可以使用访问列表或接口来限制偏移列表
步骤 8	**timers basic** *update invalid holddown*	（可选）调整路由协议计时器。所有计

	flush **示例：** `Device(config)# timers basic 45 360 400 300`	时器的有效取值范围都是 0 至 4294967295 秒。 • *update*——每次发送路由更新之间的时间。默认值为 30 秒 • *invalid*——路由被认为失效的时间。默认值为 180 秒 • *holddown*——路由被动路由表中移除的时间，默认值为 180 秒 • *flush*——暂时保留路由更新的时间。默认值为 240 秒
步骤 9	**version** {**1** | **2**} **示例：** `Device(config)# version 2`	（可选）配置交换机只接收和发送 RIP 版本 1 或 RIP 版本 2 数据包。默认情况下，交换机会同时接收版本 1 和 2 数据包，但只发送版本 1 数据包。用户也可以使用接口模式的配置命令 **ip rip** {**send** | **receive**} **version 1** | **2** | **1 2**}，控制这个接口在发送和接收 RIP 数据包时使用的版本
步骤 10	**no auto summary** **示例：** `Device(config)# no auto summary`	（可选）禁用自动汇总功能。默认情况下，交换机会在跨越有类网络边界时对子前缀进行汇总。禁用汇总（只适用于 RIP 版本 2 环境）能够让设备在有类网络边界上通告子网和主机路由信息
步骤 11	**no validate-update-source** **示例：** `Device(config)# no validdate-update-source`	（可选）禁止对入站 RIP 路由更新的源 IP 地址进行有效性校验。默认情况下交换机会验证入站 RIP 路由更新源 IP 地址的有效性，并在源地址不合法时丢弃更新消息。在普通环境中，不建议禁用这个特性。但如果用户有一台路由器是离网的，并且用户还希望从它那里接收更新，就需要使用这条命令
步骤 12	**output-delay** *delay*	（可选）在发送 RIP 更新时，增加数据

	示例： `Device(config)# output-delay 8`	包之间的延迟。默认情况下，多个 RIP 更新数据包之间是没有延迟的，如果用户需要向一台速率较低的设备发送这些数据包，就可以在数据包之间增加 8 至 50 毫秒的延迟
步骤 13	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 14	**show ip protocols** **示例：** `Device# show ip protocols`	检查用户输入的信息
步骤 15	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 RIP 认证

RIP 版本 1 不支持认证功能。如果用户的设备是在发送和接收 RIP 版本 2 数据包，就可以在接口上启用 RIP 认证功能。密钥链（Key Chain）指定了接口可以使用的一系列密钥。如果用户没有配置密钥链，认证功能就无法执行，没有默认值。

设备可以在接口上支持两种 RIP 认证模式：明文和 MD5。默认为明文。

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： Device# **configure terminal**	
步骤 3	**interface** *interface-id* 示例： Device(config)# **interface gigabitethernet 1/0/1**	进入接口配置模式，并指定用户想要配置的接口
步骤 4	**ip rip authentication key-chain** *name-of-chain* 示例： Device(config-if)# **ip rip authentication key-chain trees**	启用 RIP 认证功能
步骤 5	**ip rip authentication mode {text | md5}** 示例： Device(config-if)# **ip rip authentication mode md5**	配置接口使用明文认证（默认）或 MD5 摘要认证
步骤 6	**end** 示例： Device(config-if)# **end**	返回特权 EXEC 模式
步骤 7	**show running-config** 示例： Device# **show running-config**	检查用户输入的信息
步骤 8	**copy running-config startup-config** 示例： Device# **copy running-config**	（可选）把输入的命令保存到配置文件中

	startup-config	

配置汇总地址和水平分割

注释： 总的来说，不建议用户禁用水平分割特性，除非用户确定网络中的应用需要禁用水平分割才能正确通告路由。

如果用户希望让运行 RIP 的接口通告汇总的本地 IP 地址池，为网络访问服务器的拨号用户服务，可以使用接口配置命令 **ip summary-address rip**。

注释： 如果水平分割特性是启用的，自动汇总地址和接口 IP 汇总地址也都不会通告出去。

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/1`	进入接口配置模式，并指定用户想要配置的接口
步骤 4	**ip address** *ip-address subnet-mask* **示例：** `Device(config-if)# ip address 10.1.1.10 255.255.255.0`	配置 IP 地址和 IP 子网
步骤 5	**ip summary-address rip ip address** *ip-network mask*	配置要被汇总的 IP 地址和 IP 网络掩码

	示例： `Device(config-if)# ip summary-address rip ip address 10.1.1.30 255.255.255.0`	
步骤 6	**no ip split horizon** **示例：** `Device(config-if)# no ip split horizon`	在接口上禁用水平分割
步骤 7	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 8	**show ip interface** *interface-id* **示例：** `Device# show ip interface gigabitethernet 1/0/1`	检查用户输入的信息
步骤 9	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置水平分割

连接到广播类型 IP 网络的路由器在使用距离矢量路由协议时，通常会使用水平分割机制来减少可能出现的环路。水平分割会禁止路由器把一条路由信息从它收到这条路由信息的接口再次通告出去。这个特性通常优化了多台路由器之间的通信，尤其是当链路出现故障时。

注释： 总的来说，不建议用户禁用水平分割特性，除非用户确定网络中的应用需要禁用水平分割才能正确通告路由。

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/1`	进入接口配置模式，并指定用户想要配置的接口
步骤 4	**ip address** *ip-address subnet-mask* **示例：** `Device(config-if)# ip address 10.1.1.10 255.255.255.0`	配置 IP 地址和 IP 子网
步骤 5	**no ip split horizon** **示例：** `Device(config-if)# no ip split horizon`	在接口上禁用水平分割
步骤 6	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 7	**show ip interface** *interface-id*	检查用户输入的信息

	示例： Device# show ip interface gigabitethernet 1/0/1	
步骤 8	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

汇总地址和水平分割的配置示例

在这个示例中，主网络是 10.0.0.0。汇总地址 10.2.0.0 覆盖了汇总地址 10.0.0.0，因此设备从 GigabitEhternet 2 端口把 10.2.0.0 通告出去，并且不通告 10.0.0.0。在这个示例中，如果接口仍工作在二层模式（默认），用户必须在接口配置模式中使用命令 **no switchport**，之后才能输入接口配置模式的命令 **ip address**。

注释： 如果水平分割是启用的，自动汇总地址和接口汇总地址（使用路由器配置模式的命令 **ip summary-address rip** 进行配置的地址）都不会被通告出去。

```
Device(config)# router rip
Device(config-router)# interface gigabitethernet1/0/2
Device(config-if)# ip address 10.1.5.1 255.255.255.0
Device(config-if)# ip summary-address rip 10.2.0.0 255.255.0.0
Device(config-if)# no ip split-horizon
Device(config-if)# exit
Device(config)# router rip
Device(config-router)# network 10.0.0.0
Device(config-router)# neighbor 2.2.2.2 peer-group mygroup
Device(config-router)# end
```

OSPF 的相关信息

OSPF 是一项内部网关协议（IGP），是专门为 IP 网络设计的，它能够支持 IP 子网划分并且能够标记从外部获得的路由信息。OSPF 也能够提供数据包认证功能，以及使用 IP 组播发送和

接收数据包。Inspur 的实现方案支持 RFC 1253 中定义的 OSPF 管理信息库（MIB）。

注释： IP Base 镜像可以支持 OSPF。

Inspur 的 OSPF 实现符合 OSPF 版本 2 的定义，并且具有以下重要特性：

- 支持定义末节区域；
- 从任意 IP 路由协议学到的路由都可以重分发到另一种 IP 路由协议中。在域间级别上，这意味着 OSPF 可以导入从 EIGRP 和 RIP 学到的路由。OSPF 路由也可以被导入到 RIP 中；
- 支持在一个区域中，邻居路由器之间实施明文和 MD5 认证；
- 可以配置的路由接口参数包括接口输出开销、重传时间间隔、接口传输延迟、路由器优先级、路由器失效时间间隔和 Hello 时间间隔，以及认证密钥；
- 支持虚链路；
- 支持 RFC 1587 中定义的非完全末节区域（NSSA）。

OSPF 通常需要多种设备进行协调工作，比如内部路由器、连接多个区域的区域边界路由器（ABR），以及自治系统边界路由器（ASBR）。用户可以全部使用默认参数来实施最简配置，也就是没有认证，只是把接口分配到相应的区域中。如果用户需要自定义在自己的 OSPF 环境，必须确保所有路由器上的配置统一。

OSPF 的无间断转发

设备或交换机堆栈能够支持以下两个级别的无间断转发（NSF）：

OSPF NSF 感知

IP Services 特性集能够为 IPv4 提供 OSPF NSF 感知特性。当邻居路由器具有 NSF 功能时，三层设备可以在以下事件发生时继续从邻居路由器转发数据包：路由器中的主用路由处理器（RP）失效且备用 RP 接管工作的时间段内，或者用户手动重启主用 RP 来实施无间断软件升级的过程中。

这个特性默认是禁用的。

OSPF NSF 功能

IP Services 特性集除了像之前的版本一样能够支持 OSPFv2 NSF Inspur 格式外，还能够支持 OSPFv2 NSF IETF 格式。有关这个特性的更多信息，用户可以参考“*NSF——OSPF（RFC 3623 OSPF Graceful Restart）*”。

IP Services 特性集还能够为 IPv4 提供 OSPF NSF 功能路由，以此在堆栈主用设备迁移期间，实现更快的收敛和更少的流量丢失。当在具有 OSPF NSF 功能的堆栈中发生堆栈主用设备迁移时，新的堆栈主用设备必须执行以下两个任务，来重新同步 OSPF 邻居的链路状态数据库：

- 释放网络上可用的 OSPF 邻居，同时不重置邻居关系；
- 重新获取有关网络的链路状态数据库。

在堆栈主用设备迁移后，新的主用设备会向支持 NSF 感知功能的邻居设备发送 OSPF NSF 消息。收到这个消息后，邻居设备就知道自己不应该重置与这个堆栈建立的邻居关系。支持 NSF 功能的堆栈主用设备在从网络上的其他路由器收到这个消息后，它会开始重新建立邻居列表。

在重新建立邻居关系时，支持 NSF 功能的堆栈主用设备会与 NSF 感知邻居之间重新同步数据库，并且在 OSPF 邻居之间交换路由信息。新的堆栈主用设备会使用这些路由信息代替陈旧的路由，以此更新路由信息数据库（RIB），并以新的信息更新转发信息库（FIB）。之后 OSPF 协议就完全收敛了。

注释： 要想使用 OSPF NSF 特性，需要所有互联邻居设备都支持 NSF 感知特性。如果一台具有 NSF 功能的路由器在一个网段上感知到了一台不具备 NSF 感知功能的邻居，它会为这个网段禁用 NSF 功能。并且对其他网段上支持 NSF 感知或 NSF 功能的邻居继续提供 NSF 功能。

用户可以使用 OSPF 路由配置命令 **nsf** 来启用 OSPF NSF 路由功能。并使用特权 EXEC 命令 **show ip ospf** 来验证是否启用了 OSPF NSF 特性。

更多信息可以参考 *Inspur Nonstop Forwarding*：

http://www.icntnetworks.com

OSPF 区域参数

用户可以（可选的）配置一些 OSPF 区域参数。这些参数包括对一个区域、末节区域、非完全末节区域（NSSA）提供基于密码的认证博阿虎。OSPF 不会向末节区域中发送外部路由信息。区域边界路由器（ABR）会生成并向末节区域发送一条默认外部路由，以提供自治系统（AS）外部目的地的路由。OSPF 不会从核心区域向 NSSA 区域泛洪所有 LSA，但可以通过重分发向区域中注入 AS 外部路由。

路由汇总是指把多个通告地址合并为一条汇总路由，并把这条汇总路由通告到其他区域中。如果网络号是连续的，用户就可以使用路由器配置命令 **area range** 来配置 ABR，让它通过汇总路由通告这个范围能够覆盖的所有网络。

其他 OSPF 参数

用户可以（可选的）在路由器配置模式中配置其他 OSPF 参数。

- 路由汇总：用于从其他协议重分发路由时。每条路由都是单独在一个外部 LSA 中通告的。为了减少 OSPF 链路状态数据库的大小，用户可以使用路由器配置模式的命令 **summary-address**，以指定网络地址和掩码，通过一条路由重分布指定范围中的所有路由；
- 虚链路：在 OSPF 中，所有区域必须都连接到骨干区域。用户可以在骨干连接中断的环境中，把两台区域边界路由器作为虚链路的两个端点，建立一条虚链路。配置信息中包括标识其他虚拟端点（其他 ABR），以及两台路由器共有的非骨干链路（传输区域）。不能通过末节区域配置虚链路；
- 默认路由：在用户把外部路由重分发到 OSPF 路由域中后，这台路由器就自动成为了自治系统边界路由器（ASBR）。用户可以让 ASBR 生成一条默认路由并将其注入 OSPF 路由域中；
- 用户可以在特权 EXEC 模式的所有 OSPF **show** 命令中使用域名服务器（DNS）名称，这种显示方式能够使用户较轻松地识别路由器，而不是通过路由器 ID 或邻居 ID 来进行识别；
- 默认度量值：OSPF 在为接口计算 OSPF 度量值时，会参考接口的带宽。度量值的计算方法是 *ref-bw* 除以带宽，其中 *ref* 默认为 10，带宽（*bw*）是由接口配置模式的命令 **bandwidth** 指定的。对于多条高带宽链路来说，用户可以指定较大的值来区分这些链路上的开销；
- 管理距离是评估一个路由信息源是否可靠的参数，取值范围是 0 至 255 之间的整数，值越大表示可信度越低。管理距离为 255 表示这个路由信息源完全不可信，设备应该忽略它。OSPF 会使用三个不同的管理距离：区域内的路由（区域内）、其他区域的路由（区域间），以及通过重分发学到的其他路由域中的路由（外部）。用户可以改变这些距离值；
- 被动接口：由于一个以太网上两台设备之间的接口只表示一个网段，因此为了防止 OSPF 为发送方接口发送 Hello 包，用户必须把发送方设备配置为被动接口。使这两台设备能够通过为接收方接口发送的 Hello 包来识别彼此；
- 路由计算计时器：当 OSPF 收到拓扑变化事件时，它需要开始最短路径优先（SPF）计算，用户可以配置这两个事件之间的延迟时间，也可以配置两个 SPF 计算之间的保持时间；
- 记录邻居变化：用户可以配置路由器在 OSPF 邻居状态发生改变时，发送系统日志消息，提供路由器变化的高层视图。

LSA 组步调（Pacing）

OSPF LSA 组步调（Pacing）特性使路由器能够把 OSPF LSA 集合起来，并调整刷新、校验和老化功能，以提高路由器的使用效率。这个特性默认就是启用的，默认的步调间隔为 4 分钟，用户通常并不需要更改这个参数。在优化组步调间隔时，这个间隔与路由器需要刷新、校验和老化的 SLA 数量成反比。举例来说，如果数据库中有大约 10,000 个 LSA，那么减小步调间隔会有助于网络运行。如果数据库非常小（40 至 100 个 LSA），那么把步调间隔增加为 10 至 20 分钟会带来些许帮助。

环回接口

OSPF 会把接口上配置的最大 IP 地址作为自己的路由器 ID。如果这个接口失效了或被移除，OSPF 进程必须重新计算一个新的路由器 ID，并从自己的接口重新发送所有路由信息。如果环回接口上配置了 IP 地址，那么 OSPF 会使用这个地址作为自己的路由器 ID，即使这个地址并不是最大的 IP 地址。由于环回接口永远不会失效，因此这种做法提供了更强的可靠性。OSPF 会自动优选环回接口，并选择所有换回接口中最大的 IP 地址作为路由器 ID。

如何配置 OSPF

默认的 OSPF 配置

图 109：默认的 OSPF 配置

特性	**默认设置**
接口参数	开销：1 重传间隔：5 秒钟 传输延迟：1 秒钟 优先级：1 Hello 间隔：10 秒钟 失效（Dead）间隔：Hello 间隔的 4 倍 无认证 未指定密码

	禁用 MD5 认证
区域	认证类型：0（无认证） 默认开销：1 范围（Range）：禁用 末节：未定义末节区域 NSSA：未定义 NSSA 区域
自动开销	100 Mbit/s
生成默认信息	禁用。当启用后，默认度量值设置为 10，外部路由类型默认为类型 2
默认度量值	内建的自动度量值转换，适用于每个路由协议
距离 OSPF	dist1（一个区域内的所有路由）：110。dist2（从一个区域到另一个区域的所有路由）：110。dist3（从其他路由域来的路由）：110
OSPF 数据库过滤器	禁用。从接口泛洪所有出向链路状态通告（LSA）
IP OSPF 域名查找	禁用
记录邻接的变化	启用
邻居	未指定
邻居数据库过滤器	禁用。向邻居泛洪所有出向 LSA
网络区域	禁用
无间断转发（NSF）感知	启用。三层交换机能够在硬件或软件更换期间，继续从具有 NSF 功能的邻居路由器那里转发数据包
NSF 功能	禁用。 **注释：** 交换机堆栈能够为 IPv4 提供 OSPF NSF 功能的支持
路由器 ID	无定义的 OSPF 路由进程
汇总地址	禁用
计时器 LSA 组步调	240 秒钟
计时器最短路径优先（spf）	SPF 延迟：5 秒钟；SPF 保持时间：10 秒钟
虚链路	未指定区域 ID 或路由器 ID Hello 间隔：10 秒钟 重传间隔：5 秒钟 传输延迟：1 秒钟 失效（Dead）间隔：40 秒钟 认证密钥：无预定义密钥

	消息摘要密钥（MD5）：无预定义密钥

配置基本的 OSPF 参数

要想启用 OSPF，用户需要创建一个 OSPF 路由进程、指定关联到这个路由进程的 IP 地址范围，并且为这个范围分配一个区域 ID。对于运行 IP Services 镜像的交换机来说，用户可以配置 Inspur OSPFv2 NSF 格式，或者 IETF OSPFv2 NSF 格式。

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**router ospf** *process-id* **示例：** `Device(config)# router ospf 15`	启用 OSPF 路由并进入路由器配置模式。进程 ID 是用户分配的在内部使用的标识参数，可以是任意正整数。每个 OSPF 路由进程都有一个唯一的值。 **注释：** OSPF 路由访问仅支持一个 OSPFv2 和一个 OSPFv3 实例，最多有 200 条动态学习的路由
步骤 3	**nsf inspur** [**enforce global**] **示例：** `Device(config)# nsf inspur enforce global`	（可选）为 OSPF 启用 Inspur NSF 特性。关键字 **enforce global** 用来在设备检测到非 NSF 感知功能的邻居设备时，取消 NSF 重启动。 **注释：** 在步骤 3 或步骤 4 中输入这条命令，然后配置步骤 5
步骤 4	**nsf ietf** [**restart-interval** *seconds*] **示例：** `Device(config)# nsf ietf restart-interval 60`	（可选）为 OSPF 启用 IETF NSF 特性。关键字 **restart-interval** 指定了平滑重启间隔的长度，以秒为单位。取值范围是 1 至 1800。默认值为 120。 **注释：** 在步骤 3 或步骤 4 中输入这条命令，然后配置步骤 5

步骤 5	**network** address wildcard-mask **area** area-id **示例：** `Device(config)# network 10.1.1.1 255.240.0.0 area 20`	定义运行 OSPF 的接口，以及接口所属的区域 ID。用户可以使用通配符掩码，在单条命令中定义要与特定 OSPF 区域关联的一个或多个多个接口。区域 ID 可以是十进制数值，也可以是 IP 地址
步骤 6	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 7	**show ip protocols** **示例：** `Device# show ip protocols`	检查用户输入的信息
步骤 8	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 OSPF 接口

用户可以使用接口配置模式的命令 **ip ospf** 来修改接口的 OSPF 参数。用户不必修改这些参数，但有些接口参数（Hello 间隔、失效间隔和认证密钥）必须在网络中的所有路由器上统一。如果用户修改了这些参数，要确保网络中所有路由器上的这些参数都相同。

注释： 接口配置模式的命令 **ip ospf** 都是可选的。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 2	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/1`	进入接口配置模式，并指定用户想要配置的接口
步骤 3	**ip ospf** *cost* **示例：** `Device(config-if)# ip ospf 8`	（可选）明确指定在接口上发送数据包的开销
步骤 4	**ip ospf retransmit-interval** *seconds* **示例：** `Device(config-if)# ip ospf transmit-interval 10`	（可选）指定传输链路状态通告之间的秒数。取值范围是 1 至 65535 秒，默认值为 5 秒钟
步骤 5	**ip ospf transmit-delay** *seconds* **示例：** `Device(config-if)# ip ospf transmit-delay 2`	（可选）设置发送链路状态更新数据包之前等待的秒数。取值范围是 1 至 65535 秒，默认值为 1 秒钟
步骤 6	**ip ospf priority** *number* **示例：** `Device(config-if)# ip ospf priority 5`	（可选）设置优先级，有助于确定网络中的 OSPF 指定路由器。取值范围是 0 至 255，默认值为 1
步骤 7	**ip ospf hello-interval** *seconds* **示例：** `Device(config-if)# ip ospf hello-interval 12`	（可选）设置 OSPF 接口发送 Hello 数据包的间隔秒数。网络中所有节点上的 Hello 间隔必须都相同。取值范围是 1 至 65535 秒，默认值为 10 秒钟
步骤 8	**ip ospf dead-interval** *seconds* **示例：**	（可选）设置在从邻居收到最后一个 Hello 数据包的多少秒之后，认为这个 OSPF 路由器已失效。网络中所有节点

	Device(config-if)# ip ospf dead-interval 8	上的失效间隔必须相同。取值范围是 1 至 65535，默认值为 Hello 间隔的 4 倍
步骤 9	**ip ospf authentication-key** *key* **示例：** Device(config-if)# ip ospf authentication-key password	（可选）指定邻居 OSPF 路由器使用的密码。密码可以由任意字符构成，最长 8 字节。一个网络中的所有邻居路由器必须使用相同的密码来交换 OSPF 信息
步骤 10	**ip ospf message digest-key** *keyid* **md5** *key* **示例：** Device(config-if)# ip ospf message digest-key 16 md5 your1pass	（可选）启用 MD5 认证。 • *keyid*——取值从 1 至 255 的识别符 • *key*——由字母和数字构成的密码，最长 16 字节
步骤 11	**ip ospf database-filter all out** **示例：** Device(config-if)# ip ospf database-filter all out	（可选）禁止在这个接口上泛洪 OSPF LSA。默认情况下，OSPF 会在同一个区域中的除了收到这个 LSA 的接口之外的所有接口上泛洪新的 LSA
步骤 12	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 13	**show ip ospf interface** [*interface-name*] **示例：** Device# show ip ospf interface	显示与 OSPF 相关的接口信息
步骤 14	**show ip ospf neighbor detail** **示例：** Device# show ip ospf neighbor	显示邻居交换机的 NSF 感知状态。输出信息会匹配以下示例： • *Options is 0x52* *LLS Options is 0x1 (LR)*

	detail	同时显示这两行时，表示邻居交换机具有 NSF 感知功能 • *Options is 0x42*——表示邻居交换机不具备 NSF 感知功能
步骤 15	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

配置 OSPF 区域参数

在开始前

注释： OSPF 路由器配置命令 **area** 都是可选配置。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式
步骤 2	**router ospf** *process-id* **示例：** Device(config)# router ospf 109	启用 OSPF 路由并进入路由器配置模式。
步骤 3	**area** *area-id* **authentication** **示例：** Device(config-router)# area 1 authentication	（可选）对指定区域实施基于密码的访问保护。识别符可以是十进制数值，也可以是 IP 地址
步骤 4	**area** *area-id* **authentication**	（可选）在区域上启用 MD5 认证

	message-digest **示例：** `Device(config-router)# area 1 authentication message-digest`	
步骤 5	**area** *area-id* **stub** [**no-summary**] **示例：** `Device(config-router)# area 1 stub`	（可选）把一个区域定义为末节区域。关键字 **no-summary** 用来阻止 ABR 向末节区域发送汇总链路通告
步骤 6	**area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate**] [**no-summary**] **示例：** `Device(config-router)# area 1 nssa default-information-originate`	（可选）定义一个区域为非完全末节区域（NSSA）。同一个区域中的所有路由器都要认同这个区域是 NSSA。用户可以从以下关键字中选择一个： • **no-redistribution**——当路由器是 NSSA ABR 并且用户希望使用 **redistribute** 命令向普通区域（而不是向 NSSA 区域）中注入路由时使用这条命令 • **default-information-originate**——在 ABR 上进行配置，允许把类型 7 LSA 注入到 NSSA 区域中 • **no-redistribution**——不要把汇总 LSA 发送到 NSSA 区域中
步骤 7	**area** *area-id* **range** *address mask* **示例：** `Device(config-router)# area 1 range 255.240.0.0`	（可选）指定一个地址范围，通告为一条路由。只在区域边界路由器上配置这条命令
步骤 8	**end** **示例：**	返回特权 EXEC 模式

	Device(config)# **end**	
步骤 9	**show ip ospf** [*process-id*] **示例：** Device# show ip ospf	显示有关通用 OSPF 路由进程的信息，或指定进程 ID 来验证配置的正误
步骤 10	**show ip ospf** [*process-id* [*area-id*]] **database** **示例：** Device# show ip osfp database	显示指定路由器上与 OSPF 数据库相关的信息
步骤 11	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

配置其他 OSPF 参数

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式
步骤 2	**router ospf** *process-id* **示例：** Device(config)# router ospf 10	启用 OSPF 路由并进入路由器配置模式。
步骤 3	**summary-address** *address mask* **示例：** Device(config)#	（可选）为重分发路由指定地址和 IP 子网掩码，以便只通告一条汇总路由

	summary-address 10.1.1.1 255.255.255.0	
步骤 4	**area** *area-id* **virtual-link** *router-id* [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] [**trans**] [[**authentication-key** *key*] | **message-digest-key** *keyed* **md5** *key*]] **示例：** Device(config)# area 2 virtual-link 192.168.255.1 hello-interval 5	（可选）建立一条虚链路并设置其参数
步骤 5	**default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-name*] **示例：** Device(config)# default-information originate metric 100 metric-type 1	（可选）让 ASBR 生成一条默认路由并将其注入 OSPF 路由域中。所有参数都是可选的
步骤 6	**ip ospf name-lookup** **示例：** Device(config)# ip ospf name-lookup	（可选）配置 DNS 域名查找特性。该特性默认是禁用的
步骤 7	**ip auto-cost reference-bandwidth** *ref-bw* **示例：** Device(config)# ip auto-cost reference-bandwidth 5	（可选）指定一个地址范围，以便以单一路由的形式进行通告。只在区域边界路由器上使用这条命令
步骤 8	**distance ospf** {[**inter-area** *dist1*] [**inter-area** *dist2*] [**external** *dist3*]}	（可选）改变 OSPF 距离值。默认每种类型的距离值是 110，取值范围是 1 至

	示例： `Device(config)# distance ospf inter-area 150`	255
步骤 9	**passive-interface** *type number* **示例：** `Device(config)# passive-interface gigabitethernet 1/0/6`	（可选）禁止从指定接口向外发送 Hello 数据包
步骤 10	**timers throttle spf** *spf-delay spf-holdtime spf-wait* **示例：** `Device(config)# timers throttle spf 200 100 100`	（可选）配置路由计算计时器。 • *spf-delay*——取值范围是 1 至 600000 毫秒 • *spf-holdtime*——第一次与第二次 SPF 计算之间的延迟。取值范围是 1 至 600000 毫秒 • *spf-wait*——等待 SPF 计算的最大时间。取值范围是 1 至 600000 毫秒
步骤 11	**ospf log-adj-changes** **示例：** `Device(config)# ospf log-adj-changes`	（可选）当邻居状态发生变化时发送系统日志消息
步骤 12	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 13	**show ip ospf** [*process-id* [*area-id*]] **database** **示例：**	显示与某台路由器的 OSPF 数据库相关的信息列表

	Device# show ip ospf database	
步骤 14	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

改变 LSA 组步调

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式
步骤 2	**router ospf** *process-id* **示例：** Device(config)# router ospf 25	启用 OSPF 路由并进入路由器配置模式。
步骤 3	**timers lsa-group-pacing** *seconds* **示例：** Device(config-router)# timers lsa-group-pacing 15	改变 LSA 组步调
步骤 4	**end** **示例：** Device(config-router)# **end**	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** Device# **show running-config**	检查用户输入的信息

步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置环回接口

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**interface loopback 0** **示例：** `Device(config)# interface loopback 0`	创建一个环回接口并进入接口配置模式
步骤 3	**ip address** *address mask* **示例：** `Device(config-if)# ip address 10.1.1.5 255.255.240.0`	为接口分配一个 IP 地址
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show ip interface** **示例：** `Device# show ip interface`	检查用户输入的信息

步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

监控 OSPF

用户可以查看特定的状态统计信息，比如 IP 路由表的内容、缓存和数据库。

表 110：显示 IP OSPF 状态统计信息命令

命令	目标
show ip ospf [*process-id*]	显示 OSPF 路由进程的通用信息
show ip ospf [*process-id*] **database** [**router**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**router**] [**self-originate**] **show ip ospf** [*process-id*] **database** [**router**] [**adv-router** [*ip-address*]] **show ip ospf** [*process-id*] **database** [**network**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**summary**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**asbr-summary**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**external**] [*link-state-id*] **show ip ospf** [*process-id area-id*] **database** [**database-summary**]	显示有关 OSPF 数据库的信息列表
show ip ospf border-routes	显示内部 OSPF 路由 ABR 和 ASBR 表条目
show ip ospf interface [*interface-name*]	显示与 OSPF 相关的接口信息
show ip ospf neighbor [*interface-name*]	显示 OSPF 接口邻居信息

[*neighbor-id*] **detail**	
show ip ospf virtual-links	显示与 OSPF 相关的虚链路信息

OSPF 的配置示例

示例：配置基本的 OSPF 参数

以下示例展示了如何配置一个 OSPF 路由进程并为其分配进程号 109：

```
Device(config)# router ospf 109
Device(config-router)# network 131.108.0.0 255.255.255.0 area 24
```

EIGRP 的相关信息

增强型 IGRP（EIGRP）是 Cisco 私有的 IGRP 增强版本。EIGRP 使用与 IGRP 相同的距离矢量算法和距离信息；但 EIGRP 的收敛属性和运行效率都得到了显著提高。

EIGRP 把被称为扩散更新算法（DUAL）的算法作为收敛技术，在整个路由计算中的每时每刻都保证不出现环路，并且允许拓扑中涉及的所有设备同时进行同步。不受拓扑变化影响的路由器不参与重新计算。

IP EIGRP 提供了可扩展的网络宽度。在使用 RIP 时，网络的最大宽度为 15 跳。由于 EIGRP 的度量值足够大，能够支持上千跳，因此限制网络扩展范围的唯一障碍是传输层跳数限制。EIGRP 只会在 IP 数据包穿越 15 台路由器并且去往目的地的下一跳是通过 EIGRP 获知的时，才会增加传输控制字段。当 RIP 路由被用作去往目的地的下一跳时，传输控制字段会如常增加。

EIGRP 的特性

EIGRP 提供了以下特性：

- 快速收敛；
- 目的地状态发生改变时的增量更新，不再发送整个路由表的内容，减小了 EIGRP 数据包对于带宽的需求；

- 更少的 CPU 利用率，因为不用每次都接收并处理完整更新的数据包；
- 使用与协议无关的邻居发现机制来发现邻居路由器；
- 可变长子网掩码（VLSM）；
- 任意的路由汇总；
- EIGRP 能够适用于大型网络。

EIGRP 的组成部分

EIGRP 拥有以下四个基本组成部分：

- 邻居发现和恢复是路由器用来动态学习直连网络上其他路由器的过程。当邻居便得不可达或无法运作时，路由器必须能够发现这一情况。邻居发现和恢复通过周期性发送小 Hello 包，以低开销实现了这一需求。在收到 Hello 包后，Inspur INOS 软件就能够学习到邻居的在线情况和运行情况。当检测到这个状态后，路由器就能够和邻居路由器之间交换路由信息了；
- 使用可靠的传输协议负责向所有邻居确保有序的传输 EIGRP 数据包。EIGRP 支持混合传输组播和单播数据包。一些 EIGRP 数据包必须通过可靠的方式进行发送，其他 EIGRP 数据包不必通过可靠的方式进行发送。为了提高效率，EIGRP 只会在必要时为数据包提供可靠传输。举例来说，在具有组播功能的多路访问网络（例如以太网）上，没有必要单独通过可靠的方式向所有邻居发送 Hello 包。因此，EIGRP 会发送单个组播 Hello 包，并在数据包中告知接收该数据包的路由器，不需要对这个数据包进行确认。其他类型的数据包（例如更新）需要进行确认，数据包中会携带确认要求。当存在未确认的数据包时，可靠传输能够快速重新发送这部分组播数据包。这样做有助于确保在速度不同的链路上，保持较低的收敛时间；
- DUAL 有限状态机描述了所有路由计算的决策过程。它会追踪所有邻居通告的所有路由。DUAL 使用距离信息（也就是度量值）来选择有效的无环路径。DUAL 会根据可行后继（Feasible Successor）来选择把哪些路由放入路由表中。后继（Successor）是指设备用来转发数据包的邻居路由器，这个邻居提供了去往目的地的最低开销路径，并且保证这是一条无环路径。当没有可行后继，但有多个邻居通告了同一个目的地时，DUAL 就必须重新进行计算。这个过程也就决定了新的后继。用来重新计算路由的时间会影响到收敛时间。重新计算的过程需要消耗处理器资源；如果不是必要的话，尽量避免重新计算。当拓扑发生变化时，DUAL 会对可行后继进行测试。如果有可行后继的话，DUAL 就会尽量使用可行后继，来避免不必要的重新计算；
- 与协议相关的模块会负责域网络层相关的工作。比如 IP EIGRP 模块，它负责发送和接收

封装在 IP 中的 EIGRP 数据包。它还负责解析 EIGRP 数据包，并通知 DUAL 它收到的新信息。EIGRP 会要求 DUAL 做出路由决策，但决策会储存在 IP 路由表中。EIGRP 还会负责分发从其他 IP 路由协议学到的路由。

注释： 要想启用 EIGRP，设备或堆栈主用设备必须运行 IP Services 特性集。

EIGRP 无间断转发

设备或交换机堆栈能够支持以下两个级别的 EIGRP 无间断转发（NSF）：

- EIGRP NSF 感知
- EIGRP NSF 功能

EIGRP NSF 感知

IP Services 特性集能够为 IPv4 提供 OSPF NSF 感知特性。当邻居路由器具有 NSF 功能时，三层设备可以在以下事件发生时继续从邻居路由器转发数据包：路由器中的主用路由处理器（RP）失效且备用 RP 接管工作的时间段内，或者用户手动重启主用 RP 来实施无间断软件升级的过程中。

用户无法禁用这个特性。有关该特性的更多信息，用户可以参考 *Inspur INOS IP Routing Protocols Configuration Guide, Release 12.4* 的“EIGRP Nonstop Forwarding (NSF) Awareness”部分。

EIGRP NSF 功能

IP Services 特性集能够为 IPv4 提供 EIGRP NSF 功能路由，以此在堆栈主用设备迁移期间，实现更快的收敛和更少的流量丢失。更多有关 NSF 功能的信息，用户可以参考 *High Availability Configuration Guide, Inspur INOS* 中“Configuring Nonstop Forwarding”一章。

IP Services 特性集能够为 IPv4 提供 EIGRP NSF 功能路由，以此在堆栈主用设备迁移期间，实现更快的收敛和更少的流量丢失。当具有 EIGRP NSF 功能的堆栈主用设备重启、新的堆栈主用设备启动且 NSF 重启时，设备是没有邻居的，拓扑表也是空的。设备必须把接口打开、重新获取邻居信息，并重新建立拓扑表和路由表，同时还不中断发往设备堆栈的流量。EIGRP 对等体路由器会保留从新堆栈主用设备学到的路由，并在 NSF 重启过程中继续转发流量。

为了防止邻居重置邻接关系，新的堆栈主用设备会在 EIGRP 数据包头部使用新的重启（RS）比特，来说明重启事件。邻居接收到这个信息后，它会在对等体列表中的堆栈进行同步，并

保留与堆栈建立的邻接关系。接着邻居会把它的拓扑表发送给堆栈主用设备，同时也设置 RS 比特，表示自己具有 NSF 感知功能并且会协助新的堆栈主用设备。

如果至少有一个堆栈对等体邻居具有 NSF 感知功能，堆栈主用设备就能够接收更新并重建自己的数据库。每个具有 NSF 感知功能的邻居都会在最新的更新数据包中发送表结束（EOT）标记，来标记表内容的结束。堆栈主用设备在收到 EOT 标记后就会识别出收敛时间，并且自己会开始发送更新。当堆栈主用设备从所有邻居那里都接收到了 EOT 标记，或者当 NSF 收敛计时器超时后，EIGRP 就会告诉路由信息数据库（RIB）这个收敛事件，并开始向所有 NSF 感知对等体泛洪自己的拓扑表。

EIGRP 末节路由

所有特性集都可以使用 EIGRP 末节路由特性，它通过把路由流量移动到更靠近终端用户的方式减少了资源利用率。

注释： IP Base 特性集中包含 EIGRP 末节路由功能，并且只能向网络中的其他设备通告路由表中的直连路由或汇总路由。在接入层设备上使用 EIGRP 末节路由特性，可以让设备无需使用其他类型的路由通告。要想使用高级功能和完整的 EIGRP 路由功能，设备必须运行 IP Base 特性集。在运行 IP Base 特性集的设备上，如果用户尝试同时配置 Multi-VRF-CE 特性和 EIGRP 末节路由特性，设备会拒绝配置命令。IP Base 特性集不支持 IPv6 EIGRP 末节路由特性。

在使用了 EIGRP 末节路由特性的网络中，只能由配置了 EIGRP 末节路由特性的设备，来对用户的 IP 流量进行路由。设备会把流量发送到配置为用户接口的接口，或者连接其他设备的接口。

在使用 EIGRP 末节路由特性时，用户需要配置分布层和远端路由器来使用 EIGRP 并把设备配置为末节设备。设备只能传播指定的路由。设备能够对所有汇总路由、直连路由和路由更新请求做出响应。

任何邻居在收到对端设备告知自己为末节状态的数据包后，都不会向末节路由器查询任何路由，并且拥有末节对等体的路由器不会向这个对等体进行查询。末节路由器依赖于分布路由器，来将自己的更新发送给所有对等体。

如下图所示，用户把设备 B 配置为 EIGRP 末节路由器。设备 A 和设备 C 连接着 WAN 的其余部分。设备 B 向设备 A 和设备 C 通告了直连路由、静态路由、重分发路由和汇总路由。设备 B 没有通告任何从设备 A 学到的路由（反之亦然）。

图 93：EIGRP 末节路由器配置

Routed to WAN	去往 WAN 的路由

Switch A	交换机 A
Switch B	交换机 B
Switch C	交换机 C
Host A	主机 A
Host B	主机 B
Host C	主机 C

更多有关 EIGRP 末节路由的信息，用户可以参考 *Inspur INOS IP Configuration Guide, Volume 2 of 3: Routing Protocols* 中“Configuring EIGRP Stub Routing”部分。

如何配置 EIGRP

要想创建一个 EIGRP 路由进程，用户必须启用 EIGRP 并关联相关的网络。EIGRP 会从属于指定网络的接口发送更新。如果用户没有指定接口网络，EIGRP 就不会通告任何 EIGRP 更新。

注释： 如果用户网络中有一些路由器配置了 IGRP，并且用户希望改用 EIGRP，用户就必须指定转换路由器，也就是同时配置了 IGRP 和 EIGRP 的路由器。在这种情况中，用户需要执行下述的步骤 1 至步骤 3，并且还要查看“配置水平分割”部分。用户必须使用相同的 AS 号，才能自动重分发路由。

默认的 EIGRP 配置

表 111：默认的 EIGRP 配置

特性	**默认设置**
自动汇总	禁用
默认信息	在进行重分发时接受外部路由，并且在 EIGRP 进程之间传递默认信息
默认度量值	只有直连路由和接口静态路由可以在重分发时没有默认度量值。度量值包括以下参数： • 带宽：0 或更大的 kbit/s • 延迟（几十微秒）：0 或 39.1 纳秒的倍数 • 可靠性：0 至 255 之间的任何数值（255 表示 100%可靠） • 负载：有效带宽，0 至 255 之间的任何数值（255 表示 100%

	负载） • MTU：最大传输单元，以字节为单位的路由大小。0 或任意正整数
距离	内部距离：90 外部距离：170
EIGRP 记录邻居变化	禁用。不记录邻接关系的变化
IP 认证密钥链	不提供认证
IP 认证模式	不提供认证
IP 带宽百分比	50%
IP Hello 间隔	对于低速非广播多访问（NBMA）网络：60 秒钟 对于所有其他网络：5 秒钟
IP 保持（Hold）时间	对于低速 NBMA 网络：180 秒钟 对于所有其他网络：15 秒钟
IP 水平分割	启用
IP 汇总地址	没有预定义的汇总地址
度量值权重	tos：0；k1 和 k3：1；k2、k4 和 k5：0
网络	未指定
无间断转发（NSF）感知	在运行 IP Services 特性集的交换机上为 IPv4 启用。所有三层交换机会在硬件或软件发生变化期间，继续从具有 NSF 功能的邻居路由器那里转发数据包
NSF 功能	禁用 **注释：** 设备为 IPv4 支持 EIGRP NSF 功能路由
偏移列表	禁用
路由器 EIGRP	禁用
设置度量值	route-map 中没有设置度量值
流量共享	按照度量比率进行分布
变量	1（等价负载均衡）

配置基本的 EIGRP 参数

在开始前

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**router eigrp** *autonomous-system* **示例：** `Device(config)# router eigrp 10`	启用 EIGRP 路由进程，并进入路由器配置模式。AS 号标识去往其他 EIGRP 路由器的路由，并且用于标记路由信息
步骤 3	**nsf** **示例：** `Device(config)# nsf`	（可选）启用 EIGRP NSF。在堆栈主用设备以及它的所有对等体上输入这条命令
步骤 4	**network** *network-number* **示例：** `Device(config)# network 192.168.0.0`	把网络关联到 EIGRP 路由进程。EIGRP 会从指定网络中的接口发送更新
步骤 5	**eigrp log-neighbor-changes** **示例：** `Device(config)# eigrp log-neighbor-changes`	（可选）为 EIGRP 邻居的变化启用日志记录功能，用来监控路由系统的稳定性
步骤 6	**metric weights** *tos k1 k2 k3 k4 k5* **示例：** `Device(config)# metric weights 0 2 0 2 0 0`	（可选）调整 EIGRP 度量值。尽管默认值是经过精心设置，旨在为大多数网络提供最优的运作，但用户可以对其进行调整。 **注意：** 设置度量值是复杂的工作，如果没有资深网络设计者的指导，不建议用户随意更改
步骤 7	**offset-list** [*access-list number* | *name*]	（可选）对路由度量值应用一个偏移列

	{**in** | **out**} *offset* [*type number*] **示例：** `Device(config)# offset-list 21 out 10`	表，来增加从 EIGRP 学到的入站和出站路由的度量值。用户可以使用访问列表或接口来限制偏移列表
步骤 8	**auto-summary** **示例：** `Device(config)# auto-summary`	（可选）把子网路由汇总为网络级别的路由
步骤 9	**ip summary-address eigrp** *autonomous-system-number address mask* **示例：** Device(config)# **ip summary-address eigrp 1 192.168.0.0 255.255.0.0**	（可选）配置汇总路由
步骤 10	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 11	**show ip protocols** **示例：** `Device# show ip protocols`	检查用户输入的信息。 要想检查 NSF 感知功能，输出信息如下所示： *** IP Routing is NSF aware *** EIGRP NSF enabled
步骤 12	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 EIGRP 接口

用户可以在接口上配置其他可选的 EIGRP 参数。

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/1`	进入接口配置模式，并指定用户想要配置的三层接口
步骤 3	**ip bandwidth-percent eigrp** *percent* **示例：** `Device(config-if)# ip bandwidth-percent eigrp 60`	（可选）配置接口上 EIGRP 能够使用的带宽百分比。默认值是 50%
步骤 4	**ip summary-address eigrp** *autonomous-system-number address mask* **示例：** `Device(config-if)# ip summary-address eigrp 109 192.161.0.0 255.255.0.0`	（可选）为指定接口配置汇总地址（如果自动汇总功能一起用的话，一般不必配置这条命令）
步骤 5	**ip hello-interval eigrp** *autonomous-system-number seconds* **示例：** `Device(config-if)# ip`	（可选）改变 EIGRP 路由进程的 Hello 时间间隔。取值范围是 1 至 65535 秒。默认值为：低速 NBMA 网络 60 秒钟，所有其他网络 5 秒钟

	hello-interval eigrp 109 10	
步骤 6	**ip hold-time eigrp** *autonomous-system-number seconds* **示例：** Device(config-if)# ip hold-time eigrp 109 40	（可选）改变 EIGRP 路由进程的保持（Hold）时间间隔。取值范围是 1 至 65535 秒。默认值为：低速 NBMA 网络 180 秒钟，所有其他网络 15 秒钟。 **注意：** 在咨询 Inspur 技术支持之前，不要调整保持时间
步骤 7	**no ip split-horizon eigrp** *autonomous-system-number* **示例：** Device(config-if)# no ip split-horizon eigrp 109	（可选）禁用水平分割，允许路由器从接收到路由信息的接口向外通告该路由信息
步骤 8	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 9	**show ip eigrp interface** **示例：** Device# show ip eigrp interface	显示哪些接口上启用了 EIGRP，以及与这些接口相关的 EIGRP 信息
步骤 10	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

配置 EIGRP 路由认证

EIGRP 路由认证为 EIGRP 路由协议提供了使用 MD5 对路由更新进行认证的功能，以防止从未经批准的源引入未授权的路由消息或伪路由消息。

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/1`	进入接口配置模式，并指定用户想要配置的三层接口
步骤 3	**ip authentication mode eigrp** *autonomous-system* **md5** **示例：** `Device(config-if)# ip authentication mode eigrp 104 md5`	在 IP EIGRP 数据包中启用 MD5 认证
步骤 4	**ip authentication key-chain eigrp** *autonomous-system key-chain* **示例：** `Device(config-if)# ip authentication key-chain eigrp 105 chain1`	启用 IP EIGRP 数据包的认证
步骤 5	**exit** **示例：** `Device(config-if)# exit`	返回全局配置模式
步骤 6	**key chain** *name-of-chain* **示例：** `Device(config)# key chain`	指定密钥链并进入密钥链配置模式。要域步骤 4 中配置的名称相匹配

	chain1	
步骤 7	**key** *number* **示例：** Device(config-keychain)# key 1	在密钥链配置模式中指定密钥编号
步骤 8	**key-string** *text* **示例：** Device(config-keychain-key)# key-string key1	在密钥链配置模式中指定密钥字符串
步骤 9	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **示例：** Device(config-keychain-key)# accept-lifetime 13:30:00 Jan 25 2011 duration 7200	（可选）指定能够使用这个密钥的时间周期。 *start-time* 和 *end-time* 可以是 *hh:mm:ss Month date year* 或 *hh:mm:ss date Month year*。默认值是永远和默认的 *start-time*，也就是最早可以接受的日期，比如 1993 年 1 月 1 日。默认的 *end-time* 和 **duration** 是 **infinite**
步骤 10	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **示例：** Device(config-keychain-key)# send-lifetime 14:00:00 Jan 25 2011 duration 3600	（可选）指定能够发送这个密钥的时间周期。 *start-time* 和 *end-time* 可以是 *hh:mm:ss Month date year* 或 *hh:mm:ss date Month year*。默认值是永远和默认的 *start-time*，也就是最早可以接受的日期，比如 1993 年 1 月 1 日。默认的 *end-time* 和 **duration** 是 **infinite**
步骤 11	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 12	**show key chain**	显示认证的密钥信息

	示例： `Device# show key chain`	
步骤 13	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

监控和维护 EIGRP

用户可以从邻居表中删除邻居。用户还可以查看各种不同的EIGRP路由状态统计信息。下面这个表格中列出了特权EXEC模式中的命令，用来删除邻居和显示状态统计信息。有关显示字段的相关解释信息，用户可以参考*Inspur INOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.4*。

表112：IP EIGRP clear和show命令

命令	目的
clear ip eigrp neighbors [*if-address* | *interface*]	从邻居表汇总删除邻居
show ip eigrp interface [*interface*] [*as number*]	显示配置了EIGRP的接口信息
show ip eigrp neighbors [*type-number*]	显示EIGRP发现的邻居
show ip eigrp topology [*autonomous-system-number*] | [[*ip-address*] *mask*]]	显示指定进程的EIGRP拓扑表
show ip eigrp traffic [*autonomous-system-number*]	显示所有或指定EIGRP进程发送和接收的数据包数量

BGP 的相关信息

边界网关协议（BGP）是一项外部网关协议，用来建立域间路由系统，确保在自治系统之间实现无环的路由信息交互。自治系统是由处于相同管理域的路由器组成的，自治系统的边界

之内运行内部网关协议（IGP），比如 RIP 或 OSPF；多个自治系统通过外部网关协议（EGP）连接在一起。BGP 版本 4 是 Internet 中使用的域间路由标准 EGP。BGP 协议定义在 RFC 1163、1267 和 1771 文档中。有关 BGP 的更多信息，用户可以参考 *Internet Routing Architectures*，由 Cisco Press 出版；以及 *Inspur IP and IP Routing Configuration Guide* 中的“Configuring BGP”一章。

有关 BGP 命令和关键字的更多信息，用户可以参考 *Inspur INOS IP Command Reference, Volume 2 of 3: Routing Protocols* 中的“IP Routing Protocols”部分。

BGP 网络拓扑

属于同一个自治系统（AS）且交换 BGP 更新的路由器之间运行内部 BGP（IBGP），属于不同自治系统且交换 BGP 更新的路由器之间运行外部 BGP（EBGP）。配置 EBGP 和 IBGP 的大多数配置命令都相同。不同之处在于路由更新是在自治系统之间（EBGP）交换，还是在自治系统之内（IBGP）交换。下图展示了同时运行 EBGP 和 IBGP 的网络。

图 94：EBGP、IBGP 和多自治系统

Router A	路由器 A
Router D	路由器 D
Router B	路由器 B
Router C	路由器 C

在与外部 AS 交换信息之前，BGP 要确保 AS 内部的网络是可达的，可以通过在 AS 内与其他路由器之间建立内部 BGP 对等体来实现这一目的，或者通过把 BGP 路由信息重分发到 AS 内运行的 IGP 中来实现这一目的，比如 IGRP 和 OSPF。

运行 BGP 路由进程的路由器通常被称为 BGP 设备（BGP Speaker）。BGP 会使用传输控制协议（TCP）作为自己的传输协议（通常使用 179 端口）。两台 BGP 设备之间会建立 TCP 连接来交换路由信息，它们也称为对等体或邻居。在上图中，路由器 A 和路由器 B 就是 BGP 对等体，路由器 B 和路由器 C、以及路由器 C 和路由器 D 也是 BGP 对等体。路由信息是一系列 AS 号，描述了去往目的网络的完整路径。BGP 使用这个信息来构建无环的自治系统拓扑。

网络具有以下特征：

- 路由器 A 和路由器 B 上运行 EBGP，路由器 B 和路由器 C 上运行 IBGP。需要注意的是，EBGP 对等体之间是直连关系，IBGP 对等体之间不是直连关系。只要网络中运行了 IGP，使两个邻居之间能够相互访问，IBGP 对等体就不必是直连的；
- 一个 AS 内部的所有 BGP 设备彼此之间都会建立对等体关系。也就是说，一个 AS 内部的 BGP 设备之间是在逻辑上全互联的。BGP4 提供了两种机制来规避逻辑全互联的需求：

联盟和路由反射器；

- AS 200 是 AS 100 和 AS 300 的传输 AS——也就是说，AS 200 用来传输 AS 100 和 AS 300 之间的数据包。

BGP 对等体初始时会交换完整的 BGP 路由表，此后只发送增量更新。BGP 对等体之间也会交换存活消息（来确保连接是正常工作的），以及通知消息（对错误或特殊时间做出响应）。在 BGP 中，每条路由由一个网络号、穿越的自治系统列表（自治系统路径），以及其他路径属性列表构成。BGP 系统的主要功能是与其他 BGP 系统交换网络可达性信息，其中包括 AS 路径（AS Path）信息。这个信息可以用来确定 AS 连通性、消除环路，以及施加 AS 级别的策略决策。

运行 Inspur INOS 的路由器或设备不会选择或使用 IBGP 路由，除非它有去往下一跳路由器可用的路由，并且它从 IGP 收到了同步（除非禁用了 IGP 同步）。

当有多条路由可用时，BGP 根据属性值来做出路径选择。有关 BGP 属性的更多信息，用户可以参考“配置 BGP 决策属性”部分。

BGP 版本 4 能够支持无类域间路由（CIDR），因此用户可以通过创建聚合路由来缩小路由表的大小。CIDR 在 BGP 中消除了网络类别的个 i 呵年，并且支持通告 IP 前缀。

无间断转发感知

IP Services 特性集能够为 IPv4 支持 BGP NSF 感知（Awareness）特性。要想为 BGP 路由器用该特性，用户需要启用平滑重启（Graceful Restart）特性。当邻居路由器具有 NSF 功能（NSF-Capable）并且启用了该特性时，三层设备可以在以下事件发生时继续从邻居路由器转发数据包：路由器中的主用路由处理器（RP）失效且备用 RP 接管工作的时间段内，或者用户手动重启主用 RP 来实施无间断软件升级的过程中。

更多信息用户可以参考 *Inspur INOS IP Routing Protocols Configuration Guide, Release 12.4* 中的“BGP Nonstop Forwarding (NSF) Awareness”部分。

BGP 路由的相关信息

要想启用 BGP 路由，用户需要建立一个 BGP 路由进程并定义本地网络。由于 BGP 必须完全识别与邻居之间的关系，因此用户必须指定 BGP 邻居。

BGP 支持两种类型的邻居：内部和外部。内部邻居是指属于同一个 AS 中的邻居；外部邻居是指属于不同 AS 的邻居。外部邻居通常是彼此直连的并且连接在一个子网中，但内部邻居可以处于 AS 中的任何位置。

交换机支持使用私有 AS 号，通常由服务提供商分配给不会把自己的路由通告给外部邻居的系统。私有 AS 号的范围是 64512 至 65535。用户可以使用路由器配置命令 **neighbot remove-private-as**，配置外部邻居把私有 AS 号从 AS Path 中移除。这样当更新被发送到外部邻居时，如果 AS Path 中包含私有 AS 号的话，这些私有 AS 号会被丢弃。

如果用户的 AS 需要把另一个 AS 的流量传输给第三个 AS，一定要使它通告的路由保持统一。如果 BGP 通告了一条路由，但网络中的所有路由器还没有通过 IGP 学到这条路由，那么这个 AS 就会接收到一些它还无法路由的流量。要想防止发生这种情况，BGP 必须等待 IGP 已经把相关信息传播到了整个 AS 中，这样 BGP 才算是与 IGP 进行了同步。同步默认就是启用的。如果用户的 AS 不需要把一个 AS 的流量传输给另一个 AS，或者用户 AS 中的所有路由器都运行 BGP，那么用户可以禁用同步特性，这样可以使网络中承载更少的 IGP 路由，并且让 BGP 的收敛速度更快。

路由策略的变更

对一个对等体实施的路由策略中包括所有可能会影响入向或出向路由表更新的配置。当用户把两台路由器定义为 BGP 邻居后，它们之间会形成 BGP 连接并交换路由信息。如果用户稍后变更了 BGP 过滤器、权重、距离、版本或计时器，或者实施了类似的配置变更，就必须重置 BGP 会话才能使配置变更生效。

用户可以使用两种类型的重置：硬重置和软重置。Inspur INOS 12.1 及其后续版本无需预先配置就可以支持软重置。要想无需预先配置就使用软重置的话，两台 BGP 对等体必须都支持软路由刷新（Soft Route Refresh）功能，对等体之间建立 TCP 会话时，会通过发送 OPEN 消息来通告这个功能。软重置能够在 BGP 路由器之间动态交换路由刷新请求和路由信息，并随后重新通告相应的出向路由表。

- 当软重置特性从一个邻居生成入向更新时，称为动态入向软重置；
- 当软重置特性向一个邻居发送一组更新时，称为出向软重置。

入向软重置能够使新的入向策略生效。出向软重置可以使新的本地出向策略生效，而无需重置 BGP 会话。随着在出向策略重置期间发送新的一组更新，新的入向策略也可以生效。

下面这个表格中列出了硬重置和软重置的优势和劣势。

表 113：硬重置和软重置的优势和劣势

重置类型	**优势**	**劣势**
硬重置	无内存开销	邻居所提供的 BGP 中的前缀、IP 和 FIB 表都会丢失。不推荐

		使用
出向软重置	无配置，不储存路由表更新	不会重置入向路由表更新
动态入向软重置	不会清除 BGP 会话和缓存。不需要储存路由表更新并且没有内存开销	两台 BGP 路由器都必须支持路由刷新功能（Inspur INOS 12.1 及其后续版本）

BGP 决策属性

当 BGP 设备从多个自治系统接收到去往相同目的地的多条描述不同路径的路由更新时，它必须选择一条到达这个目的地的最优路径。在选择时，选中的路径会被放入 BGP 路由表中并被传播给它的邻居。决策是根据更新中包含的属性值和其他 BGP 配置因素作出的。

当 BGP 对等体从一个邻居 AS 学到了两条 EBGP 路径时，它会选择最优路径并将其放入 IP 路由表中。如果用户启用了 BGP 多路径支持特性，并且多条 EBGP 路径是从相同的邻居自治系统学到的，BGP 对等体就不会只选择一条最优路径，而是会把多条路径放入 IP 路由表中。接着在数据包交换期间，BGP 对等体会在多条路径上根据每个数据包或每个目的地执行负载均衡。用户可以使用路由器配置命令 **maximum-paths** 来控制允许同时使用的路径数量。

BGP 在为选择最优路径而评估属性值时，会按序考虑以下因素：

1. 如果路径中指定的下一跳不可访问，BGP 就会丢弃这个更新。软件会自动确定 BGP 下一跳属性，这是设备要用来到达目的地的下一跳 IP 地址。对于 EBGP 来说，BGP 下一跳属性通常是由配置命令 **neighbor remote-as router** 指定的邻居 IP 地址。用户可以通过使用 route-map 或路由器配置命令 **neighbor next-hop-self**，来禁止软件决定下一跳；
2. 使用最大权重来（Inspur 私有参数）选择路径。权重属性对于路由器来说是本地属性，并不会在路由更新中传播。默认情况下，路由器生成路径的权重属性值是 32768，其他路径的权重属性值是 0。优选拥有最大权重值的路由。用户可以使用访问列表、route-map 或路由器配置命令 **neighbor weight**，来设置权重值；
3. 优选拥有最高本地优先级的俄路由。本地优先级是路由更新中的一部分，会在同一个 AS 中的路由器之间交换。本地优先级属性的默认值是 100。用户可以使用路由器配置命令 **bgp default local-preference** 或 route-map 来设置本地优先级值；
4. 优选本地路由器的 BGP 生成的路由；
5. 优选 AS Path 最短的路由；
6. 优选起源类型最低的路由。内部路由或 IGP 的起源属性值低于从 EGP 学到的路由，从 EGP 学到的路由低于从未知起源或其他途径学到的路由；
7. 如果所有备选路由的邻居 AS 都相同的话，优选多出口鉴别器（MED）度量属性值最低

的路由。用户可以使用 route-map 或路由器配置命令 **default-metric**，来配置 MED 值。在 BGP 设备向 IBGP 对等体发送更新时，MED 值包含在内；

8. 优选外部（EBGP）路径，而不是内部（IBGP）路径；
9. 优选能够从最近的 IGP 邻居到达的路由（IGP 度量值最低）。这意味着路由器会优选 AS 内部最短的内部路径，来到达目的地（去往 BGP 下一跳最短的路径）；
10. 如果满足以下条件，就把这条路径的路由插入 IP 路由表中：
    最优路由和这条路由都是外部的；
    最优路由和这条路由都来自相同的邻居自治系统；
    启用了多路径特性。
11. 如果启用了多路径特性，优选 BGP 路由器 ID 的 IP 地址值最小的路由。路由器 ID 通常是路由器上最大的 IP 地址，或者环回（虚拟）地址，但用户也可以单独指定。

route-map

在 BGP 中，用户可以使用 route-map 来控制和修改该路由信息，并定义在路由域间重分发路由的条件。有关 route-map 的更多信息，用户可以参考“使用 route-map 重分发路由信息”部分。每个 route-map 都有一个用来标识自己的名称（*map tag*）和（可选的）序列号。

BGP 过滤器

用户可以使用 AS Path 过滤器来过滤 BGP 通告，比如全局配置命令 **as-path access-list** 和路由器配置命令 **neighbor filter-list**。用户也可以使用访问列表和路由器配置命令 **neighbor filter-list** 来过滤 BGP 通告。distribute-list 过滤器中需要应用网络号。有关 **distribute-list** 命令的更多信息，用户可以参考“控制路由更新中的通告和处理”部分。

用户可以针对每个邻居使用 route-map 来过滤更新并修改各种属性。route-map 可以应用在入向更新或出向更新上。只有 route-map 中允许的路由才会在更新中发送或接收。对于入向和出向更新来说，匹配的参数都是 AS Path、团体和网络号。用户需要使用 route-map 命令 **match as-path access-list** 来匹配自治系统路径，使用 route-map 命令 **match community-list** 来匹配团体，以及使用全局配置命令 **ip access-list** 来匹配网络号。

使用前缀列表进行 BGP 过滤

用户可以在众多 BGP 路由过滤命令中使用前缀列表来代替访问列表，其中包括路由器配置

命令 **neighbor distribute-list**。使用前缀列表的好处包括：提高了加载和查找大列表的效率、支持增量更新、CLI 配置更简单，以及提供了更强的灵活性。

要想使用前缀列表进行过滤，用户需要在前缀列表中匹配路由的前缀，就像在访问列表中的配置一样。当路由匹配列表中的条件时，BGP 就会使用这条路由。BGP 根据以下规则来允许或拒绝特定前缀：

- 空的前缀列表表示允许所有前缀；
- 隐含的拒绝语句用来匹配那些与前缀列表中所有条目都不匹配的前缀；
- 当前缀列表中的多个条目都与指定前缀相匹配时，与指定前缀相匹配的前缀列表中序列号最小的条目是这个前缀最终匹配的条目。

默认情况下，序列号是自动生成的，并以 5 为单位递增。如果用户禁止设备自动生成序列号，就必须在每个条目中手动指定序列号。用户可以以任意数值为单位递增序列号。如用用户以 1 为单位递增的话，以后就不能在某两个条目中间插入新条目了；如果用户使用非常大的增量，可能会很快耗尽所有序列号值。

BGP 团体过滤器

BGP 用来控制路由信息分发的一种方法是使用 COMMUNITIES（团体）属性值。这个属性总是用来把一组目的地放入团体中，并根据团体来应用路由决策。这种方法简化了 BGP 设备上的路由信息分发配置。

一个团体中的一组目的地拥有相同的属性。一个目的地可以属于多个团体。AS 管理员可以为目的地定义它属于哪些团体。默认情况下，所有目的地都属于通用 Internet 团体。团体是的取值范围是 1 至 4294967200，由 COMMUNITIES 属性、可选项、传输性、全局属性标识。以下这些是预定义的周知团体：

- **internet——**向 Internet 团体通告这条路由。所有路由器都属于这个团体；
- **no-export**——不向 EBGP 对等体通告这条路由；
- **no-advertise**——不向任何对等体（内部或外部）通告这条路由；
- **local-as——**不向本地自治系统之外的对等体通告这条路由。

根据团体，用户可以控制应该从其他邻居接受、优选，或向其他邻居通告哪些路由信息。一个 BGP 设备上可以设置、附加或修改它所学到、通告、或分发的路由的团体。当路由被聚合时，聚合后的结果也有一个 COMMUNITIES 属性，其中包含所有初始路由的所有团体。

用户可以使用团体列表来创建团体组，用来匹配 route-map 中的条目。在使用访问列表时，用户可以创建一系列团体列表。设备会逐一检查每条规则直到发现匹配项目为止。一旦满足了匹配条件，匹配过程就会结束。

要想设置 COMMUNITIES 属性并基于属性进行规则匹配的话，用户可以使用 route-map 配置命令 **match community-list** 和 **set community**，用户可以参考“使用 route-map 分发路由信息”部分。

BGP 邻居和对等体组

通常用户会为多个 BGP 邻居配置相同的更新策略（也就是说相同的出向 route-map、distribute-map、filter-list、更新源等）。拥有相同更新策略的邻居可以被划分为一个对等体组中，这种做法可以简化配置并提高更新的效率。当用户需要配置多个对等体时，建议使用这种方法。

要想配置一个 BGP 对等体组，用户可以先创建对等体组（peer-group），然后为对等体组分配选项，并把邻居添加为对等体组成员。用户可以使用路由器配置命令来配置 **neighbor** 对等体组。默认情况下，对等体组成员会继承对等体组中的所有配置选项，其中包括远端 AS（remote-as；如果配置了的话）、版本、更新源、出向 route-map、出向 filter-list、出向 distribute-list、最小通告间隔和 next-hop-self。所有对等体组成员还会继承用户对这个对等体组做出的变更。用户可以单独配置成员来覆盖不会影响出向更新的选项。

聚合路由

无类域间路由（CIDR）使用户可以创建聚合路由（或超网），来减小路由表的大小。用户可以通过把聚合路由重分发到 BGP 中，或者通过在 BGP 路由表中创建一个聚合条目，来在 BGP 中配置聚合路由。对于一个聚合地址来说，只要 BGP 表中至少有一个明细路由条目，这个聚合路由就会被添加到 BGP 表中。

路由域联盟

其中一个能够减少 IBGP 全互联的做法是把一个自治系统分为多个子自治系统，并把它们汇集为一个联盟，对外展示为一个自治系统。每个子自治系统之内是全互联的，并且与联盟内的其他自治系统之间有一些连接。即使不同自治系统中的对等体之间会建立 EBGP 会话，但它们会像 IBGP 对等体那样交换路由信息。尤其是下一跳、MED 和本地优先级信息都会保留。用户可以所有自治系统使用一个 IGP。

BGP 路由反射器

BGP 要求所有 IBGP 设备之间是全互联的。当路由器从一个外部邻居那里接收到一条路由时，它必须把这条路由通告给所有内部邻居。为了防止出现路由信息环路，所有 IBGP 设备之间必须全互联。内部邻居不会把从一个内部邻居那里学到的路由发送给其他内部邻居。

通过使用路由反射器，所有 IBGP 设备之间无需全互联，因为这时可以使用另一种方法来向邻居传递学到的路由。在用户把一个内部 BGP 设备配置为路由反射器后，它就负责向一组 IBGP 邻居传递从 IBGP 学到的路由。路由反射器的内部对等体分为以下两组：客户端对等体和非客户端对等体（自治系统中的所有其他路由器）。路由反射器会在这两个组之间进行路由反射。路由反射器和它的客户端对等体形成了一个集群（Cluster）。非客户端对等体之间必须全互联，但客户端对等体之间不必全互联。集群中的客户端不会和集群外的 IBGP 设备进行通信。

但路由反射器接收和通告路由时，它会根据邻居的不同做出以下行为：

- 把从外部 BGP 设备学到的路由通告给所有客户端对等体和非客户端对等体；
- 把从非客户端对等体学到的路由通告给所有客户端对等体；
- 把从客户端对等体学到的路由通告给所有客户端对等体和非客户端对等体。因此客户端之间不用全互联。

通常一个集群中的客户端都只有一个路由反射器，并且这个集群是由路由反射器的路由器 ID 来标识的。为了增加冗余并避免单点故障，一个集群中可以有多个路由反射器。在这种情况中，一个集群中的所有路由反射器必须配置相同的 4 字节集群 ID，这样一来，路由反射器就能够识别出同一个集群中的其他路由反射器通告的更新了。所有服务于同一个集群的路由反射器之间应该全互联，并且有相同的客户端对等体和非客户端对等体。

路由阻尼（Dampening）

路由翻动阻尼是一向 BGP 特性，用来减少互联网络中翻动路由的传播。当一条路由反复变得可用和不可用时，就认为这条路由正在翻动。当用户启用了路由阻尼特性后，当路由发生翻动时，会有一个惩罚值应用到这条路由上。当这条路由的惩罚值积累到一个用户配置的限制标准时，BGP 就会不再通告这条路由，即使路由是可用的。用户可以针对惩罚行为，配置重用限制。如果惩罚值小于重用限制的话，这条被抑制的路由就会再次被通告出去。

路由阻尼不会应用在通过 IBGP 学到的路由上。这个策略会防止 IBGP 对等体对 AS 外部的路由应用更高的惩罚。

更多 BGP 信息

有关 BGP 配置的详细信息，用户可以参考 *Inspur INOS IP Configuration Guide, Release 12.4* 的“IP Routing Protocols”部分。有关具体命令的详细信息，用户可以参考 *Inspur INOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.4*。

如何配置 BGP

默认的 BGP 配置

下面这个表格中展示了基本的默认 BGP 配置。有关所有特征的默认设置，用户可以参考 *Inspur INOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.4*。

表 114：默认的 BGP 配置

特性	**默认设置**
聚合地址	禁用：未定义
AS Path 访问列表	未定义
自动汇总	禁用
最优路径	• 路由器在选择路由时会考虑 *as-path*，并且不会与外部 BGP 对等体对比类似的路由 • 对比路由器 ID：禁用
BGP 团体列表	• 编号：未定义。当用户设置了一个团体编号值时，这个列表默认会隐含拒绝所有用户没有明确允许的路由 • 格式：Inspur 默认格式（32 比特编号）
BGP 联盟 识别符/对等体	• 识别符：未配置 • 对等体：未指定
BGP 快速外部切换	启用
BGP 本地优先级	100。取值范围是 0 至 4294967295，优选较高数值
BGP 网络	未指定；不通告后门路由
BGP 路由阻尼	默认为禁用状态。当启用时： • 半衰期是 15 分钟 • 重新启用限制值是 750（增量为 10 秒）

	• 抑制限制值 2000（增量为 10 秒） • 最大抑制时间是半衰期的 4 倍；60 分钟
BGP 路由器 ID	如果用户配置了环回接口，就使用环回接口的 IP 地址；或者使用路由器上物理接口中配置的最大 IP 地址
生成默认信息（协议或网络重分发）	禁用
默认度量值	内建，自动度量值转换。
距离	• 外部路由管理距离：20（可用值范围是 1 至 255） • 内部路由管理距离：200（可用值范围是 1 至 255） • 本地路由管理距离：200（可用值范围是 1 至 255）
分发列表	• 入向（过滤更新中接收到的网络）：禁用 • 出向（抑制在更新中通告某些网络）：禁用
内部路由重分发	禁用
IP 前缀列表	未定义
多出口鉴别器（MED）	• 总是对比：禁用。对于从不同自治系统中邻居学到的路径，不对比 MED • 最优路径对比：禁用 • 缺少 MED 就是最差路径：禁用 • 禁用确定性 MED 对比
邻居	• 通告间隔：外部对等体 30 秒钟；内部对等体 5 秒钟 • 变更日志记录：启用 • 条件通告：禁用 • 生成默认：不向邻居发送默认路由 • 描述：无 • 分发列表：未定义 • 外部 BGP 多跳：只允许直连邻居 • 过滤列表：未使用 • 接收到的最大前缀数量：无限制 • 下一跳（路由器作为 BGP 邻居的下一跳）：禁用 • 密码：禁用 • 对等体组：未定义；未分配成员 • 前缀列表：未定义

	• 远端 AS（向邻居 BGP 表中添加条目）：未定义对等体 • 移除私有 AS 编号：禁用 • route-map：未向对等体应用 • 发送团体属性：未向邻居发送团体属性 • 关闭或软重配置：未启用 • 计时器：存活时间：60 秒钟；保持时间：180 秒钟 • 更新源：最优本地地址； • 版本：BGP 版本 4 • 加权：从 BGP 对等体学到的路由：0；本地路由器生成的路由：32768
NSF[6] 感知	禁用 [7]。启用后，三层交换机在硬件或软件变更期间仍能够继续从具有 NSF 功能的邻居那里转发数据包
路由反射器	未配置
同步（BGP 和 IGP）	禁用
table-map 更新	禁用
计时器	存活时间：60 秒钟；保持时间：180 秒钟

[6] 无间断转发

[7] 用户可以在运行 IP Services 特性集的交换机上通过启用平滑重启（Graceful Restart），为 IPv4 启用 NSF 感知功能。

启用 BGP 路由

在开始前

注释： 要想启用 BGP，交换机或堆栈主用设备上必须运行 IP Services 特性集。

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**ip routing**	启用 IP 路由功能

	示例： `Device(config)# ip routing`	
步骤 3	**router bgp** *autonomous-system* **示例：** `Device(config)# router bgp 45000`	启用 BGP 路由进程、指定 AS 号，并进入路由器配置模式。AS 号的取值范围是 1 至 65535，其中 64512 至 65535 是私有自治系统编号
步骤 4	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] **示例：** `Device(config)# network 10.108.0.0`	配置这个 AS 本地的网络，并把它放入 BGP 表中
步骤 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number* **示例：** `Device(config)# neighbor 10.108.1.2 remote-as 65200`	在 BGP 邻居表中添加一个条目，用 IP 地址识别邻居，并指明邻居所属的 AS。对于 EBGP 来说，邻居通常是直连的，IP 地址就是连接对端接口的 IP 地址。对于 IBGP 来说，IP 地址可以是路由器上任意接口的 IP 地址
步骤 6	**neighbor** {*ip-address* | *peer-group-name*} **remove-private-as** **示例：** `Device(config)# neighbor 172.16.2.33 remove-private-as`	（可选）在出向路由更新的 AS Path 中移除私有 AS 号
步骤 7	**synchronization** **示例：** `Device(config)# synchronization`	（可选）启用 BGP 与 IGP 之间的同步

步骤 8	**auto-summary** **示例：** `Device(config)# auto-summary`	（可选）启用自动网络汇总特性。当从 IGP 向 BGP 中重分发一个子网时，只有网络路由会被放入到 BGP 表中
步骤 9	**bgp graceful-restart** **示例：** `Device(config)# bgp graceful-start`	（可选）在交换机上启用 NSF 感知特性。默认情况下，NSF 感知特性是禁用的
步骤 10	**end** **示例：** `Device(config-if)# end`	退出接口配置模式并返回特权 EXEC 模式
步骤 11	**show ip bgp network** *network-number* **示例：** `Device# show ip bgp network 10.108.0.0`	检查用户输入的信息
步骤 12	**show ip bgp neighbor** **示例：** `Device# show ip bgp neighbor`	确认邻居上是否启用了 NSF 感知特性（平滑重启）。 如果交换机和邻居都启用了 NSF 感知特性，该命令的输出内容中会显示如下信息： *Graceful Restart Capability: advertised and received* 如果交换机上启用了 NSF 感知特性，但邻居上没有启用，消息如下所示： *Graceful Restart Capability: advertised*
步骤 13	**copy running-config startup-config** **示例：** `Device# copy running-config`	（可选）把输入的命令保存到配置文件中

	startup-config	

管理路由策略的变更

用户可以使用以下命令来确认 BGP 对等体是否支持路由刷新特性，并重置 BGP 会话：

具体步骤

	命令或操作	目的
步骤 1	**show ip bgp neighbors** **示例：** `Device# show ip bgp neighbors`	显示邻居是否支持路由刷新功能。当邻居支持路由刷新功能时，路由器上会显示如下信息： *Received route refresh capability from peer.*
步骤 2	**clear ip bgp** {***** | *address* | *peer-group-name*} **示例：** `Device# clear ip bgp *`	重置指定连接上的路由表： • 输入星号（*）来重置所有连接 • 输入 IP 地址来指定需要重置的连接 • 输入对等体组名称来重置对等体组
步骤 3	**clear ip bgp** {***** | *address* | *peer-group-name*} **soft out** **示例：** `Device# clear ip bgp * soft out`	（可选）执行出向软重置，来重置指定连接上的入向路由表。
步骤 4	**show ip bgp** **示例：** `Device# show ip bgp`	通过检查有关路由表和 BGP 邻居的信息，来确认重置结果
步骤 5	**show ip bgp neighbors** **示例：** `Device# show ip bgp neighbors`	通过检查有关路由表和 BGP 邻居的信息，来确认重置结果

配置 BGP 决策属性

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**router bgp** *autonomous-system* **示例：** `Device(config)# router bgp 45000`	启用 BGP 路由进程、指定 AS 号，并进入路由器配置模式
步骤 3	**bgp best-path as-path ignore** **示例：** `Device(config-router)# bgp bestpath as-path ignore`	（可选）配置路由器在选择路由时忽略 AS Path 长度
步骤 4	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self** **示例：** `Device(config-router)# neighbor 10.108.1.1 next-hop-self`	（可选）对一个邻居禁止在 BGP 更新中生成下一跳，指定一个 IP 地址来代替下一跳地址
步骤 5	**neighbor** {*ip-address* | *peer-group-name*} **weight** *weight* **示例：** `Device(config-router)# neighbor 172.16.12.1 weight 50`	（可选）为一个邻居连接分配一个权重。取值范围是 0 至 65535；优先使用权重最大的路由。从另一个 BGP 对等体学到的路由默认权重为 0；本地路由器为起源的路由默认权重为 32768

步骤 6	**default-metric** *number* **示例：** `Device(config-router)# default-metric 300`	（可选）设置 MED 度量值，让外部邻居优选路径。所有未设置 MED 值的路由都会被设置为这个值。取值范围是 1 至 4294967295。最低值是最优选路由
步骤 7	**bgp bestpath med missing-as-worst** **示例：** `Device(config-router)# bgp bestpath med missing-as-worst`	（可选）配置交换机把缺少 MED 值的路由当作拥有无限大的值，使缺少 MED 值的路径称为最差路径
步骤 8	**bgp always-compare med** **示例：** `Device(config-router)# bgp always-compare-med`	（可选）配置交换机，让它对从不同自治系统中的邻居那里学到的路径比较 MED 值。默认情况下，交换机只会对从相同 AS 中学到的路径比较 MED 值
步骤 9	**bgp bestpath med confed** **示例：** `Device(config-router)# bgp bestpath med confed`	（可选）配置交换机，让它对同一个联盟中不同子自治系统中通告的路径对比 MED 值
步骤 10	**bgp deterministic med** **示例：** `Device(config-router)# bgp deterministic med`	（可选）配置交换机，让它对同一个 AS 种不同对等体通告的路由对比 MED 值
步骤 11	**bgp default local-preference** *value* **示例：** `Device(config-router)# bgp default local-preference 200`	（可选）配置默认的本地优先级值。取值范围是 1 至 4294967295；默认值为 100。最高的本地优先级值是最优的
步骤 12	**maximum-paths** *number*	（可选）配置交换机能够向 IP 路由表汇总添加的路径数量。默认只有最优路

	示例： Device(config-router)# maximum-paths 8	径会被放入路由表中。取值范围是 1 至 16。使用多条路径能够在多条路径上实现负载均衡(尽管交换机软件允许最多使用 32 条等价路由，但交换机硬件不允许为一条路由使用超过 16 条路径）
步骤 13	**end** **示例：** Device(config-if)# **end**	退出接口配置模式并返回特权 EXEC 模式
步骤 14	**show ip bgp** **示例：** Device# show ip bgp	通过检查有关路由表和 BGP 邻居的信息，来确认重置结果
步骤 15	**show ip bgp neighbors** **示例：** Device# show ip bgp neighbors	通过检查有关路由表和 BGP 邻居的信息，来确认重置结果
步骤 16	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选)把输入的命令保存到配置文件中

使用 route-map 配置 BGP 过滤

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式

步骤 2	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **示例：** `Device(config)# route-map set-peer-address permit 10 Device(config)# route-map set-peer-address permit 10`	创建一个 route-map，并进入 route-map 配置模式
步骤 3	**set ip next-hop** *ip-address* [*...ip-address*] [**peer-address**] **示例：** `Device(config)# set ip next-hop 10.1.1.3`	（可选）设置一个 route-map 来禁用下一跳特性 • 在入向 route-map 中，用户可以把匹配路由的下一跳设置为邻居对等体地址，覆盖第三方下一跳 • 在一个 BGP 对等体的出向 route-map 中，用户可以把下一跳设置为本地路由器的对等体地址，禁用下一跳计算
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show route-map** [*map-name*] **示例：** `Device# show route-map`	显示用户配置的所有 route-map，或者只显示指定的一个 route-map
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

为邻居配置 BGP 过滤

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**router bgp** *autonomous-system* **示例：** `Device(config)# router bgp 45000`	启用 BGP 路由进程、指定 AS 号，并进入路由器配置模式
步骤 3	**neighbor** {*ip-address* | *peer-group name*} **distribute-list** {**access-list-number** | *name*} {**in** | **out**} **示例：** `Device(config-router)# neighbor 172.16.4.1 distribute-list 39 in`	（可选）根据访问列表中的定义，过滤去往或来自邻居的 BGP 路由更新。 **注释：** 用户也可以使用路由器配置命令 **neighbor prefix-list** 来过滤更新，但用户不能针对同一个 BGP 对等体同时配置这两条命令
步骤 4	**neighbor** {*ip-address* | *peer-group name*} **route-map** *map-tag* {**in** | **out**} **示例：** `Device(config-router)# neighbor 172.16.70.24 route-map internal-map in`	（可选）应用一个 route-map 来过滤入站或出站路由
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式

步骤 6	**show ip bgp neighbors** **示例：** `Device# show ip bgp neighbors`	通过检查有关路由表和 BGP 邻居的信息，来确认重置结果
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

通过访问列表和邻居配置 BGP 过滤

另一种过滤方式是指定一个访问列表，基于 BGP 自治系统路径，来同时过滤入站和出站更新。每个过滤条件都是依赖于正则表达式描述的访问列表（有关正则表达式的更多信息，用户可以参考 *Inspur INOS Dial Technologies Command Reference, Release 12.4* 中的“Regular ExpressionsRegular Expressions”附录）。要想使用这个方法，用户需要定义一个自治系统路径访问列表，并把它应用在指定邻居的指定方向上。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**ip as-path access-list** *access-list-number* {**permit** | **deny**} *as-regular-expressions* **示例：** `Device(config)# ip as-path access-list 1 deny _65535_`	定义一个与 BGP 相关的访问列表
步骤 3	**router bgp** *autonomous-system* **示例：**	进入 BGP 路由器配置模式

	Device(config)# router bgp 110	
步骤 4	**neighbor** {*ip-address* | *peer-group name*} **filter-list** {*access-list-number* | *name*} {**in** | **out** | **weight** *weight*} **示例：** Device(config-router)# neighbor 172.16.1.1 filter-list 1 out	基于访问列表建立 BGP 过滤器
步骤 5	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 6	**show ip bgp neighbors** [**paths** *regular-expression*] **示例：** Device# show ip bgp neighbors	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

为 BGP 过滤特性配置 prefix-list

用户在移除配置条目时无需指定序列号。**show** 命令的显示信息中会包含序列号。

要想在命令中使用 prefix-list，用户必须首先建立一个 prefix-list。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal**	进入全局配置模式

	示例： `Device# configure terminal`	
步骤 2	**ip prefix-list** *list-name* [**seq** *seq-value*] **deny** | **permit** *network/len* [**ge** *ge-value*] [**le** *le-value*] **示例：** `Device(config)# ip prefix-list BLUE permit 172.16.1.0/24`	创建一个 prefix-list，（可选的）设置序列号，为匹配条件指定 **deny** 或 **permit** 行为。用户必须至少输入一条 **permit** 或 **deny** 命令。 • 在 *network/len* 部分输入网络号和网络掩码的长度（以比特为单位） • （可选）使用 **ge** 和 **le** 值来指定匹配的前缀长度范围。用户指定的 *ge-value* 和 *le-value* 必须满足以下条件：*len* < *ge-value* < *le-value* < 32
步骤 3	**ip prefix-list** *list-name* **seq** *seq-value* **deny** | **permit** *network/len* [**ge** *ge-value*] [**le** *le-value*] **示例：** `Device(config)# ip prefix-list BLUE seq 10 permit 172.24.1.0/24`	（可选）在 prefix-list 中添加一个条目，并为这个条目分配一个序列号
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show ip prefix list** [**detail** | **summary**] *name* [*network/len*] [**seq** *seq-num*] [**longer**] [**first-match**] **示例：** `Device# show ip prefix list summary test`	通过显示有关 prefix-list 及其条目来检查用户输入的信息

步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 BGP 团体过滤

默认情况下，设备不会把 COMMUNITIES 属性发送给邻居。用户可以使用路由器配置命令 **neighbor send-community**，指定要发送给邻居（IP 地址）的 COMMUNITIES 属性。

总步骤

1. configure terminal

2. ip community-list *community-list-number* {**permit** | **deny**} *community-number*

3. router bgp *autonomous-system*

4. neighbor {*ip-address* | *peer-group name*} **send-community**

5. set comm-list *list-num* **delete**

6. exit

7. ip bgp-community new-format

8. end

9. show ip bgp community1. configure terminal

10. copy running-config startup-config

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**ip community-list** *community-list-number* {**permit** | **deny**} *community-number* **示例：** `Device(config)# ip`	创建一个 community-list，并为它分配一个编号。 • 在 *community-list-number* 部分输入 1 至 99 之间的整数，指定一个或多个行为为允许或拒绝的团体组

	`community-list 1 permit 50000:10`	• 在 *community-number* 部分配置的编号是用户在 route-map 配置命令 **set community** 中设置的值
步骤 3	**router bgp** *autonomous-system* **示例：** `Device(config)# router bgp 108`	进入 BGP 路由器配置模式
步骤 4	**neighbor** {*ip-address* | *peer-group name*} **send-community** **示例：** `Device(config-router)# neighbor 172.16.70.23 send-community`	把指定 COMMUNITIES 属性发送给这个 IP 地址上的邻居
步骤 5	**set comm-list** *list-num* **delete** **示例：** `Device(config-router)# set comm-list 500 delete`	（可选）从入向或出向更新的团体属性中移除团体，移除的俄团体需要匹配 route-map 中指定的标准或扩展 community-list
步骤 6	**exit** **示例：** `Device(config-if)# exit`	返回全局配置模式
步骤 7	**ip bgp-community new-format** **示例：** `Device(config)# ip bgp-community new format`	（可选）以格式 AA:NN 显示并解析 BGP 团体。 一个 BGP 团体会显示为一个分为两部分的格式，长度为 2 字节。Inspur 默认的团体格式是 NNAA 格式。在大多数当前定义 BGP 的 RFC 文档中，团体使用 AA:NN 格式，其中第一部分是 AS 号，第二部分是 2 字节编号

步骤 8	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 9	**show ip bgp community** **示例：** `Device# show ip bgp community`	检查用户输入的信息
步骤 10	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 BGP 邻居和对等体组

要想为指定邻居分配配置选项，用户可以使用邻居的 IP 地址来指定这些路由器配置命令。要想为一个对等体组分配配置选项，用户可以使用对等体组名称来指定这些命令。用户可以使用路由器配置命令 **neighbor shutdown**，来禁用 BGP 对等体或对等体组，同时不移除所有配置信息。

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**router bgp** *autonomous-system*	进入 BGP 路由器配置模式
步骤 3	**neighbor** *peer-group-name* **peer-group**	创建一个 BGP 对等体组
步骤 4	**neighbor** *ip-address* **peer-group** *peer-group-name*	使一个 BGP 邻居成为对等体组的成员
步骤 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number*	指定一个 BGP 邻居。如果对等体组中没有配置 **remote-as** *number*，用户可以

		使用这条命令来创建包含 EBGP 邻居的对等体组。取值范围是 1 至 65535
步骤 6	**neighbor** {*ip-address* | *peer-group-name*} **description** *text*	（可选）为邻居关联一个描述信息
步骤 7	**neighbor** {*ip-address* | *peer-group-name*} **default-originate** [**route-map** *map-name*]	（可选）允许 BGP 设备（本地路由器）向邻居发送默认路由 0.0.0.0，作为默认路由使用
步骤 8	**neighbor** {*ip-address* | *peer-group-name*} **send-community**	（可选）指定向邻居 IP 地址发送的 COMMUNITIES 属性
步骤 9	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface*	（可选）允许内部 BGP 会话使用任何工作接口建立 TCP 连接
步骤 10	**neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop**	（可选）允许不在直连网段上的邻居之间建立 BGP 会话。如果去往多跳对等体地址的唯一一条路由是默认路由（0.0.0.0）的话，类设备就不会与这个对等体建立多跳会话
步骤 11	**neighbor** {*ip-address* | *peer-group-name*} **local-as** *number*	（可选）指定作为本地 AS 使用的 AS 号。取值范围是 1 至 65535
步骤 12	**neighbor** {*ip-address* | *peer-group-name*} **advertisement-interval** *seconds*	（可选）设置发送 BGP 路由更新的最小间隔
步骤 13	**neighbor** {*ip-address* | *peer-group-name*} **maximum-prefix** *maximum* [*threshold*]	（可选）控制能够从一个邻居那里接收多少个前缀。取值范围是 1 至 4294967295。（可选）在 *threshold* 部分指定设备生成警告消息的最大百分比。默认值是 75%
步骤 14	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**	（可选）在向一个邻居发送的 BGP 更新中禁用下一跳计算
步骤 15	neighbor {*ip-address* | *peer-group-name*} password *string*	（可选）在与一个 BGP 对等体建立的 TCP 会话上设置 MD5 认证。两个 BGP 对等体上必须配置相同的密码，否则它们之间的会话无法建立
步骤 16	**neighbor** {*ip-address* | *peer-group-name*}	（可选）在入站或出站路由上应用一个

	route-map *map-name* {**in** | **out**}	route-map
步骤 17	**neighbor** {*ip-address* | *peer-group-name*} **send-community**	（可选）设置要发送给邻居 IP 地址的 COMMUNITIES 属性
步骤 18	**neighbor** {*ip-address* | *peer-group-name*} **timers** *keepalive holdtime*	（可选）为邻居或对等体组设置计时器。 • *keepalive* 间隔是向对等体发送存活消息的时间间隔。取值范围是 1 至 4294967295 秒；默认值为 60 秒钟 • *holdtime* 是指在多长时间之后仍没有从对等体那里收到存活消息，就认为这个对等体失效。取值范围是 1 至 4294967295 秒；默认值为 180 秒钟
步骤 19	**neighbor** {*ip-address* | *peer-group-name*} **weight** *weight*	（可选）为从一个邻居学到的所有路由指定权重
步骤 20	**neighbor** {*ip-address* | *peer-group-name*} **distribute-list** {*access-list-number* | *name*} {**in** | **out**}	（可选）过滤发往邻居或从邻居接收到的 BGP 路由更新，使用访问列表进行匹配
步骤 21	**neighbor** {*ip-address* | *peer-group-name*} **filter-list** *access-list-number* {**in** | **out** | **weight** *weight*}	（可选）建立一个 BGP 过滤器
步骤 22	**neighbor** {*ip-address* | *peer-group-name*} **version** *value*	（可选）指定与一个邻居通信时使用的 BGP 版本
步骤 23	**neighbor** {*ip-address* | *peer-group-name*} **soft-reconfiguration inbound**	（可选）配置让软件开始储存接收到的更新
步骤 24	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 25	**show ip bgp neighbors**	检查用户输入的信息
步骤 26	**copy running-config startup-config**	（可选）把输入的命令保存到配置文件

	示例： `Device# copy running-config startup-config`	中

在路由表中配置聚合地址

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**router bgp** *autonomous-system* **示例：** `Device(config)# router bgp 106`	启用 BGP 路由进程、指定 AS 号，并进入路由器配置模式
步骤 3	**aggregate-address** *address mask* **示例：** `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0`	在 BGP 路由表中创建一个聚合条目。聚合路由会被通告为来自这个 AS，并且会设置原子聚合属性，来表示这个信息可能会丢失
步骤 4	**aggregate-address** *address mask* **as-set** **示例：** `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 as-set`	（可选）生成 AS-Set 路径信息。这条命令按照与前一条命令相同的规则，创建出一个聚合条目，但通告的路径会是包含了所有路径元素的 AS_SET。用户不要再聚合多条路径时使用这个关键字，因为这条路由必须会持续撤销和更新
步骤 5	**aggregate-address** *address-mask*	（可选）只通告汇总地址

	summary-only **示例：** `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 summary-only`	
步骤 6	**aggregate-address** *address mask* **suppress-map** *map-name* **示例：** `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 suppress-map map1`	（可选）抑制用户选中的指定路由或更多路由
步骤 7	**aggregate-address** *address mask* **advertise-map** *map-name* **示例：** `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 advertise-map map2`	（可选）根据 route-map 中指定的条件生成一个聚合路由
步骤 8	**aggregate-address** *address mask* **attribute-map** *map-name* **示例：** `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 attribute-map map3`	（可选）使用 route-map 中指定的属性生成一个聚合路由
步骤 9	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 10	**show ip bgp neighbors**	验证用户输入的配置

	[**advertised-routes**] **示例：** `Device# show ip bgp neighbors`	
步骤 11	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置路由域联盟

用户必须指定联盟识别符，它的作用与为自治系统组设置自治系统号一样。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**router bgp** *autonomous-system* **示例：** `Device(config)# router bgp 100`	进入 BGP 路由器配置模式
步骤 3	**bgp confederation identifier** *autonomous-system* **示例：** `Device(config)# bgp confederation identifier 50007`	配置 BGP 联盟识别符
步骤 4	**bgp confederation peers**	指定属于这个联盟的自治系统，这些会

	autonomous-system [*autonomous-system* ...] **示例：** `Device(config)# bgp confederation peers 51000 51001 51002`	被当作特殊 EBGP 对等体
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show ip bgp neighbor** **示例：** `Device# show ip bgp neighbor`	检查用户输入的信息
步骤 7	**show ip bgp network** **示例：** `Device# show ip bgp network`	检查用户输入的信息
步骤 8	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 BGP 路由反射器

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：**	进入全局配置模式

	Device# **configure terminal**	
步骤 2	**router bgp** *autonomous-system* **示例：** Device(config)# router bgp 101	进入 BGP 路由器配置模式
步骤 3	**neighbor** {*ip-address* | *peer-group-name*} **route-reflector-client** **示例：** Device(config-router)# neighbor 172.16.70.24 route-reflector-client	把本地路由器配置为 BGP 路由反射器并把一个邻居指定为客户端
步骤 4	**bgp cluster-id** *cluster-id* **示例：** Device(config-router)# bgp cluster-id 10.0.1.2	（可选）如果集群中有多个路由反射器的话，配置集群 ID
步骤 4	**no bgp client-to-client reflection** **示例：** Device(config-router)# no bgp client-to-client reflection	（可选）禁用客户端到客户端的路由反射器。默认情况下，从路由反射器发来的路由会被反射给其他客户端。但如果客户端之间是全互联的，路由反射器就无须把路由反射给客户端了
步骤 6	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 7	**show ip bgp** **示例：** Device# show ip bgp	验证用户输入的配置。显示起源 IDhe 集群列表属性
步骤 8	**copy running-config startup-config**	（可选）把输入的命令保存到配置文件

	示例： `Device# copy running-config startup-config`	中

配置路由阻尼（Dampening）

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**router bgp** *autonomous-system* **示例：** `Device(config)# router bgp 100`	进入 BGP 路由器配置模式
步骤 3	**bgp dampening** **示例：** `Device(config-router)# bgp dampening`	启用 BGP 路由阻尼特性
步骤 4	**bgp dampening** *half-life reuse suppress max-suppress* [**route-map** *map*] **示例：** `Device(config-router)# bgp dampening 30 1500 10000 120`	（可选）变更路由阻尼因子的默认值
步骤 5	**end** **示例：**	返回特权 EXEC 模式

	Device(config)# **end**	
步骤 6	**show ip bgp flap-statistics** [{**regexp** *regexp*} | {**filter-list** *list*} | {*address mask* [**longer-prefix**]}] **示例：** Device# show ip bgp flap-statistics	（可选）监控所有路径的翻动情况。当路由不再被抑制且稳定后，状态统计信息就会被删除
步骤 7	**show ip bgp dampened-paths** **示例：** Device# show pi bgp dampened-paths	（可选）显示阻尼路由，其中包括路由被抑制前的保留时间
步骤 8	**clear ip bgp flap-statistics** [{**regexp** *regexp*} | {**filter-list** *list*} | {*address mask* [**longer-prefix**]} **示例：** Device# clear ip bgp flap-statistics	（可选）清除 BGP 翻动状态统计信息，使它看起来不太像是会被惩罚
步骤 9	**clear ip bgp dampening** **示例：** Device# clear ip bgp dampening	（可选）清除路由阻尼信息，取消抑制路由的抑制状态
步骤 10	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

监控和维护 BGP

用户可以移除指定缓存、表或数据库中的全部内容。当指定结构中的内容不可用，或似乎变得不可用时，用户可能就需要移除这些内容。

用户可以查看特定的统计状态信息，比如 BGP 路由表、缓存和数据库中的内容。用户可以使用这些信息来获得资源利用率并解决网络问题。用户还可以查看节点可达性信息，以及用户设备的数据包可能会使用哪条路径穿越网络的信息。

下面这个表格中列出了一些特权 EXEC 命令，用户可以使用这些命令来清除或展示 BGP 信息。有关显示字段的解释信息，用户可以参考 *Inspur INOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.4*。

表 115：IP BGP clear 和 show 命令

命令	目标
clear ip bgp *address*	重置指定的 BGP 连接
clear ip bgp *	重置所有 BGP 连接
clear ip bgp peer-group *tag*	移除一个 BGP 对等体组中的所有成员
show ip bgp *prefix*	显示通告了指定前缀的对等体组，以及不属于对等体组的对等体。这条命令也会显示诸如下一跳和本地优先级等前缀属性
show ip bgp cidr-only	显示包含子网和超网网络掩码的所有 BGP 路由
show ip bgp community [*community-number*] [**exact**]	显示属于指定团体的路由
show ip bgp community-list *community-list-number* [**exact-match**]	显示团体列表中允许的路由
show ip bgp filter-list *access-list-number*	显示指定 AS Path 访问列表中匹配的路由
show ip bgp inconsistent-as	显示携带不一致起源自治系统的路由
show ip bgp regexp *regular-expression*	显示匹配 AS Path 的路由，AS Path 的匹配规则由正则表达式描述
show ip bgp	显示 BGP 路由表的内容
show ip bgp neighbors [*address*]	显示某个邻居的 BGP 和 TCP 连接的详细信息
show ip bgp neighbors [*address*]	显示从指定 BGP 邻居学到的路由

[**advertised-routes** | **dampened-routes** | **flap-statistics** | **paths** *regular-expression* | **received-routes** | **routes**]	
show ip bgp paths	显示数据库中的所有 BGP 路径
show ip bgp peer-group [*tag*] [**summary**]	显示有关 BGP 对等体组的信息
show ip bgp summary	显示所有 BGP 连接的状态

bgp log-neighbor changes 命令是默认就启用的。它能够在 BGP 邻居重置、上线或下线时生成日志消息。

BGP 的配置示例

示例：在路由器上配置 BGP

这个示例展示了如何在下图所示的路由器上配置 BGP。

图 95：EBGP、IBGP 和多自治系统

Router A	路由器 A
Router D	路由器 D
Router B	路由器 B
Router C	路由器 C

路由器 A：

```
Device(config)# router bgp 100
Device(config-router)# neighbor 129.213.1.1 remote-as 200
```

路由器 B：

```
Device(config)# router bgp 200
Device(config-router)# neighbor 129.213.1.2 remote-as 100
Device(config-router)# neighbor 175.220.1.2 remote-as 200
```

路由器 C：

```
Device(config)# router bgp 200
Device(config-router)# neighbor 175.220.212.1 remote-as 200
Device(config-router)# neighbor 192.208.10.1 remote-as 300
```

路由器 D：

```
Device(config)# router bgp 300
Device(config-router)# neighbor 192.208.10.2 remote-as 200
```

要想验证 BGP 对等体是否正在运行，用户可以使用特权 EXEC 命令 show ip bgp neighbors 来进行确认。以下为路由器 A 上的这条命令输出示例：

```
Device# show ip bgp neighbors
BGP neighbor is 129.213.1.1, remote AS 200, external link
BGP version 4, remote router ID 175.220.212.1
BGP state = established, table version = 3, up for 0:10:59
Last read 0:00:29, hold time is 180, keepalive interval is 60 seconds
Minimum time between advertisement runs is 30 seconds
Received 2828 messages, 0 notifications, 0 in queue
Sent 2826 messages, 0 notifications, 0 in queue
Connections established 11; dropped 10
```

除了 *state = established* 之外的所有状态都表示对等体没有运行。远端路由器 ID 是远端路由器上的最大 IP 地址（或最大环回接口地址）。每次表中使用新信息进行更新时，表的版本号都会递增。表版本号如果持续递增，意味着有路由正在翻动，导致路由持续更新。

对于外部协议来说，路由器配置命令 **network** 中的 IP 网络只控制着通告哪些网络。这与内部网关协议（IGP）是不同的，比如 EIGRP，在 IGP 中 **network** 命令指定了向哪里发送更新。

有关 BGP 配置的更多信息，用户可以参考 *Inspur INOS IP Configuration Guide, Release 12.4* 中的“IP Routing Protocols”部分。有关指定命令的详细信息，用户可以参考 *Inspur INOS I Command Reference, Volume 2 of 3: Routing Protocols, Release 12.4*。

ISO CLNS 路由的相关信息

无连接路由

国际标准化组织（ISO）无连接网络服务（CLNS）协议是开放系统互联（OSI）模型中网络层的标准。ISO 网络架构中的地址被称为网络服务接入点（NSAP）地址和网络实体名称（NET）。OSI 网络中的每个节点都有一个或多个 NET。除此之外，每个节点还有多个 NSAP 地址。

当用户在设备上使用全局配置命令 **clns routing**，启用了无连接路由后，设备只会做出转发决策，而不提供与路由相关的功能。对于动态路由来说，用户必须同时启用一项路由协议。

设备能够支持中间系统到中间系统（IS-IS）动态路由协议，这是一项为 ISO CLNS 网络提供的基于 OSI 的路由协议。

在使用动态路由时，用户可以使用 IS-IS。这个路由协议能够支持区域（Area）的概念。在一个区域内部，所有路由器都知道如何到达所有系统 ID。在区域之间，路由器指导如何到达适当的区域。IS-IS 支持两个级别的路由：工作站路由（一个区域内）和区域路由（多个区域间）。

ISO IGRP 和 IS-IS NASP 编址机制之间最大的区别在于区域地址的定义。它们都为 Level-1 路由（区域内路由）使用系统 ID。但为区域路由指定的地址却不同。ISO IGRP NSAP 地址中包含三个独立的部分用于实现路由：域（Domain）、区域和系统 ID。IS-IS 地址包含两个部分：单一连续的区域字段（对应域和区域字段）和系统 ID。

注释： 有关 ISO CLNS 的更多详细信息，用户可以参考 *Inspur INOS Apollo Domain, Banyan VINES, DECnet, ISO CLNS and XNS Configuration Guide, Release 12.4*。对于本章中涉及命令的完整语法和用法解释信息，用户可以使用 INOS 命令参考主索引或在线搜索，来参考 *Inspur INOS Apollo Domain, Banyan VINES, DECnet, ISO CLNS and XNS Command Reference, Release 12.4*。

IS-IS 动态路由

IS-IS 是一项 ISO 动态路由协议（ISO 105890 文档中描述了该协议）。与其他路由协议不同，要想启用 IS-IS，用户需要创建一个 IS-IS 路由进程，并把它分配到指定的接口，而不是分配到指定的网络。用户可以在每台三层设备或路由器上，使用多区域 IS-IS 配置语法，指定多个 IS-IS 路由进程。此后用户可以为每个 IS-IS 路由进程实例配置各种参数。

小型 IS-IS 网络被建立为一个单区域网络，网络中的所有路由器都包含在其中。当网络扩大时，通常会由用户进行重新规划，用骨干区域连接所有区域中的全部 Level-2 路由器，这样所有区域也就都连接到了本地区域中。在一个本地区域内部，路由器能够知道如何到达所有系统 ID。在区域之间，路由器知道如何到达骨干，骨干路由器知道如何到达其他区域。

路由器会建立 Level-1 邻接关系，以此来形成一个本地区域内部的路由（工作站路由）。路由器会建立 Level-2 邻接关系，以此来形成 Level-1 区域之间的路由（区域路由）。

单台 Inspur 路由器可以最多参与 29 个区域的路由，同时可以在骨干中执行 Level-2 路由。通常来说，每个路由进程对应着一个区域。默认情况下，路由进程中配置的第一个实例会同时执行 Level-1 和 Level-2 路由。用户可以配置其他路由器实例，这些实例会自动被当作 Level-1 区域。用户必须为 IS-IS 路由进程中的每个实例单独配置各种参数。

对于 IS-IS 多区域路由来说，用户可以只配置一个进程来执行 Level-2 路由，尽管用户可以为每台 Inspur 设备定义多达 29 个 Level-1 区域。如果用户在任意进程上配置了 Level-2 路由，

那么所有其他进程会自动被配置为 Level-1。用户可以通过这种配置方式，同时执行 Level-1 路由。如果一个路由器实例并不需要 Level-2 路由，用户可以使用全局配置命令 **is-type** 来移除 Level-2 功能。用户也可以使用命令 **is-type** 把一个不同的路由器实例配置为 Level-2 路由器。

注释： 有关 IS-IS 的详细信息，用户可以参考 *Inspur INOS IP Configuration Guide, Release 12.4* 中的“IP Routing Protocols”一章。有关这部分涉及命令的完整语法和用途解释信息，用户可以参考 *Inspur INOS IP Command Reference, Release 12.4*。

无间断转发感知

设备中集成的 IS-IS NSF 感知（Awareness）特性能够对 IPv4 提供支持。这个特性能够使支持 NSF 感知特性的客户端设备（CPE）路由器帮助支持 NSF 功能（NSF-Capable）的路由器执行无间断转发。本地路由器不必执行 NSF，但它要能够感知 NSF，以便在切换期间在具有 NSF 功能的邻居路由器上维护完整且精确的路由数据库和链路状态数据库。

这个特性是自动启用的额，无需配置。更多有关该特性的信息，用户可以参考 *Integrated IS-IS Nonstop Forwarding (NSF) Awareness Feature Guide*。

IS-IS 全局参数

用户可以配置下列可选的 IS-IS 全局参数：

- 用户可以通过 route-map 来对默认路由进行控制，以此向 IS-IS 路由域中注入默认路由。用户可以在 route-map 下指定其他过滤选项；
- 用户可以配置路由器在接收到内部检验和错误的 IS-IS LSP 时，或清除损坏的 LSP 时，忽略这些 IS-IS LSP，这样做会让 LSP 初始方重新生成 LSP；
- 用户可以为区域和域分配密码；
- 用户可以创建聚合地址，这个聚合地址以汇总地址（路由汇总）的形式出现在路由表中。从其他路由协议学到的路由也可以被汇总。用来通告这个汇总路由的度量值是所有明细路由中最小的度量值；
- 用户可以设置超载（Overload）比特；
- 用户可以配置 LSP 刷新间隔，以及路由器数据库中能够保留 LSP 而不刷新的最大时间；
- 用户可以为 LSP 的生成、最短路径优先计算，以及部分路由计算设置门限值计时器；
- 用户可以配置设备，让它在 IS-IS 邻接状态发生变化（Up 或 Down）时生成日志消息；
- 如果网络中链路的最大传输单元（MTU）的大小小于 1500 字节，用户可以降低 LSP MTU，

使路由功能能够正常运作；

- 分区避免这条路由器配置命令能够防止在 Level-1-2 边界路由器、邻接 Level-1 路由器和终端主机之间失去完全连接时，区域被分割。

IS-IS 接口参数

用户可以配置一些与接口相关的 IS-IS 可选参数，这些参数与相连的其他路由器无关。但是如果用户改变了一些参数的默认值，比如乘数和时间间隔，明智的做法是也在多个路由器和接口上进行相应的变更。大多数接口参数都可以配置在 Level-1、Level-2，或同时配置在 Level-1 和 Level-2 上。

用户可以配置以下接口级别的参数：

- 接口的默认度量值，这个只会被用作 IS-IS 度量值的一个值，并在未执行服务质量（QoS）路由的环境中使用；
- 接口上使用的 Hello 间隔（接口发送 Hello 数据包的时间长度）或默认的 Hello 数据包乘数，决定了 IS-IS Hello 数据包中发送的保持（Hold）时间。保持时间决定了邻居的等待另一个 Hello 数据包的时长，在这之后它会认为这个邻居已失效。这决定了链路或邻居失效检测的速度，检测到失效后设备就可以重新计算路由了。用户可以在 Hello 数据包丢失频繁，以及 IS-IS 邻接关系意外失效的情况下更改 Hello 乘数。用户可以相应地增加 Hello 乘数和减少 Hello 间隔，这样做可以提高 Hello 协议的可靠性，而不会增加检测链路失效的时间；
- 其他时间间隔：
  - 完整序列号 PDU（CSNP）间隔。CSNP 是由指定路由器发送的，用来保持数据库同步；
  - 重传间隔。这是点到点连路上重传 IS-IS LSP 的时间间隔；
  - IS-IS LSP 重传门限值间隔。这是在点到点链路上，重新发送 IS-IS LSP 的最大速率（数据包之间的毫秒时长）。这个间隔与重传间隔不同，重传间隔是成功重传相同 LSP 的时间；
- 指定路由器选举优先级，用户可以在多访问网络上根据需要降低邻接数量，从而减少路由协议流量和拓扑数据库的大小；
- 接口电路类型，在指定接口上与邻居建立邻接关系所需的类型；
- 为接口配置密码认证

如何配置 ISO CLNS 路由

默认的 IS-IS 配置

表 116：默认的 IS-IS 配置

特性	默认设置
忽略链路状态 PDU（LSP）错误	启用
IS-IS 类型	常规 IS-IS：同时充当 Level-1（工作站）和 Level-2（区域）路由器的设备 多区域 IS-IS：IS-IS 路由进程的第一个实例是 Level-2 路由器。其他实例是 Level-1 路由器
生成默认信息	禁用
生成 IS-IS 邻接状态变化日志	禁用
LSP 生成门限值计时器	连续 LSP 之间的最大时间间隔：5 秒钟 初始 LSP 生成延迟：50 毫秒 第一个和第二个 LSP 之间的保持时间：5000 毫秒
LSP 最大生存时间（无刷新）	1200 秒钟（20 分钟）后 LSP 数据包会被删除
LSP 刷新间隔	每 900 秒钟（15 分钟）发送 LSP 刷新
最大 LSP 数据包大小	1497 字节
NSF 感知	启用。允许三层设备在硬件或软件变更期间持续转发从具有 NSF 功能的邻居路由器来的数据包
部分路由计算（PRC）门限值计时器	最大 PRC 等待间隔：5 秒钟 拓扑变化后初始 PRC 计算延迟：2000 毫秒 第一次和第二次 PRC 计算之间的保持时间：5000 毫秒
分区避免特性	禁用
密码	未定义区域密码或域密码，认证功能是禁用的
设置超载比特	禁用。启用后，如果用户没有输入任何变量，

	设备会立即设置超载比特并保持这个设置直到用户输入了 **no set-overload-bit** 命令
最短路径优先（SPF）门限值计时器	连续 SFP 之间的最大间隔：10 秒钟 拓扑变化后初始 SFP 计算：5500 毫秒 第一次和第二次 SFP 计算之间的保持时间：5500 毫秒
汇总地址	禁用

启用 IS-IS 路由

要想启用 IS-IS，用户需要为每个路由进程指定一个名称和 NET。之后在接口上启用 IS-IS 路由，并未每个路由进程实例指定区域。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**clns routing** **示例：** `Device(config)# clns routing`	在交换机上启用 ISO 无连接路由
步骤 3	**router isis** [*area tag*] **示例：** `Device(config)# router isis tag1`	为指定路由进程启用 IS-IS 路由并进入 IS-IS 路由配置模式 （可选）使用 *area tag* 参数来指定这个 IS-IS 路由器所属的区域。用户在配置多个 IS-IS 区域时，必须输入一个值 默认第一个配置的 IS-IS 实例是 Level-2。之后的实例会自动成为 Level-1。用户可以使用全局配置命令 **is-type** 来更改路由级

		别
步骤 4	**net** network-entity-title **示例：** `Device(config-router)# net 47.0004.004d.0001.0001.0c11.1111.00`	为这个路由进程配置 NET。如果用户在配置多区域 IS-IS，就要为每个路由进程指定一个 NET。用户可以为 NET 和地址指定名称
步骤 5	**is-type** {**level-1** | **level-1-2** | **level-2-only**} **示例：** `Device(config-router)# is-type level-2-only`	（可选）把路由器配置为 Level-1 路由器、或提供多区域路由的 Level-2 路由器，或者 Level-1 和 Level-2 路由器（默认）： • **level-1**——只作为工作站路由器 • **level-1-2**——同时作为工作站路由器和区域路由器 • **level-2**——只作为区域路由器
步骤 6	**exit** **示例：** `Device(config-router)# exit`	返回全局配置模式
步骤 7	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/1`	指定接口提供 IS-IS 路由，并进入接口配置模式。如果接口还没有被配置为三层接口，用户需要使用 **no switchport** 命令把接口置于三层模式
步骤 8	**ip router isis** [*area tag*] **示例：** `Device(config-if)# ip router isis tag1`	在接口上为 ISO CLNS 配置 IS-IS 路由进程，并把一个区域指示符关联到这个路由进程
步骤 9	**clns router isis** [*area tag*]	在接口上启用 ISO CLNS

	示例： `Device(config-if)# clns router isis tag1`	
步骤 10	**ip address** *ip-address-mask* **示例：** `Device(config-if)# ip address 10.0.0.5 255.255.255.0`	为接口定义一个 IP 地址。如果在接口上配置 IS-IS 路由的话，区域中所有启用了 IS-IS 的借口上都需要配置 IP 地址
步骤 11	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 12	**show isis** [*area tag*] **database detail** **示例：** `Device# show isis database detail`	检查用户输入的信息
步骤 13	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 IS-IS 全局参数

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**clns routing**	在交换机上启用 ISO 无连接路由

	示例： `Device(config)# clns routing`	
步骤 3	**router isis** **示例：** `Device(config)# router isis`	指定 IS-IS 路由协议，并进入路由器配置模式
步骤 4	**default-information originate** [**route-map** *map-name*] **示例：** `Device(config-router)# default-information originate route-map map1`	（可选）在 IS-IS 路由域中添加默认路由。如果用户输入了 **route-map** *map-name* 命令，路由进程就会当 route-map 中的条件满足时生成默认路由
步骤 5	**ignore-lsp-errors** **示例：** `Device(config-router)# ignore-lsp-errors`	（可选）配置路由器忽略有内部校验和错误的 LSP，而是清除 LSP。这条命令默认就是启用的（损坏的 LSP 会被丢弃）。要想清除损坏的 LSP，用户可以使用路由器配置模式的命令 **no ignore-lsp-errors**
步骤 6	**area-password** *password* **示例：** `Device(config-router)# area-password 1password`	（可选）配置区域认证密码，它会被插入在 Level-1（工作站路由器级别）LSP 中
步骤 7	**domain-password** *password* **示例：** `Device(config-router)# domain-password 2password`	（可选）配置路由域认证密码，它会被插入在 Level-2（区域路由器级别）LSP 中
步骤 8	**summary-address** *address mask* [**level-1** | **level-1-2** | **level-2**]	（可选）为指定级别创建一个汇总地址

	示例： `Device(config-router)# summary-address 10.1.0.0 255.255.0.0 level-2`	
步骤 9	**set-overload-bit** [**on-startup** {*seconds* | **wait-for-bgp**}] **示例：** `Device(config-router)# set-overload-bit on-startup wait-for-bgp`	（可选）设置超载比特（hippity 比特），允许其他路由器在这台路由器出现问题时，在它们的最短路径优先（SPF）计算中忽略这台路由器。 • （可选）**on-startup**——在启动时设置超载比特。如果没有指定 **on-startup** 的话，设备会立即设置超载比特，并且维持这个设置直到用户输入了 **no set-overload-bit** 命令。如果指定了 **on-startup** 的话，用户必须输入一个秒数或 **wait-for-bgp** • *sconds* ——在用户配置了 **on-startup** 关键字后，设备会在系统启动时设置超载比特，并维持这个设置指导 BGP 收敛完成。如果 BGP 没有通知 IS-IS 收敛完成，IS-IS 会在 10 分钟后撤销超载比特
步骤 10	**lsp-refresh-interval** *seconds* **示例：** `Device(config-router)# lsp-refresh-interval 1080`	（可选）以秒为单位设置 LSP 刷新间隔。取值范围是 1 至 65535 秒。默认每 900 秒钟（15 分钟）发送 LSP 刷新
步骤 11	**max-lsp-lifetime** *seconds* **示例：** `Device(config-router)# max-lsp-lifetime 1000`	（可选）设置在刷新前，LSP 数据包保留在路由器数据库中的最大时间。取值范围是 1 至 65535 秒。默认值为 1200 秒钟（20 分钟）。在指定时间间隔后，设备会删除 LSP 数据包

步骤 12	**lsp-gen-interval** [**level-1** | **level-2**] *lsp-max-wait* [*lsp-initial-wait* *lsp-second-wait*] **示例：** `Device(config-router)# lsp-gen-interval level-2 2 50 100`	（可选）设置 IS-IS LSP 生成门限值计时器： • *lsp-max-wait*——生成两个连续 LSP 之间的最大间隔（以秒为单位）。取值范围是 1 至 120，默认值为 5 • *lsp-initial-wait*——初始 LSP 生成延迟（以毫秒为单位）。取值范围是 1 至 10000；默认值为 50 • *lsp-second-wait*——生成第一个和第二个 LSP 之间的保持时间（以毫秒为单位）。取值范围是 1 至 10000；默认值为 5000
步骤 13	**spf-interval** [**level-1** | **level-2**] *spf-max-wait* [*spf-initial-wait* *spf-second-wait*] **示例：** `Device(config-router)# spf-interval level-2 5 10 20`	（可选）设置 IS-IS 最短路径优先（SPF）门限值计时器： • *spf-max-wait*——两次连续 SPF 计算之间的最大间隔（以秒为单位）。取值范围是 1 至 120，默认值为 10 • *spf-initial-wait*——拓扑变化后初始 SPF 计算延迟（以毫秒为单位）。取值范围是 1 至 10000；默认值为 5500 • *spf-second-wait*——第一次和第二次 LSP 计算之间的保持时间（以毫秒为单位）。取值范围是 1 至 10000；默认值为 5500
步骤 14	**prc-interval** *prc-max-wait* [*prc-initial-wait* *prc-second-wait*] **示例：** `Device(config-router)# prc-interval 5 10 20`	（可选）设置 IS-IS 部分路由计算（PRC）门限值计时器： • *prc-max-wait*——两次连续 PRC 计算之间的最大间隔（以秒为单位）。取值范围是 1 至 120，默认值为 5 • *prc-initial-wait*——拓扑变化后初始 PRC 计算延迟（以毫秒为单位）。取值范围是 1 至 10000；默认值为 2000

		• *prc-second-wait*——第一次和第二次 PRC 计算之间的保持时间（以毫秒为单位）。取值范围是 1 至 10000；默认值为 5000
步骤 15	**log-adjacency-changes** [**all**] **示例：** `Device(config-router)# log-adjacency-changes all`	（可选）设置路由器在 IS-IS 邻接关系发生变化时生成日志消息。输入 **all** 来包含所有与中间系统到中间系统 Hello 无关的事件引发的变化，其中包括终端系统到中间系统 PDU 和链路状态数据包（LSP）
步骤 16	**lsp-mtu** *size* **示例：** `Device(config-router)# lsp mtu 1560`	（可选）以字节为单位指定 LSP 数据包的最大值。取值范围是 128 至 4352；默认值为 1497 字节。 **注释：** 如果网络中的任意链路的 MTU 值比较小，用户必须在网络中所有路由器上更改 LSP MTU 大小
步骤 17	**partition avoidance** **示例：** `Device(config-router)# partition avoidance`	（可选）让 IS-IS Level-2 边界路由器在骨干路由器、所有邻接 Level-1 路由器和终端主机之间失去完全连接时，停止向 Level-2 骨干通告 Level-1 区域前缀
步骤 18	**end** **示例：** `Device(config-router)# end`	返回特权 EXEC 模式
步骤 19	**show clns** **示例：** `Device# show clns`	检查用户输入的信息
步骤 20	**copy running-config startup-config** **示例：**	（可选）把输入的命令保存到配置文件中

	Device# **copy running-config startup-config**	

配置 IS-IS 接口参数

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式
步骤 2	**interface** *interface-id* **示例：** Device(config)# interface gigabitethernet 1/0/1	指定接口提供 IS-IS 路由，并进入接口配置模式。如果接口还没有被配置为三层接口，用户需要使用 **no switchport** 命令把接口置于三层模式
步骤 3	**isis metric** *default-metric* [**level-1** | **level-2**] **示例：** Device(config-if)# isis metric 15	（可选）为指定接口配置度量值（或开销值）。取值范围是 0 至 63，默认值为 0。如果用户没有输入级别，默认是同时应用 Level-1 和 Level-2 路由器
步骤 4	**isis hello-interval** {*seconds* | **minimal**} [**level-1** | **level-2**] **示例：** Device(config-if)# isis hello-interval minimal	（可选）指定交换机发送 Hello 数据包的间隔时间。默认情况下，Hello 间隔 *seconds* 的 3 倍会在 Hello 数据包中被通告为 *holdtime* 值。使用较小的 Hello 间隔会更快发现拓扑变化，但也会生成更多的路由流量。 • **minimal**——让系统根据 Hello 乘数来计算 Hello 间隔，最后保持时间会是 1 秒钟 • *seconds*——取值范围是 1 至 65535。

		默认值是 0 秒钟
步骤 5	**isis hello-multiplier** *multiplier* [**level-1** | **level-2**] **示例：** `Device(config-if)# isis hello-multiplier 5`	（可选）指定邻居在没有收到多少个 IS-ISHello 数据包后，才能认为邻接关系已失效。取值范围是 3 至 1000。默认值为 3。使用较小的 Hello 乘数会实现更快的收敛，但也会损害路由稳定性
步骤 6	**isis csnp-interval** *seconds* [**level-1** | **level-2**] **示例：** `Device(config-if)# isis csnp-interval 15`	（可选）为接口配置 IS-IS 完成序列号 PDU（CSNP）间隔。取值范围是 0 至 65535，默认值为 10 秒钟
步骤 7	**isis retransmit-interval** *seconds* **示例：** `Device(config-if)# isis retransmit-interval 7`	（可选）以秒为单位，为点到点链路配置 IS-IS LSP 重传间隔。用户指定的值应该是整数，并且要大于网络中任意两台路由器之间的往返延迟。取值范围是 0 至 65535，默认值为 5 秒钟
步骤 8	**isis retransmit-throttle-interval** *milliseconds* **示例：** `Device(config-if)# isis retransmit-throttle-interval 4000`	（可选）配置 IS-IS LSP 重传门限值间隔，这是点到点链路上重新发送 IS-IS LSP 的最大速率（数据包之间的毫秒数）。取值范围是 0 至 65535，默认值是由 **isis lsp-interval** 命令决定的
步骤 9	**isis priority** *value* [**level-1** | **level-2**] **示例：** `Device(config-if)# isis priority 50`	（可选）配置选举指定路由器使用的优先级。取值范围是 0 至 127，默认值为 64
步骤 10	**isis circuit-type** {**level-1** | **level-1-2** | **level-2-only**}	（可选）配置指定接口上连接的邻居所使用的邻接类型（指定接口电路类型）。

	示例： `Device(config-if)# isis circuit-type level-1-2`	• **level-1**——如果这个节点与邻居上至少有一个相同的区域地址，就把邻接关系设置为 Level-1 • **level-2**——如果邻居上同时配置了 Level-1 和 Level-2，并且至少有一个共同的区域，就把邻接关系设置为 Level-1 和 Level-2。如果没有共同区域，就建立 Level-2 邻接关系。这是默认设置 • **level-2**——建立 Level-2 邻接关系。如果邻居路由器是 Level-1 路由器的话，它们之间就不会建立邻接关系
步骤 11	**isis password** *password* [**level-1** | **level-2**] **示例：** `Device(config-if)# isis password secret`	（可选）为接口配置认证密码。默认情况下，认证是禁用的。指定 Level-1 或 Level-2，表示分别只为 Level-1 或 Level-2 启用密码。如果用户没有指定级别，默认为 Level-1 和 Level-2
步骤 12	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 13	**show clns interface** *interface-id* **示例：** `Device# show clns interface gigabitethernet 1/0/1`	检查用户输入的信息
步骤 14	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

监控和维护 ISO IGRP 和 IS-IS

用户可以移除 CLNS 缓存中的全部内容，也可以移除指定邻居或路由的信息。用户可以显示指定的 CLNS 或 IS-IS 状态统计信息，比如路由表、缓存和数据库的内容。用户还可以查看有关指定接口、过滤器或邻居的信息。

下面这个表格中列出了一些特权 EXEC 命令，用户可以使用这些命令清除或显示 ISO CLNS 和 IS-IS 路由。有关显示字段的解释信息，用户可以使用 INOS 命令参考主索引或在线搜索，来参考 *Inspur INOS Apollo Domain, Banyan VINES, DECnet, ISO CLNS and XNS Command Reference, Release 12.4*。

表 117：ISP CLNS 和 IS-IS clear 和 show 命令

命令	目的
clear clns cache	清除并重新初始化 CLNS 路由缓存
clear clns es-neighbors	从邻接数据库中移除终端系统（ES）邻居信息
clear clns is-neighbors	从邻接数据库中移除中间系统（IS）邻居信息
clear clns neighbors	从邻接数据库中移除 CLNS 邻居信息
clear clns route	移除动态生成的 CLNS 路由信息
show clns	显示有关 CLNS 网络的信息
show clns cache	显示 CLNS 路由缓存中的条目
show clns es-neighbors	显示 ES 邻居条目，其中包括与之相关联的区域
show clns filter-expr	显示过滤取表达式
show clns filter-set	显示过滤器设置
show clns interface [*interface-id*]	显示每个接口的 CLNS 或 ES-IS 信息
show clns neighbor	显示有关 IS-IS 邻居的信息
show clns protocol	列出本地路由器中每个 IS-IS 或 ISO IGRP 路由进程的与协议相关信息
show clns route	显示路由器知道所有目的地，也就是路由器指导如何向这些目的地路由 CLNS 数据包
show clns traffic	显示本地路由器看到的 CLNS 数据包的相关信息
show ip route isis	显示 IS-IS IP 路由表的当前状态
show isis database	显示 IS-IS 链路状态数据库
show isis routes	显示 IS-IS Level-1 路由表

show isis spf-log	显示 IS-IS 最短路径优先（SPF）计算的历史
show isis topology	显示所有区域中连接的所有路由器
show route-map	显示用户配置的所有 route-map 或指定 route-map
trace clns *destination*	发现去往指定目的地的数据包在网络中使用的路径
which-route {*nsap-address* | *clns-name*}	显示拥有指定 CLNS 目的地的路由表

ISO CLNS 路由的配置示例

示例：配置 IS-IS 路由

这个示例展示了如何配置三台路由器把常规 IS-IS 当作 IP 路由协议。在常规 IS-IS 中，所有路由器都是 Level-1 和 Level-2 路由器（默认）。

路由器 A：

```
Device(config)# clns routing
Device(config)# router isis
Device(config-router)# net 49.0001.0000.0000.000a.00
Device(config-router)# exit
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip router isis
Device(config-if)# clns router isis
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# ip router isis
Device(config-if)# clns router isis
Device(config-router)# exit
```

路由器 B：

```
Device(config)# clns routing
Device(config)# router isis
Device(config-router)# net 49.0001.0000.0000.000b.00
Device(config-router)# exit
Device(config)# interface gigabitethernet1/0/1
```

```
Device(config-if)# ip router isis
Device(config-if)# clns router isis
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# ip router isis
Device(config-if)# clns router isis
Device(config-router)# exit
```

路由器 C：

```
Device(config)# clns routing
Device(config)# router isis
Device(config-router)# net 49.0001.0000.0000.000c.00
Device(config-router)# exit
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip router isis
Device(config-if)# clns router isis
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# ip router isis
Device(config-if)# clns router isis
Device(config-router)# exit
```

Multi-VRF CE 的相关信息

虚拟专用网（VPN）为在一个 ISP 骨干网络中共享带宽的客户提供了安全性。VPN 是一些共享通用路由表的站点的集合。客户站点通过一个或多个接口连接到服务提供商网络，服务提供商把每个接口关联到 VPN 路由表中，这称为 VPN 路由/转发（VRF）表。

运行 IP Services 或高级 IP Services 特性集的交换机，能够支持客户端边缘（CE）设备中的多 VPN 路由/转发（Multi-VRF）实例（Multi-VRF CE）。Multi-VRF CE 使一个服务提供商能够使用重叠 IP 地址支持两个或多个 VPN。

注释： 交换机不使用多协议标签交换（MPLS）来支持 VPN。

理解 Multi-VRF CE

Multi-VRF CE 是一项特性，使服务提供商能够支持两个或多个 VPN，并且 VPN 之间的 IP 地址可以重叠。Multi-VRF CE 特性使用入站接口来区分来自不同 VPN 的路由，并通过把一个或

多个三层接口与每个 VRF 相关联，来形成虚拟数据包转发表。一个 VRF 中的接口可以是物理接口，比如以太网端口；或者逻辑接口，比如 VLAN SVI 接口；但一个接口不能同时属于多个 VRF。

注释： Multi-VRF CE 接口必须是三层接口。

Multi-VRF CE 包含以下设备：

- 客户边界（CE）设备负责通过一条数据链路，通过一台或多台服务提供商边界路由器，为客户提供服务提供商网络的接入服务。CE 设备会向路由器通告站点的本地路由，并从路由器学习远端 VPN 路由。交换机可以是 CE 设备；
- 服务提供商边缘（PE）路由器会与 CE 设备之间交换路由信息，它会使用静态路由或路由协议，比如 BGP、RIPv2、OSPF 或 EIGRP。每台 PE 路由器只需要为与其直连的 VPN 维护 VPN 路由，而无需维护服务提供商的所有 VPN 路由。每台 PE 路由器会为它直连的每站点各维护一个 VRF。PE 路由器上的多个接口可以分配给一个 VRF，如果这些站点都参与同一个 VPN 的话。每个 VPN 都映射到指定 VRF 中。在从 CE 路由器学到本地 VPN 路由后，PE 路由器会使用内部 BGP（IBGP）与其他 PE 路由器交换 VPN 路由信息；
- 服务提供商路由器或核心路由器是指服务提供商网络中的路由器，它们不与 CE 设备相连。

在使用 Multi-VRF CE 时，多个客户可以共享一台 CE 路由器，并且在 CE 路由器和 PE 路由器之间只有一条物理链路。共享的 CE 路由器会为每个客户维护单独的 VRF 表，并且会根据每个客户自己的路由表，为这些客户交换或路由数据包。Multi-VRF CE 特性把有限的 PE 路由器功能扩展到了 CE 设备上，让它能够维护独立的 VRF 表，把 VPN 的私密性和安全性扩展到了分支办公室。

网络拓扑

下图展示了使用交换机作为多虚拟 CE 设备的配置。这个环境适用于通过低带宽实现 VPN 服务的客户，比如小公司。在这个示例中，交换机上需要支持 Multi-VRF CE 特性。由于 Multi-VRF CE 特性是三层特性，因此 VRF 中的每个接口必须是三层接口。

图 96：交换机充当多虚拟 CE 设备

Service provider	服务 提供商
CE = Customer-edge device PE = Provider-edge device	CE = 客户边缘设备 PE = 服务提供商边缘设备

当 CE 交换机收到命令让它把一个三层接口添加到 VRF 中时，它会在 Multi-VRF-CE 相关的数

据结构中建立 VLAN ID 与策略标签（PL）的相应映射，并把 VLAN ID 和 PL 添加到 VLAN 数据库中。

当用户在配置 Multi-VRF CE 时，三层转发表在概念上分为以下两个部分：

- Multi-VRF CE 路由部分，包含来自不同 VPN 的路由；
- 全局路由部分，包含去往非 VPN 网络的路由，比如去往 Internet。

来自不同 VRF 的 VLAN ID 会被映射到不同的策略标签，策略标签用来在处理中区分 VRF。对于每一条新学到的 VPN 路由，三层建立功能会使用入向端口的 VLAN ID 来检索策略标签，并且把策略标签和新路由插入到 Multi-VRF CE 路由部分。如果交换机是从一个路由端口收到的数据包，它会使用这个端口的内部 VLAN ID；如果是从一个 SVI 接口收到的数据包，它会使用 VLAN 号。

数据包转发过程

在启用了 Multi-VRF CE 特性的网络中，数据包的交换过程如下所示：

- 当交换机从一个 VPN 那里接收到了一个数据包，它会根据入站策略标签号来查看路由表。当找到相应的路由时，它会把数据包转发给 PE 设备；
- 当入向 PE 设备从 CE 设备那里接收到数据包后，它会执行 VRF 查找。当找到相应的路由时，它会在数据包上添加相应的 MPLS 标签，并把数据包发送到 MPLS 网络中；
- 当出向 PE 设备从网络中接收到数据包后，它会剥除标签，并使用标签来识别正确的 VPN 路由表。之后它会执行普通路由查询。当找到相应的路由后，它会把数据包转发给正确的邻接设备；
- 当 CE 设备从出向 PE 设备那里接收到数据包后，它会使用入站策略标签来查找正确的 VPN 路由表。当找到相应的路由后，它会把数据包转发到 VPN 中。

网络组成部分

要想配置 VRF，用户需要创建一个 VRF 表，并指定与这个 VRF 相关联的三层接口。然后在 VPN、CE 设备和 PE 设备之间配置一个路由协议。BGP 是在服务提供商骨干中分布 VPN 路由信息的首选路由协议。Multi-VRF CE 网络有以下三大组成部分：

- VPN 路由目标团体——列出一个 VPN 团体的所有其他成员。用户需要为每个 VPN 团体成员配置 VPN 路由目标；
- VPN 团体 PE 路由器的多协议 BGP 对等体——向一个 VPN 团体中的所有成员传播 VRF 可达性信息。用户需要在一个 VPN 团体中的所有 PE 路由器上都配置 BGP 对等体；

- VPN 转发——在 VPN 服务提供商网络中，在所有 VPN 团体成员之间传输所有流量。

VRF 感知服务

用户可以在全局接口上配置 IP 服务，这些服务运行在全局路由实例中。IP 服务能够运行在多个路由实例中；它们具有 VRF 感知功能。系统中配置的任何 VRF 都可以被指定为一个 VRF 感知服务。

VRF 感知服务是在与平台无关的模块中进行实施的。VRF 在 Inspur INOS 中表示多路由实例。每个平台能够支持不同数量的 VRF。

VRF 感知服务拥有以下特征：

- 用户可以 ping 用户指定 VRF 中的主机；
- 在单独的 VRF 中学习 ARP 条目。用户可以查看指定 VRF 中的地址解析协议（ARP）条目。

如何配置 Multi-VRF CE

默认的 Multi-VRF CE 配置

表 118：默认的 VRF 配置

特性	**默认设置**
VRF	禁用。未定义任何 VRF
各种 map	未指定 import-map、export-map 或 route-map
VRF 最大路由条目	快速以太网交换机：8000；千兆以太网交换机：12000
转发表	接口的默认设置是全局路由表

Multi-VRF CE 配置指导

注释： 要想使用 Multi-VRF CE 特性，用户必须在交换机上启用 IP Services 或高级 IP Services 特性集。

- 启用了 Multi-VRF CE 特性的交换机是由多个客户共享的，每个客户拥有自己的路由表；
- 由于客户使用不同的 VRF 表，因此可以复用相同的 IP 地址。不同的 VPN 中可以使用重叠 IP 地址；

- Multi-VRF CE 特性可以让多个客户共享 PE 和 CE 之间相同的物理链路。允许传输多个 VLAN 数据的 Trunk 端口用来区分多个客户之间的数据包。每个客户有自己的 VLAN；
- Multi-VRF CE 特性不支持全部 MPLS-VRF 功能。它不支持标签交换、LDP 邻接或携带标签的数据包；
- 对于 PE 路由器来说，使用 Multi-VRF CE 或使用多 CE 没有任何区别。在图 41-6 中，多个虚拟三层接口连接到一台 Multi-VRF CE 设备；
- 交换机支持使用物理端口、VLAN SVI 接口或同时使用两者来配置 VRF。SVI 接口可以通过 Access 端口或 Trunk 端口进行连接；
- 客户可以使用多个 VLAN，只要不与其他客户重叠就行。客户的 VLAN 会映射为一个指定的路由表 ID，这个路由表 ID 用来识别交换机上储存的相应路由表；
- 交换机能够支持一个全局网络和最多 26 个 VRF；
- CE 设备和 PE 设备之间可以使用多种路由协议（BGP、OSPF、RIP 和静态路由）。但我们建议使用外部 BGP（EBGP），原因如下所示：
  - BGP 不需要使用多种算法与多个 CE 进行通信；
  - BGP 设计用来在属于不同管理域的系统之间传递路由信息；
  - BGP 使得向 CE 传输路由属性变得容易。
- Multi-VRF CE 特性不会影响数据包的交换速率；
- 不支持 VPN 组播；
- 用户可以在私有 VLAN 上启用 VRF，反之亦然；
- 用户不能当接口上启用了策略路由（PBR）时启用 VRF，反之亦然；
- 用户不能在接口上启用了 Web 缓存通信协议（WCCP）时启用 VRF，反之亦然。

配置 VRF

有关命令的完整语法和使用信息，用户可以参考这个版本的交换机命令参考和 *Inspur INOS Switching Services Command Reference*。

注释： 在堆栈交换机上更改 VRF 配置时，建议重启整个堆栈。这对于维护 CEF 和 VRF 控制平面之间的统一性是重要的做法，避免在主用设备切换时出现一致性错误消息。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal**	进入全局配置模式

	示例： `Device# configure terminal`	
步骤 2	**ip routing** **示例：** `Device(config)# ip routing`	启用 IP 路由功能
步骤 3	**ip vrf** *vrf-name* **示例：** `Device(config)# ip vrf vpn1`	为 VRF 命名，并进入 VRF 配置模式
步骤 4	**rd** *route-distinguisher* **示例：** `Device(config-vrf)# rd 100:2`	通过定义路由识别符来创建一个 VRF 表。用户可以输入一个 AS 号和一个任意编号（xxx:y），或者一个 IP 地址和一个任意编号（A.B.C.D:y）
步骤 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community* **示例：** `Device(config-vrf)# route-target both 100:2`	为指定 VRF 创建一个输入列表、输出列表，或出入和输出路由目标团体。用户可以输入一个 AS 号和一个任意编号（xxx:y），或者一个 IP 地址和一个任意编号（A.B.C.D:y）。*route-target-ext-community* 应该与步骤 4 中输入的 *route-distinguisher* 相同
步骤 6	**import map** *route-map* **示例：** `Device(config-vrf)# import map importmap1`	（可选）为 VRF 关联一个 route-map
步骤 7	**interface** *interface-id* **示例：** `Device(config-vrf)# interface gigabitethernet 1/0/1`	指定与 VRF 相关联的三层接口，并进入接口配置模式。这个接口可以是路由端口或 SVI 接口

步骤 8	**ip vrf forwarding** *vrf-name* **示例：** `Device(config-if)# ip vrf forwarding vpn1`	关联 VRF 和三层接口。 **注释：** 如果管理接口上启用了 **ip vrf forwarding**，接入点不会加入
步骤 9	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 10	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] **示例：** `Device# show ip vrf interfaces vpn1`	检查用户输入的配置。显示有关用户配置的 VRF 的相关信息
步骤 11	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 VRF 感知服务

以下这些服务具有 VRF 感知功能：

- ARP
- Ping
- 简单网络管理协议（SNMP）
- 单播反向路径转发（uRPF）
- 系统日志（Syslog）
- 路由追踪（Traceroute）
- FTP 和 TFTP

注释： 交换机不能为单播反向路径转发（uRPF）或网络时间协议（NTP）支持 VRF 感知服务。

为 ARP 配置 VRF 感知服务

有关命令的完整语法和使用信息，用户可以参考这个版本的交换机命令参考和 *Inspur INOS Switching Services Command Reference, Release 12.4*。

具体步骤

	命令或操作	目的
步骤 1	**show ip arp vrf** *vrf-name* **示例：** `Device# show ip arp vrf vpn1`	显示指定 VRF 中的 ARP 表

为 Ping 配置 VRF 感知服务

有关命令的完整语法和使用信息，用户可以参考这个版本的交换机命令参考和 *Inspur INOS Switching Services Command Reference, Release 12.4*。

具体步骤

	命令或操作	目的
步骤 1	**ping vrf** *vrf-name* **ip-host** **示例：** `Device# ping vrf vpn1 ip-host`	显示指定 VRF 中的 ARP 表

为 SNMP 配置 VRF 感知服务

有关命令的完整语法和使用信息，用户可以参考这个版本的交换机命令参考和 *Inspur INOS Switching Services Command Reference, Release 12.4*。

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 2	**snmp-server trap authentication vrf** **示例：** `Device(config)# snmp-server trap authentication vrf`	在 VRF 上为数据包启用 SNMP Trap
步骤 3	**snmp-server engineID remote** *host* **vrf** *vpn-instance engine-id string* **示例：** `Device(config)# snmp-server engineID remote 172.16.20.3 vrf vpn1 80000009030000B064EFE100`	在交换机上为远端 SNMP 引擎配置一个名称
步骤 4	**snmp-server host** *host* **vrf** *vpn-instance* **traps** *community* **示例：** `Device(config)# snmp-server host 172.16.20.3 vrf vpn1 traps comaccess`	指定 SNMP Trap 消息的接收方，并指定用来发送 SNMP Trap 消息的 VRF 表
步骤 5	**snmp-server host** *host* **vrf** *vpn-instance* **informs** *community* **示例：** `Device(config)# snmp-server host 172.16.20.3 vrf vpn1 informs comaccess`	指定 SNMP Inform 消息的接收方，并指定用来发送 SNMP Inform 消息的 VRF 表
步骤 6	**snmp-server user** *user group* **remote** *host* **vrf** *vpn-instance security model* **示例：** `Device(config)# snmp-server`	为 VRF 上的远端主机，在 SNMP 组中添加一个用户

	user abcd remote 172.16.20.3 vrf vpn1 priv v2c 3des secure3des	
步骤 7	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式

为 uRPF 配置 VRF 感知服务

用户可以在分配到 VRF 的接口上配置 uRPF 特性，源查找是在 VRF 表中完成的。

有关命令的完整语法和使用信息，用户可以参考这个版本的交换机命令参考和 *Inspur INOS Switching Services Command Reference, Release 12.4*。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式
步骤 2	**interface** *interface-id* **示例：** Device(config)# interface gigabitethernet 1/0/1	进入接口配置模式，并指定用户要配置的三层接口
步骤 3	**no switchport** **示例：** Device(config-if)# no switchport	如果这是一个物理接口的话，移除接口的二层模式
步骤 4	**ip vrf forwarding** *vrf-name* **示例：**	在接口上配置 VRF

	`Device(config-if)# ip vrf forwarding vpn2`	
步骤 5	**ip address** *ip-address* **示例：** `Device(config-if)# ip address 10.1.5.1`	为接口配置 IP 地址
步骤 6	**ip verify unicast reverse-path** **示例：** `Device(config-if)# ip verify unicast reverse-path`	在接口上启用 uRPF
步骤 7	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式

配置 VRF 感知 RADIUS

要想配置 VRF 感知 RADIUS，用户必须首先在一台 RADIUS 服务器上启用 AAA。交换机能够支持服务组配置 **ip vrf forwarding** *vrf-name* 和全局配置命令 **ip radius source-interface**，具体信息用户可以参考 VRF AAA 特性指导。

为 Syslog 配置 VRF 感知服务

有关命令的完整语法和使用信息，用户可以参考这个版本的交换机命令参考和 *Inspur INOS Switching Services Command Reference, Release 12.4*。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：**	进入全局配置模式

	Device# **configure terminal**	
步骤 2	**logging on** **示例：** Device(config)# logging on	为存储路由器时间消息启用或暂时禁用日志功能
步骤 3	**logging host** *ip-address* **vrf** *vrf-name* **示例：** Device(config)# logging host 10.10.1.0 vrf vpn1	指定日志服务器的主机地址，也就是向哪里发送日志消息
步骤 4	**logging buffered** *logging buffered size* **debugging** **示例：** Device(config)# logging buffered critical 6000 debugging	向内部缓存记录日志消息
步骤 5	**logging trap debugging** **示例：** Device(config)# logging trap debugging	限制发送到系统日志服务器的日志消息
步骤 6	**logging facility** *facility* **示例：** Device(config)# logging facility user	向日志设备发送系统日志消息
步骤 7	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式

为 Traceroute 配置 VRF 感知服务

有关命令的完整语法和使用信息，用户可以参考这个版本的交换机命令参考和 *Inspur INOS Switching Services Command Reference, Release 12.4*。

具体步骤

	命令或操作	目的
步骤 1	**traceroute vrf** *vrf-name ipaddress* **示例：** `Device(config)# traceroute vrf vpn2 10.10.1.1`	指定 VPN VRF 的名称，也就是在这里查找目的地址

为 FTP 和 TFTP 配置 VRF 感知服务

FTP 和 TFTP 是具有 VRF 感知的服务，用户必须配置一些 FTP/TFTP CLI。举例来说，如果用户希望使用一个关联在接口的 VRF 表，比如说 E1/0，用户需要配置 ip tftp source-interface E1/0 或 ip ftp source-interface E1/0 命令，来告知 TFTP 或 FTP 服务器使用哪个路由表。再找个示例中，VRF 表会被用来查找目的 IP 地址。这些变更具有向后兼容性，不会影响现有行为。也就是说，用户可以使用源接口 CLI 从指定接口向外发送数据包，即使这个接口上配置了 VRF。

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**ip ftp source-interface** *interface-type interface-number* **示例：** `Device(config)# ip ftp source-interface`	为 FTP 连接指定源 IP 地址

	gigabitethernet 1/0/2	
步骤 3	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 4	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式
步骤 5	**ip tftp source-interface** *interface-type interface-number* **示例：** Device(config)# ip tftp source-interface gigabitethernet 1/0/2	为 TFTP 连接指定源 IP 地址
步骤 6	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式

配置组播 VRF

有关命令的完整语法和使用信息，用户可以参考这个版本的交换机命令参考和 *Inspur INOS IP Multicast Command Reference*。

有关在Multi-VRF CE中配置组播的更多信息，用户可以参考*IP Routing: Protocol-Independent Configuration Guide, Inspur INOS Release 15S*。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：**	进入全局配置模式

	Device# **configure terminal**	
步骤 2	**ip routing** **示例：** Device(config)# ip routing	启用 IP 路由模式
步骤 3	**ip vrf** *vrf-name* **示例：** Device(config)# ip vrf vpn1	为 VRF 命名，并进入 VRF 配置模式
步骤 4	**rd** *route-distinguisher* **示例：** Device(config-vrf)# rd 100:2	通过定义路由识别符来创建一个 VRF 表。用户可以输入一个 AS 号和一个任意编号（xxx:y），或者一个 IP 地址和一个任意编号（A.B.C.D:y）
步骤 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community* **示例：** Device(config-vrf)# route-target both 100:2	为指定 VRF 创建一个输入列表、输出列表，或出入和输出路由目标团体。用户可以输入一个 AS 号和一个任意编号（xxx:y），或者一个 IP 地址和一个任意编号（A.B.C.D:y）。*route-target-ext-community* 应该与步骤 4 中输入的 *route-distinguisher* 相同
步骤 6	**import map** *route-map* **示例：** Device(config-vrf)# import map importmap1	（可选）为 VRF 关联一个 route-map
步骤 7	**ip multicast-routing vrf** *vrf-name* **distributed** **示例：** Device(config-vrf)# ip multicast-routing vrf vpn1 distributed	（可选）为 VRF 表启用全局组播路由

步骤 8	**interface** *interface-id* **示例：** `Device(config-vrf)# interface gigabitethernet 1/0/2`	指定与 VRF 相关联的三层接口，并进入接口配置模式。这个接口可以是路由端口或 SVI 接口
步骤 9	**ip vrf forwarding** *vrf-name* **示例：** `Device(config-if)# ip vrf forwarding vpn1`	把 VRF 关联到三层接口
步骤 10	**ip address** *ip-address subnet-mask* **示例：** `Device(config-if)# ip address 10.1.5.1 255.255.255.0`	为三层接口配置 IP 地址
步骤 11	**ip pim sparse-dense mode** **示例：** `Device(config-if)# ip pim sparse-dense mode`	在关联了 VRF 的三层接口上启用 PIM
步骤 12	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 13	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] **示例：** `Device# show ip vrf detail vpn1`	检查用户输入的配置。显示有关用户配置的 VRF 的信息
步骤 14	**copy running-config startup-config**	（可选）把输入的命令保存到配置文件

	示例： `Device# copy running-config startup-config`	中

配置 VPN 路由会话

用户可以配置设备支持的任意路由协议（RIP、OSPF、EIGRP 或 BGP）或静态路由来提供 VPN 中的路由。这个示例中的配置使用的是 OSPF，但其他协议的配置过程也都相同。

注释： 要想在一个 VRF 实例中配置 EGIRP 路由进程，用户必须在地址家族配置模式中，使用命令 **autonomous-system** *autonomous-system-number* 来配置自治系统号。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**router ospf** *process-id* **vrf** *vrf-name* **示例：** `Device(config)# router ospf 1 vrf vpn1`	启用 OSPF 路由，指定一个 VPN 转发表，并进入路由器配置模式
步骤 3	**log-adjacency-changes** **示例：** `Device(config-router)# log-adjacency-changes`	（可选）记录邻接状态的变化。这是默认状态
步骤 4	**redistribute bgp** *autonomous-system-number* **subnets** **示例：** `Device(config-router)#`	设置交换机从 BGP 网络向 OSPF 网络重分发信息

	redistribute bgp 10 subnets	
步骤 5	**network** *network-number* **area** *area-id* **示例：** Device(config-router)# network 1 area 2	定义 OSPF 运行的网络地址和掩码，以及这个网络地址的区域 ID
步骤 6	**end** **示例：** Device(config-router)# **end**	返回特权 EXEC 模式
步骤 7	**show ip ospf** *process-id* **示例：** Device# show ip ospf 1	检查 OSPF 网络的配置
步骤 8	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

配置 BGP PE 到 CE 路由会话

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式
步骤 2	**router bgp** *autonomous-system* **示例：** Device(config)# router bgp 2	启用 BGP 路由进程、指定 AS 号，并进入路由器配置模式

步骤 3	**network** *network-number* **mask** *network-mask* **示例：** `Device(config-router)# network 5 mask 255.255.255.0`	指定使用 BGP 通告的网络和掩码
步骤 4	**redistribute ospf** *process-id* **match internal** **示例：** `Device(config-router)# redistribute ospf 1 match internal`	设置交换机来重分发 OSPF 内部路由
步骤 5	**network** *network-number* **area** *area-id* **示例：** `Device(config-router)# network 5 area 2`	定义 OSPF 运行的网络地址和掩码，以及这个网络地址的区域 ID
步骤 6	**address-family ipv4 vrf** *vrf-name* **示例：** `Device(config-router)# address-family ipv4 vrf vpn1`	为 PE 到 CE 路由会话定义 BGP 参数，并进入 VRF 地址家族模式
步骤 7	**neighbor** *address* **remote-as** *as-number* **示例：** `Device(config-router)# neighbor 10.1.1.2 remote-as 2`	定义 PE 和 CE 路由器之间的 BGP 会话
步骤 8	**neighbor** *address* **activate** **示例：** `Device(config-router)#`	激活 IPv4 地址家族的通告

	neighbor 10.2.1.1 activate	
步骤 9	**end** **示例：** Device(config-router)# **end**	返回特权 EXEC 模式
步骤 10	**show ip bgp** [**ipv4**] [**neighbors**] **示例：** Device# show ip bgp ipv4 neighbors	检查 BGP 配置
步骤 11	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

监控 Multi-VRF CE

表 119：显示 Multi-VRF CE 信息的命令

命令	**目的**
show ip protocols vrf *vrf-name*	显示与 VRF 相关联的路由协议信息
show ip route vrf *vrf-name* [**connected**] [*protocol* [*as-number*]] [**list**] [**mobile**] [**odr**] [**profile**] [**static**] [**summary**] [**supernets-only**]	显示与 VRF 相关联的 IP 路由表信息
show ip vrf [**brief** | **detail** | **interfaces**] [*vrf-name*]	显示用户定义的 VRF 实例的相关信息

命令显示内容的更多信息，用户可以参考 *Inspur INOS Switching Services Command Reference, Release 12.4*。

Multi-VRF CE 的配置示例

Multi-VRF CE 的配置示例

在这个示例中，VPN1、VPN2 和全局网络中使用的路由协议是 OSPF。BGP 用在 CE 和 PE 连接上。下面这个示例展示了如何把交换机配置为 CE 交换机 A，以及如何在交换机 D 和 E 上为客户配置 VRF 配置。CE 交换机 C 和其他客户交换机的配置没有显示出来，这些命令与示例中显示的命令类似。这个示例中还包括为交换机 A 配置流量的命令，把 Inspur 6000 或 Inspur 6500 交换机当作 PE 路由器。

图 97：Multi-VRF CE 配置示例

Switch A	交换机 A
Switch B	交换机 B
Switch C	交换机 C
Switch D	交换机 D
Switch E	交换机 E
Switch F	交换机 F
Switch G	交换机 G
Swich H	交换机 H
Switch J	交换机 J
Global network Switch K	全局网络 交换机 K
Fast Ethernet（共 4 处）	快速 以太网
Gigabit Ethernet	千兆 以太网
CE = Customer-edge device PE = Provider-edge device	CE = 客户边缘设备 PE = 服务提供商边缘设备

在交换机 A 上启用路由功能并配置 VRF。

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# ip routing
```

```
Device(config)# ip vrf v11
Device(config-vrf)# rd 800:1
Device(config-vrf)# route-target export 800:1
Device(config-vrf)# route-target import 800:1
Device(config-vrf)# exit
Device(config)# ip vrf v12
Device(config-vrf)# rd 800:2
Device(config-vrf)# route-target export 800:2
Device(config-vrf)# route-target import 800:2
Device(config-vrf)# exit
```

在交换机 A 上配置环回接口和物理接口。千兆以太网端口通过 Trunk 连接到 PE。千兆以太网端口 8 和 11 连接到 VPN：

```
Device(config)# interface loopback1
Device(config-if)# ip vrf forwarding v11
Device(config-if)# ip address 8.8.1.8 255.255.255.0
Device(config-if)# exit
Device(config)# interface loopback2
Device(config-if)# ip vrf forwarding v12
Device(config-if)# ip address 8.8.2.8 255.255.255.0
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/5
Device(config-if)# switchport trunk encapsulation dot1q
Device(config-if)# switchport mode trunk
Device(config-if)# no ip address
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/8
Device(config-if)# switchport access vlan 208
Device(config-if)# no ip address
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/11
Device(config-if)# switchport trunk encapsulation dot1q
Device(config-if)# switchport mode trunk
Device(config-if)# no ip address
```

```
Device(config-if)# exit
```

配置交换机 A 上使用的 VLAN。VRF 11 在 CE 和 PE 之间使用 VLAN 10。VRF 12 在 CE 和 PE 之间使用 VLAN 20。VLAN 118 和 208 由 VPN 使用，分别包括交换机 F 和交换机 D：

```
Device(config)# interface vlan10
Device(config-if)# ip vrf forwarding v11
Device(config-if)# ip address 38.0.0.8 255.255.255.0
Device(config-if)# exit
Device(config)# interface vlan20
Device(config-if)# ip vrf forwarding v12
Device(config-if)# ip address 83.0.0.8 255.255.255.0
Device(config-if)# exit
Device(config)# interface vlan118
Device(config-if)# ip vrf forwarding v12
Device(config-if)# ip address 118.0.0.8 255.255.255.0
Device(config-if)# exit
Device(config)# interface vlan208
Device(config-if)# ip vrf forwarding v11
Device(config-if)# ip address 208.0.0.8 255.255.255.0
Device(config-if)# exit
```

在 VPN1 和 VPN2 中配置 OSPF 路由。

```
Device(config)# router ospf 1 vrf vl1
Device(config-router)# redistribute bgp 800 subnets
Device(config-router)# network 208.0.0.0 0.0.0.255 area 0
Device(config-router)# exit
Device(config)# router ospf 2 vrf vl2
Device(config-router)# redistribute bgp 800 subnets
Device(config-router)# network 118.0.0.0 0.0.0.255 area 0
Device(config-router)# exit
```

为 CE 到 PE 路由配置 BGP。

```
Device(config)# router bgp 800
Device(config-router)# address-family ipv4 vrf vl2
Device(config-router-af)# redistribute ospf 2 match internal
Device(config-router-af)# neighbor 83.0.0.3 remote-as 100
```

```
Device(config-router-af)# neighbor 83.0.0.3 activate
Device(config-router-af)# network 8.8.2.0 mask 255.255.255.0
Device(config-router-af)# exit
Device(config-router)# address-family ipv4 vrf vl1
Device(config-router-af)# redistribute ospf 1 match internal
Device(config-router-af)# neighbor 38.0.0.3 remote-as 100
Device(config-router-af)# neighbor 38.0.0.3 activate
Device(config-router-af)# network 8.8.1.0 mask 255.255.255.0
Device(config-router-af)# end
```

交换机 D 属于 VPN1。使用以下命令配置交换机 A 的连接。

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# ip routing
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# no switchport
Device(config-if)# ip address 208.0.0.20 255.255.255.0
Device(config-if)# exit
Device(config)# router ospf 101
Device(config-router)# network 208.0.0.0 0.0.0.255 area 0
Device(config-router)# end
```

交换机 F 属于 VPN2。使用以下命令配置交换机 A 的连接。

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# ip routing
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# switchport trunk encapsulation dot1q
Device(config-if)# switchport mode trunk
Device(config-if)# no ip address
Device(config-if)# exit
Device(config)# interface vlan118
Device(config-if)# ip address 118.0.0.11 255.255.255.0
Device(config-if)# exit
Device(config)# router ospf 101
```

```
Device(config-router)# network 118.0.0.0 0.0.0.255 area 0
Device(config-router)# end
```

在交换机 B（PE 路由器）上，以下命令只配置与 CE 设备（交换机 A）的连接。

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip vrf v1
Router(config-vrf)# rd 100:1
Router(config-vrf)# route-target export 100:1
Router(config-vrf)# route-target import 100:1
Router(config-vrf)# exit
Router(config)# ip vrf v2
Router(config-vrf)# rd 100:2
Router(config-vrf)# route-target export 100:2
Router(config-vrf)# route-target import 100:2
Router(config-vrf)# exit
Router(config)# ip cef
Router(config)# interface Loopback1
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 3.3.1.3 255.255.255.0
Router(config-if)# exit
Router(config)# interface Loopback2
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 3.3.2.3 255.255.255.0
Router(config-if)# exit
Router(config)# interface gigabitethernet1/1/0.10
Router(config-if)# encapsulation dot1q 10
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 38.0.0.3 255.255.255.0
Router(config-if)# exit
Router(config)# interface gigabitethernet1/1/0.20
Router(config-if)# encapsulation dot1q 20
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 83.0.0.3 255.255.255.0
```

```
Router(config-if)# exit
Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf v2
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.2.0 mask 255.255.255.0
Router(config-router-af)# exit
Router(config-router)# address-family ipv4 vrf vl
Router(config-router-af)# neighbor 38.0.0.8 remote-as 800
Router(config-router-af)# neighbor 38.0.0.8 activate
Router(config-router-af)# network 3.3.1.0 mask 255.255.255.0
Router(config-router-af)# end
```

配置单播反向路径转发

单播反向路径转发（单播 RPF）特性有助于减少这个问题：当 IP 数据包缺少可验证的 IP 源地址时将其丢弃，从而把异常或伪造（欺骗）IP 源地址引入网络。举例来说，大量常见的拒绝服务（DoS）攻击类型，其中包括 Smurf and Tribal Flood Network（TFN），能够从伪造并快速改变源 IP 地址的行为获益，让攻击者能够抵挡定位攻击者和过滤攻击的行为。对于提供公共接入的 Internet 服务提供商（ISP）来说，单播 RPF 能够通过只转发源地址有效且符合 IP 路由表的数据包，来避免这种攻击。这种行为能够保护 ISP、ISP 的客户，以及 Internet 其他部分的安全。

注释：

- IP Services 特性集能够支持单播 RPF；
- 如果交换机位于一个混好了多种交换机类型的硬件堆栈中，不要配置单播 RPF 特性。举例来说，Inspur 3750-X、Inspur3750-E 和 Inspur 3750 交换机。

与协议无关特性

这部分描述了与协议无关的 IP 路由特性，运行 IP Base 特性集或 IP Services 特性集的交换机上可以支持这个特性；除了 IP Base 特性集之外，只有 RIP 能够使用与协议相关的特性。本章中与协议无关 IP 路由特性的命令详细描述，用户可以参考 *Inspur INOS IP Command*

Reference, Volume 2 of 3: Routing Protocols 中“IP Routing Protocol-Independent Commands”一章。

分布式 Inspur 快速转发

Inspur 快速转发的相关信息

Inspur 快速转发（CEF）是一项三层 IP 交换技术，用来优化网络性能。CEF 实施了高级 IP 查找和转发算法，实现了三层交换的最大性能。CEF 比快速交换路由缓存功能需要更少的 CPU 资源，使 CPU 处理资源能够专注于数据包转发。在交换机堆栈中，硬件会使用堆栈中的分布式 CEF（dCEF）。在动态网络中，快速交换缓存条目会频繁失效，因为路由会发生变化，从而导致流量的交换处理需要依赖于路由表，而不是使用路由缓存的快速交换。CEF 和 dCEF 可以使用转发信息库（FIB）查找表，来为 IP 数据包执行基于目的地的交换。

CEF 和 dCEF 中的量大主要组成部分是分布式 FIB 和分布式邻接表。

- FIB 类似于路由表或信息库，它维护了 IP 路由表中转发信息的镜像条目。当网络中的路由或拓扑发生变化时，IP 路由表就会更新，这些变化也会反映在 FIB 中。FIB 会根据 IP 路由表中的信息来维护下一跳地址信息。由于 FIB 中包含路由表中的所有已知路由，CEF 就无需维护路由缓存，这样做提高了交换流量的效率，并且对流量模式没有影响；
- 网络中的两个节点之间如果可以在链路层通过一跳访问对方，就称这两个节点为邻接关系。CEF 使用邻接表来为二层寻址信息添加前缀。邻接表中维护着所有 FIB 条目的二层下一跳地址。

由于交换机或交换机堆栈会使用专用集成电路（ASIC）来实现吉比特速率的线速 IP 流量传输，CEF 或 dCEF 转发功能只为软件转发路径提供服务，也就是说通过 CPU 进行转发的流量。

如何配置 Inspur 快速转发

CEF 或分布式 CEF 默认是在全局启用的。如果出于某些原因用户禁用了这项特性，也可以使用全局配置命令 **ip cef** 或 **ip cef distributed** 再次启用它。

在默认配置中，所有三层接口上都启用了 CEF 或 dCEF 特性。用户可以使用接口配置模式的命令 **no ip route-cache cef**，为通过软件转发的流量禁用 CEF。这条命令不会影响硬件转发路径。用户可以使用特权 EXEC 命令 **debug ip packet detail** 可以在禁用 CEF 时观察基于软件的流量转发。要想为一个接口的软件转发路径启用 CEF，用户可以使用接口配置命令 **ip**

route-cache cef。

注意： 虽然在使用接口配置命令 **no ip route-cache cef** 禁用了接口上的 CEF 特性后，用户可以在 CLI 命令输出中看到这一状态，但我们强烈建议用户不在接口上禁用 CEF 或 dCEF，除了出于调试（Debugging）目的。

用户可以按照以下步骤，在禁用了 CEF 或 dCEF 特性的环境中，在全局为软件转发流量启用 CEF 或 dCEF，以及在接口上启用 CEF 或 dCEF：

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**ip cef** **示例：** `Device(config)# ip cef`	在非堆栈交换机上启用 CEF 特性。 配置步骤 4
步骤 3	**ip cef distributed** **示例：** `Device(config)# ip cef distributed`	在活跃交换机上启用 CEF 特性
步骤 4	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/1`	进入接口配置模式，并指定用户想要配置的三层接口
步骤 5	**ip route-cache cef** **示例：** `Device(config-if)# ip route-cache cef`	在接口上为基于软件转发的流量启用 CEF
步骤 6	**end**	返回特权 EXEC 模式

	示例： `Device(config)# end`	
步骤 7	**show ip cef** **示例：** `Device# show ip cef`	显示所有接口上的 CEF 状态
步骤 8	**show cef linecard** [**detail**] **示例：** `Device# show cef linecard detail`	（可选）在非堆栈交换机上显示与 CEF 相关的接口信息
步骤 9	**show cef linecard** [*slot-number*] [**detail**] **示例：** `Device# show cef linecard 5 detail`	（可选）使用堆栈中所有交换机的堆栈编号显示交换机上与 CEF 相关的接口信息，或为指定交换机显示与 CEF 相关的接口信息 （可选）在 *slot-number* 部分输入堆栈成员的交换机编号
步骤 10	**show cef interface** [*interface-id*] **示例：** `Device# show cef interface gigabitethernet 1/0/1`	显示所有接口或指定接口上的 CEF 信息
步骤 11	**show adjacency** **示例：** `Device# show adjacency`	显示 CEF 邻接表信息
步骤 12	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

等价路由的路径数量

等价路由路径的相关信息

当路由器中有两条或多条去往相同网络且度量值相同的路由时，这些路由可以被称为等价路由（Equal-Cost Routing）。术语平行路径（Parallel Path）是在路由表中查看等价路由的另一种方式。如果路由器上有去往一个网络的两条或多条等价路径，它可以同时使用这些路径。平行路径会在链路失效时提供冗余性，并且使路由器能够在可用路径上对数据包执行负载均衡，提高可用带宽的利用率。堆栈中的交换机上能够支持等价路由。

尽管路由器能够自动学习和配置等价路由，但用户也可以控制 IP 路由协议能够在其路由表中放入的平行路径最大数量。尽管交换机软件允许最多 32 条等价路由，但交换机硬件不允许为一条路由使用超过 16 条路径。

如何配置等价路由路径

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**router** {**bgp** | **rip** | **ospf** | **eigrp**} **示例：** `Device(config)# router eigrp`	进入路由器配置模式
步骤 3	**maximum-paths** *maximum* **示例：** `Device(config-router)# maximum-paths 2`	为协议路由表设置平行路径的最大数量。取值范围是 1 至 16；大多数 IP 路由协议的默认值为 4，但 BGP 的默认值为 1
步骤 4	**end**	返回特权 EXEC 模式

	示例： `Device(config)# end`	
步骤 5	**show ip protocols** **示例：** `Device# show ip protocols`	检查 *Maximum path* 字段的设置
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

静态单播路由

静态单播路由的相关信息

静态单播路由是用户定义的路由，让数据包从源，按照指定路径去往目的地。当路由器无法对一个目的地建立路由时，静态路由的作用就很重要，并且用户也可以使用静态路由来设置默认网关路由，也就是通过这条路由发送所有不可路由的数据包。

交换机会一直保留静态路由，直到用户删除静态路由。但用户可以通过分配管理距离值，来让交换机优选动态路由，而不是静态路由。每个静态路由协议都有一个默认的管理距离，详见表 41-16。如果用户希望使用动态路由协议提供的路由信息覆盖静态路由，就可以把动态路由协议的管理距离设置得高于静态路由的管理距离值。

表 120：动态路由协议的默认管理距离

路由源	**默认距离**
直连接口	0
静态路由	1
增强型 IGRP 汇总路由	5
外部 BGP	20
内部增强型 IGRP	90
IGRP	100
OSPF	110

内部 BGP	200
未知	255

指向接口的静态路由能够通过 RIP、IGRP 和其他动态路由协议进行通告，无论用户是否为这些路由协议静态指定了路由器配置命令 **redistribute**。这些静态路由之所以能够被通告出去，是因为指向接口的静态路由在路由表中被认为是直连的，因此也就失去了它们的“静态”属性。但如果用户指向的接口网络不是在 network 命令中定义的网络，那么动态路由协议就不会自动通告这条路由，除非用户在这些协议中静态配置了命令 **distribute**。

当一个接口失效时，所有通过这个接口发送的俄静态路由也都会从 IP 路由表中移除。当软件无法为静态路由中指定的转发路由器地址，找到有效的下一跳时，它就会把这条静态路由从 IP 路由表中移除。

配置静态单播路由

静态单播路由是用户定义的路由，让数据包从源，按照指定路径去往目的地。当路由器无法对一个目的地建立路由时，静态路由的作用就很重要，并且用户也可以使用静态路由来设置默认网关路由，也就是通过这条路由发送所有不可路由的数据包。

用户可以按照以下步骤来配置一条静态路由：

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip route prefix mask** {*address* | *interface*} [*distance*] **示例：** `Device(config)# ip route`	建立一条静态路由

	prefix mask gigabitethernet 1/0/4	
步骤 4	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 5	**show ip route** **示例：** Device# show ip route	显示路由表的当前状态，以便检查用户的配置
步骤 6	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

接下来做什么？

用户可以使用全局配置命令 **no ip route** *prefix mask* {*address*| *interface*}，来移除一条静态路由。设备会在用户移除之前一直保留静态路由。

默认路由和网络

默认路由和网络的相关信息

路由器可能无法学习到去玩个所有其他网络的路由。为了提供完整的路由功能，用户可以把一些路由器作为智能路由器，为其他路由器提供去往智能路由器的默认路由（智能路由器拥有整个互联网络的路由表信息）。用户可以让路由器动态学习这些默认路由，也可以在每台路由器上单独进行配置。大多数内部动态路由协议中都包含一种机制，能够使智能路由器生成动态的默认信息，并把它转发给其他路由器。

如果一台路由器的直连接口连接着指定的默认网络，在这个设备上运行的动态路由协议就会生成一条默认路由。在 RIP 中，它会通告伪网络 0.0.0.0。

为网络生成默认路由的那台路由器本身可能也需要一条默认路由。路由器能够为自己生成默认路由的一种方式是指定通过某台设备去往网络 0.0.0.0 的静态路由。

在使用动态路由协议传播默认信息时，用户无需执行更多的配置。系统会周期性扫描自己的路由表，来选择最优的默认路由作为它的默认路由。在 IGRP 网络中，可能会有多个候选的系统默认网络。Inspur 路由器会使用管理距离和度量值信息来设置默认路由或默认网关信息。如果没有动态信息传递到系统中，候选默认路由会由全局配置命令 **ip default-network** 进行指定。如果这个网络（来自任意源）出现在路由表中，它就会被标记为候选默认路由。如果路由器没有接口属于默认网络，但有去往默认网络的路径，那么这个网络就会成为候选网络，而去往最优默认路径的网关就会成为最后一个网关。

如何配置默认路由和网络

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**ip default-network** *network number* **示例：** `Device(config)# ip default-network 1`	指定默认网络
步骤 3	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 4	**show ip route** **示例：** `Device# show ip route`	显示设备选择的默认路由
步骤 5	**copy running-config startup-config** **示例：**	（可选）把输入的命令保存到配置文件中

	Device# **copy running-config startup-config**	

使用 route-map 重分发路由信息

route-map 的相关信息

交换机可以同时运行多种路由协议，并且能够在不同路由协议之间执行重分发。从一个路由协议向另一个路由协议重分发信息的行为适用于交换机所支持的所有 IP 路由协议。

用户也可以通过在两个路由域之间定义高级数据包过滤器或 route-map，有条件地控制路由域之间的路由重分发行为。route-map 配置命令 **match** 和 **set** 定义了 route-map 的条件部分。**match** 命令指定的必须匹配的条件。**set** 命令指定了当路由更新匹配 match 条件时执行的行为。尽管重分发是一项与协议无关的特性，但 route-map 中的一些 **match** 和 **set** 命令是与指定协议相关的。

一条 **route-map** 命令后可以配置一条或多条 **match** 命令，以及一条或多条 **set** 命令。如果用户没有配置 **match** 命令，表示所有数据包都会匹配这个条件。如果用户没有配置 **set** 命令，除了匹配行为之外，不会有任何行为发生。因此用户需要至少配置一条 **match** 或 **set** 命令。

注释： 一个 route-map 中如果没有配置 route-map 配置命令 **set** 的话，所有流量都会被发送给 CPU，从而带来高 CPU 利用率。

用户也可以用 **permit** 或 **deny** 来指定 route-map 命令。如果命令中配置了 deny，那么匹配相应规则的数据包会被发送给普通转发通道（基于目的地的路由）。如果命令中配置了 permit，那么匹配条件的数据包会被应用 set 命令中的行为。不匹配 match 条件的数据包会通过普通路由通道进行转发。

如何配置 route-map

尽管下列步骤中的步骤 3 到步骤 14 都是可选配置，用户必须在 route-map 配置模式中输入至少一个 **match** 和一个 **set** 命令。

注释： 以下步骤中的关键字与控制路由重分发使用的关键字相同。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal**	进入全局配置模式

	示例： `Device# configure terminal`	
步骤 2	**route-map** *map-tag* [**permit** | **deny**] [*sequence number*] **示例：** `Device(config)# route-map rip-to-ospf permit 4`	定义一个 route-map 用来控制重分发，并进入 route-map 配置模式 *map-tag*——为 route-map 指定一个有意义的名称。路由器配置命令 **redistribute** 会使用这个名称来调用 route-map。多个 route-map 可能会使用相同的 map 标记名称 （可选）如果用户指定了 **permit** 关键字，并且路由匹配了 route-map 中定义的匹配条件，这条路由就会按照 set 定义的行为进行重分发。如果指定了 **deny** 关键字，路由就不会被重分发 （可选）*sequence number*——这个号码指示出新 route-map 条目在相同名称的 route-map 中的位置
步骤 3	**match as-path** *path-list-number* **示例：** `Device(config-route-map)#match as-path 10`	匹配一个 BGP AS Path 访问列表
步骤 4	**match community-list** *community-list-number* [**exact**] **示例：** `Device(config-route-map)# match community-list 150`	匹配一个 BGP 团体列表
步骤 5	**match ip address** {*access-list-number* | *access-list-name*} [*...access-list-number* |	通过指定名称或编号来匹配一个标准访问列表。取值范围是 1 至 199

	...*access-list-name*] **示例：** `Device(config-route-map)# match ip address 5 80`	之间的整数
步骤 6	**match metric** metric-value **示例：** `Device(config-route-map)# match metric 2000`	匹配指定的路由度量值。*metric-value* 可以是 EIGRP 度量值，取值范围是 0 至 4294967295
步骤 7	**match ip next-hop** {*access-list-number* | *access-list-name*} [...*access-list-number* | ...*access-list-name*] **示例：** `Device(config-route-map)# match ip next-hop 8 45`	匹配一个访问列表（编号范围是 1 至 199）中指定的下一跳路由器地址
步骤 8	**match tag** *tag value* [...*tag-value*] **示例：** `Device(config-route-map)# match tag 3500`	匹配一个或多个路由标记值中的一个指定的标记值。取值范围是 0 至 4294967295 之间的整数
步骤 9	**match interface** *type number* [...*type-number*] **示例：** `Device(config-route-map)# match interface gigabitethernet 1/0/1`	匹配使用指定接口的下一跳路由
步骤 10	**match ip route-source** {*access-list-number* | *access-list-name*} [...*access-list-number* | ...*access-list-name*]	匹配指定列表通告的地址

	示例： `Device(config-route-map)# match ip route-source 10 30`	
步骤 11	**match route-type** {**local** | **internal** | **external** [**type-1** | **type-2**]} **示例：** `Device(config-route-map)# match route-type local`	匹配指定的 **route-type**： • **local**——本地生成的 BGP 路由 • **internal**——OSPF 区域内和区域间路由，或 EIGRP 内部路由 • **external**——OSPF 外部路由（类型 1 或类型 2），或 EIGRP 外部路由
步骤 12	**set dampening** *halflife reuse suppress max-suppress-time* **示例：** `Device(config-route-map)# set dampening 30 1500 10000 120`	设置 BGP 路由阻尼因子
步骤 13	**set local-preference** *value* **示例：** `Device(config-route-map)# set local-preference 100`	为本地 BGP 路径分配一个值
步骤 14	**set origin** {**igp** | **egp** *as* | **incomplete**} **示例：** `Device(config-route-map)#set origin igp`	设置 BGP 起源代码
步骤 15	**set as-path** {**tag** | **prepend** *as-path-string*} **示例：** `Device(config-route-map)# set as-path tag`	更改 BGP 自治系统路径

步骤 16	**set level** {**level-1** | **level-2** | **level-1-2** | **stub-area** | **backbone**} **示例：** `Device(config-route-map)# set level level-1-2`	为被通告到路由域中指定区域的路由，设置路由级别。关键字 **stub-area** 和 **backbone** 是指 OSPF NSSA 和骨干区域
步骤 17	**set metric** *metric value* **示例：** `Device(config-route-map)# set metric 100`	为重分发路由设置度量值（只适用于 EIGRP）*metric value* 的取值范围是 -294967295 至 294967295 之间的整数
步骤 18	**set metric***bandwidth delay reliability loading mtu* **示例：** `Device(config-route-map)# set metric 10000 10 255 1 1500`	为指定的重分发路由设置度量值（只适用于 EIGRP）： • *bandwidth*——以千比特每秒为单位的路由度量值或 IGRP 带宽，取值范围是 0 至 4294967295 • *delay*——以十倍毫秒为单位的路由延迟，取值范围是 0 至 4294967295 • *reliability*——成功传输数据包的可能性，取值范围是 0 至 255，255 表示 100%可靠，0 表示不可靠 • *loading*——路由的有效带宽，取值范围是 0 至 255（255 是 100%负载） • *mtu*——以字节尾单为的最小的最大传输单元（MTU），取值范围是 0 至 4294967295
步骤 19	**set metric-type** {**type-1** | **type-2**} **示例：**	为重分发路由设置 OSPF 外部度量类型

	Device(config-route-map)# set metric-type type-2	
步骤 20	**set metric-type internal** **示例：** Device(config-route-map)# set metric-type internal	在发布给外部 BGP 邻居的前缀上设置多出口鉴别符（MED）值，以匹配下一跳的 IGP 度量值
步骤 21	**set weight** *number* **示例：** Device(config-route-map)# set weight 100	为路由表设置 BGP 权重。取值范围是 1 至 65535
步骤 22	**end** **示例：** Device(config)# end	返回特权 EXEC 模式
步骤 23	**show route-map** **示例：** Device# show route-map	显示用户配置的所有 route-map 或指定 route-map，以此检查用户的配置
步骤 24	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

如何控制路由重分发

尽管下列步骤中的步骤 3 到步骤 14 都是可选配置，用户必须在 route-map 配置模式中输入至少一个 **match** 和一个 **set** 命令。

注释： 以下步骤中的关键字与控制路由重分发使用的关键字相同。

一个路由协议的度量值不必转换为另一个路由协议的度量值。举例来说，RIP 的度量值是跳数，IGRP 的度量值是 5 个因素的结合。在这种环境中，用户可以把一个人为设置的度量值

分配给重分发路由。如果用户不对不同路由协议之间的路由信息交换行为进行控制，可能会生成路由环路并严重降低网络性能。

如果用户没有定义默认的重分发度量值来代替度量值的转换，在路由协议之间会使用一些自动的度量值转换：

- RIP 可以自动重分发静态路由。它会为静态路由分配度量值 1（直连）；
- 如果使用默认模式的话，任意协议都可以被重分发到其他路由协议中。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**router** { **rip** | **ospf** | **eigrp**} **示例：** `Device(config)# router eigrp 10`	进入路由器配置模式
步骤 3	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** *type-value*] [**match internal** | **external** *type-value*] [**tag** *tag-value*] [**route-map** *map-tag*] [**weight** *weight*] [**subnets**] **示例：** `Device(config-router)# redistribute eigrp 1`	从一个路由协议向另一个路由协议中重分发路由。如果没有指定 route-map 的话，所有路由都会被重分发。如果指定了关键字 **route-map** 但没有指定 *map-tag* 的话，不会重分发任何路由
步骤 4	**default-metric** *number* **示例：** `Device(config-router)# default-metric 1024`	让当前路由协议为所有重分发路由（RIP 和 OSPF）使用相同的度量值
步骤 5	**default-metric** *bandwidth delay reliability*	让 EIGRP 路由协议为所有非 EIGRP 的

	loading mtu **示例：** `Device(config-router)# default-metric 1000 100 250 100 1500`	重分发路由使用相同的度量值
步骤 6	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 7	**show route-map** **示例：** `Device# show route-map`	显示用户配置的所有或指定 route-map，以此检查用户的配置
步骤 8	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

基于策略的路由

策略路由的相关信息

用户可以使用基于策略的路由（PBR）来为流量配置一个策略。通过使用 PBR，用户可以对路由施加更多的控制，并减轻对于从路由协议中获得路由的依赖。PBR 可以基于以下参数，来指定并实施路由策略：

- 识别某个终端系统
- 应用
- 协议

用户可以使用 PBR 来提供等价访问路由和与源相关的路由、基于交互式批处理流量的路由，或者基于专线的路由。举例来说，用户可以使用高带宽高开销链路上以较短的时间，向指定

办公地点传输大块记录数据，同时还通过低带宽低开销链路传输日常应用数据，比如电子邮件。

通过使用 PBR，用户可以使用访问控制列表（ACL）来对流量进行分类，然后让它们分别通过不同的路径进行转发。PBR 是应用在入站数据包上的。一个接口上如果启用了 PBR，那么这个接口上接收到的所有数据包都需要通过 route-map 的检查。根据 route-map 中定义的规则，数据包会被转发（路由）给特定的下一跳。

- route-map 中被标记为 permit 的流量会：
  - match 命令可以用来匹配长度或多个 ACL。一个 route-map 中可以包含多个 match 命令。对所有 match 命令运行逻辑或算法公式，来满足 permit 或 deny 条件。

    举例来说：

    match length A B

    match ip address acl1 acl2

    match ip address acl3
- 如果数据包匹配 match length A B、acl1、acl2 或 acl3，就放行数据包
  - 如果决策为 permit，那么有 set 命令定义的行为就会被应用到数据包上；
  - 如果据测为 deny，那么 PBR 行为（set 命令指定的行为）就不会应用。相反，处理逻辑会向前移动，以查看序列中的下一个路由映射语句（下一个较大的序列号对应的语句）。如果没有下一个语句了，PBR 处理就结束了，数据包会使用默认 IP 路由表进行路由。
- 对于 PBR 来说，不支持标记为 deny 的 route-map 命令

用户可以使用标准 IP ACL 来指定匹配源地址的匹配条件，或者使用扩展 IP ACL 来根据应用、协议类型或终端工作站进行匹配。处理行为会持续到在 route-map 中找到匹配项为止。如果没有找到匹配项，交换机就会使用普通的基于目的地的路由来转发数据包。在匹配条件的末尾有隐含的 deny 语句。

如果数据包满足了匹配条件，用户可以使用 set 命令来指定 IP 地址，也就是路径中的下一跳路由器地址。

有关 PBR 命令和关键字的更多信息用户可以参考 *Inspur INOS IP Command Reference, Volume 2 of 3: Routing Protocols*。

如何配置 PBR

- 要想使用 PBR，用户必须在交换机或堆栈主用设备上启用 IP Base 特性集；
- 组播流量不能基于策略进行路由。PBR 只能应用在单播流量上；

- 用户可以在路由端口或 SVI 接口上启用 PBR；
- 交换机能够支持 PBR 来匹配长度；
- 用户可以在三层模式的 EtherChannel 上应用策略 route-map，但不能在属于 EtherChannel 成员的物理接口上应用策略 route-map。如果用户尝试这样做的话，命令会被系统拒绝。在物理接口上应用策略 route-map 后，接口就无法称为一个 EtherChannel 的成员了；
- 用户可以在交换机或交换机堆栈上定义最多 128 个 IP 策略 route-map；
- 用户可以在交换机或交换机堆栈上为 PBR 定义最多 512 个访问控制条目（ACE）；
- 在 route-map 中配置匹配条件时，用户可以遵循以下指导：
  - 不要匹配允许去往本地地址的 ACL；
- VRF 和 PBR 的配置在交换机接口上是相互排斥的。用户不能在启用了 PBR 的接口上启用 VRF。反之亦然，用户也不能在启用了 VRF 的接口上启用 PBR；
- Web 缓存通信协议（WCCP）和 PBR 的配置在交换机接口上是相互排斥的。用户不能在启用了 PBR 的接口上启用 WCCP。反之亦然，用户也不能在启用了 WCCP 的接口上启用 PBR；
- PBR 能够使用的硬件条目数量取决于 route-map 本身，如果使用了 ACL 的话，也取决月 ACL 和 route-map 条目的顺序；
- PBR 基于 TOS 进行匹配，不支持 DSCP 和 IP 优先级；
- 不支持设置接口、设置默认下一跳和设置默认接口；
- 不支持使用 **ip next-hop recursive** 和 **ip next-hop verify availability** 特性，下一跳应该是直连地址；
- 支持使用没有设置 set 行为的 policy-map，与之相匹配的数据包会如常路由；
- 支持使用没有设置 match 条件的 policy-map，其中设置的行为会应用在所有数据包上。

默认情况下，交换机上是禁用 PBR 的。要想启用 PBR，用户必须创建一个 route-map，并在其中指定匹配条件和相应的行为。接着，用户必须在接口上通过应用 route-map 来启用 PBR。这个接口接收到的所有数据包都会与 PBR 中的匹配条件进行匹配。

由交换机生成的数据包，或者本地数据包通常不使用策略路由。当用户在交换机上全局启用本地 PBR 时，所有由交换机生成的数据包会受到本地 PBR 限制。本地 PBR 默认是禁用的。

总步骤

1. configure terminal

2. route-map *map-tag* [**permit**] [*sequence number*]

3. match ip address {*access-list-number* | *access-list-name*} [*access-list-number* |...*access-list-name*]

4. match length min max

5. set ip next-hop *ip-address* [*...ip-address*]

6. exit

7. interface *interface-id*

8. ip policy route-map *map-tag*

9. ip route-cache policy

10. exit

11. ip local policy route-map *map-tag*

12. end

13. show route-map [*map-name*]

14. show ip policy

15. show ip local policy

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**route-map** *map-tag* [**permit**] [*sequence number*] **示例：** `Device(config)# route-map pbr-map permit`	定义 route-map，用它来控制从哪里发出数据包，并进入 route-map 配置模式。 • *map-tag*——为 route-map 定义一个有意义的名称。接口配置命令 **ip policy route-map** 会使用这个名称来调用 route-map。多条使用相同 map 标记的 route-map 语句定义了一个 route-map • （可选）**permit**——如果用户指定了 **permit**，并且匹配条件与这个 route-map 相符，路由就会按照 set 指定的行为对数据包进行策略路由 • （可选）*sequence number*——序列号指的是在这个 route-map 中，

		route-map 语句的位置
步骤 3	**match ip address** {*access-list-number* | *access-list-name*} [*access-list-number* |*...access-list-name*] **示例：** `Device(config-route-map)# match ip address 110 140`	匹配由一个或多个标准或扩展访问列表放行的源和目的 IP 地址。ACL 中可以匹配一个或多个源和目的 IP 地址。 如果用户没有指定 **match** 命令的话，route-map 会应用给所有数据包
步骤 4	**match length min max** **示例：** `Device(config-route-map)# match length 64 1500`	匹配数据包长度
步骤 5	**set ip next-hop** *ip-address* [*...ip-address*] **示例：** `Device(config-route-map)# set ip next-hop 10.1.6.2`	为匹配规则的数据包设置需要采取的行为。设置用来路由数据包的下一跳（下一跳必须是邻接地址）
步骤 6	**exit** **示例：** `Device(config-route-map)# exit`	返回全局配置模式
步骤 7	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet 1/0/1`	进入接口配置模式，并指定用户想要配置的接口
步骤 8	**ip policy route-map** *map-tag* **示例：** `Device(config-if)# ip policy`	在三层接口上启用 PBR，并指定要使用的 route-map。用户可以在一个接口上只配置一个 route-map。但用户可以配置多个拥有不同序列号的 route-map 条

	route-map pbr-map	目。数据包会按照序列号的顺序进行匹配，直到遇到第一个匹配规则。如果没有匹配的话，数据包会如常路由
步骤 9	**ip route-cache policy** **示例：** Device(config-if)# ip route-cache policy	（可选）启用快速交换 PBR。用户必须在启用快速交换 PBR 之前，先启用 PBR
步骤 10	**exit** **示例：** Device(config-if)# exit	返回全局配置模式
步骤 11	**ip local policy route-map** *map-tag* **示例：** Device(config)# ip local policy route-map local-pbr	（可选）启用本地 PBR，为交换机生成的数据包执行基于策略的路由。这些策略是应用于交换机生成的数据包的，而不是应用于入站数据包
步骤 12	**end** **示例：** Device(config)# end	返回特权 EXEC 模式
步骤 13	**show route-map** [*map-name*] **示例：** Device# show route-map	（可选）显示用户配置的所有或指定 route-map，以此来检查配置
步骤 14	**show ip policy** **示例：** Device# show ip policy	（可选）显示与接口相关联的策略 route-map
步骤 15	**show ip local policy** **示例：**	（可选）显示是否启用了本地策略路由，如果启用了，还会显示用户使用的 route-map

	Device# show ip local policy	

过滤路由信息

用户可以通过执行这部分描述的任务，来过滤路由协议信息。

注释： 当用户在多个 OSPF 进程之间执行重分发的话，OSPF 度量值并不会保留。

设置被动接口

为了防止本地网络上的其他路由器动态学习路由，用户可以使用路由器配置命令 **passive-interface**，禁止一个路由器接口向外发送路由更新消息。当用户在 OSPF 协议中使用这条命令时，用户指定称为被动模式的接口地址会在 OSPF 域中表现为一个末节网络。并且设备不会通过这个路由器接口来发送或接收 OSPF 路由信息。

在拥有多个接口的网络中，如用用户不希望手动设置被动接口的话，可以使用路由器配置命令 **passive-interface default**，把所有接口默认设置为被动模式，并手动在相应的接口上启用邻接关系。

用户可以使用特权 EXEC 模式的网络监控命令来检查用户是否把接口设置为被动接口，比如 **show ip ospf interface**，或者使用特权 EXEC 命令 **show ip interface** 来检查用户是否把接口设置为主动接口。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** Device# configure terminal	进入全局配置模式
步骤 2	**router** { **rip** | **ospf** | **eigrp**} **示例：** Device(config)# router ospf	进入路由器配置模式
步骤 3	**passive-interface** *interface-id* **示例：**	禁止从指定的三层接口向外发送路由更新

	Device(config-router)# passive-interface gigabitethernet 1/0/1	
步骤 4	**passive-interface default** **示例：** Device(config-router)# passive-interface default	（可选）设置所有接口的默认模式为被动接口
步骤 5	**no passive-interface** *interface type* **示例：** Device(config-router)# no passive-interface gigabitethernet1/0/3 gigabitethernet 1/0/5	（可选）只把这些需要建立邻接关系的接口设置为主动接口
步骤 6	**network** *network-address* **示例：** Device(config-router)# network 10.1.1.1	（可选）为路由进程指定一个网络列表。*network-address* 部分为 IP 地址
步骤 7	**end** **示例：** Device(config-router)# end	返回特权 EXEC 模式
步骤 8	**copy running-config startup-config** **示例：** Device# copy running-config startup-config	（可选）把输入的命令保存到配置文件中

控制路由更新中的通告和处理

用户可以使用路由器配置命令 **distribute-list** 和访问控制列表，来阻止本地路由器在路由更

新中通告一些路由，或者阻止其他路由器学习到一条或多条路由。在 OSPF 中使用这个特性时，这个特性只应用与外部路由，用户不能指定接口名称。

用户可以使用路由器配置命令 **distribute-list**，避免在入站更新中处理相应的路由（这个特性不适用于 OSPF）。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**router** { **rip** | **eigrp**} **示例：** `Device(config)# router eigrp 10`	进入路由器配置模式
步骤 3	**distribute-list** {*access-list-number* | *access-list-name*} **out** [*interface-name* | *routing process* | *autonomous-system-number*] **示例：** `Device(config-router)# distribute 120 out gigabitethernet 1/0/7`	允许或拒绝在路由更新中通告特定的路由，根据访问列表中列出的行为做出判断
步骤 4	**distribute-list** {*access-list-number* | *access-list-name*} **in** [*type-number*] **示例：** `Device(config-router)# distribute-list 125 in`	禁止处理更新中列出的路由
步骤 5	**end**	返回特权 EXEC 模式

	示例： `Device(config-router)# end`	
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

过滤路由信息的源

由于有些路由信息可能会比其他路由信息更精确，因此用户可以使用过滤机制优选来自于不同源的信息。管理距离是用来评估路由信息源（比如一台路由器或一组路由器）信任程度的度量值。在一个大型网络中，有些路由信息会比另一些路由信息更可靠。通过指定管理距离值，用户可以让路由器智能地在路由信息源之间进行判断。路由器总是会选择管理距离最低的路由协议所提供的路由信息。

由于每个网络的需求不同，因此在分配管理距离的方面并没有统一的指导方案。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**router** { **rip** | **eigrp**} **示例：** `Device(config)# router eigrp 10`	进入路由器配置模式
步骤 3	**distance** *weight* {*ip-address* {*ip-address mask*}} [*ip access list*] **示例：** `Device(config-router)#`	定义一个管理距离。 *weight*——管理距离的取值是 10 至 255 之间的整数。如果单独使用的话，*weight* 指定的是默认管理距离，如果路由信息源没有其他指定的管理距离时，就使用

	distance 50 10.1.5.1	这个值。距离为 255 的路由不会被放入到路由表中 （可选）*ip access list*——应用于入站路由更新的 IP 标准或扩展访问列表
步骤 4	**end** **示例：** Device(config-router)# end	返回特权 EXEC 模式
步骤 5	**show ip protocols** **示例：** Device# show ip protocols	显示指定路由进程的默认管理距离
步骤 6	**copy running-config startup-config** **示例：** Device# copy running-config startup-config	（可选）把输入的命令保存到配置文件中

管理认证密钥

密钥管理特性是路由协议用来控制认证密钥的一种方法。并不是所有协议都可以使用密钥管理特性。EIGRP 和 RIP 版本 2 能够使用密钥管理特性。

先决条件

在管理认证密钥之前，用户必须先启用认证功能。用户可以参考相应协议内容，来查看如何为指定协议启用认证功能。为了管理认证密钥，用户需要定义一个密钥链、指定属于这个密钥链的密钥，并指明每个密钥的有效长度。每个密钥都有自己的密钥识别符（使用密钥链配置命令 **key** *number* 进行指定），这是储存在设备本地的信息。密钥标识符和，以及与消息相关联的接口，组成了唯一标识设备使用的认证算法和消息摘要 5（MD5）认证密钥。

如何配置认证密钥

用户可以使用生存时间来配置多个密钥。尽管设备上有多个有效密钥，但设备只会发送一个认证数据包。软件会按照从低到高的顺序检查密钥编号，并使用第一个有效的密钥。在密钥交换过程中也会交换生存时间，用户要确保路由器知道这些生存时间。

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**key chain** *name-of-chain* **示例：** `Device(config)# key chain key10`	指定一个密钥链，并进入密钥链配置模式
步骤 3	**key** *number* **示例：** `Device(config-keychain)# key 2000`	指定密钥编号。取值范围是 0 至 2147483647
步骤 4	**key-string** *text* **示例：** `Device(config-keychain)# Room 20, 10th floor`	标识密钥字符串。字符串中可以包含 1 至 80 个区分大小写的字母和数字字符，但第一个字符不能是数字
步骤 5	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **示例：** `Device(config-keychain)# accept-lifetime 12:30:00 Jan`	（可选）指定能够使用这个密钥的时间周期。 *start-time* 和 *end-time* 可以是 *hh:mm:ss Month date year* 或 *hh:mm:ss date Month year*。默认值是永远和默认的 *start-time*，也就是最早可以接受的日

	25 1009 infinite	期，比如 1993 年 1 月 1 日。默认的 *end-time* 和 **duration** 是 **infinite**
步骤 6	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **示例：** Device(config-keychain)# accept-lifetime 23:30:00 Jan 25 1019 infinite	（可选）指定能够发送这个密钥的时间周期。 *start-time* 和 *end-time* 可以是 *hh:mm:ss Month date year* 或 *hh:mm:ss date Month year*。默认值是永远和默认的 *start-time*，也就是最早可以接受的日期，比如 1993 年 1 月 1 日。默认的 *end-time* 和 **duration** 是 **infinite**
步骤 7	**end** **示例：** Device(config)# end	返回特权 EXEC 模式
步骤 8	**show key chain** **示例：** Device# show key chain	显示认证的密钥信息
步骤 9	**copy running-config startup-config** **示例：** Device# copy running-config startup-config	（可选）把输入的命令保存到配置文件中

监控和维护 IP 网络

用户可以移除指定缓存、表或数据库中的所有内容。用户也可以显示指定的状态统计信息。

表 121：清除 IP 路由或显示路由状态的命令

命令	目的
clear ip route {*network* [*mask* | ***]}	从 IP 路由表中清除一条或多条路由
show ip protocols	显示活跃路由进程的参数和状态

show ip route [*address* [*mask*] [**longer-prefixes**]] | [*protocol* [*process-id*]]	显示路由表的当前状态
show ip route summary	以汇总的形式显示路由表的当前状态
show ip route supernets-only	显示超网信息
show ip cache	显示用来交换 IP 流量的路由表
show route-map [*map-name*]	显示用户配置的所有或指定 route-map

第 13 部分 安全

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

预防未授权访问

用户可以通过配置本地交换机，以及查看配置信息，来预防未授权的用户访问。通常情况下，用户会希望网络管理员能够访问交换机，同时防止网络外的用户通过异步端口拨入到交换机中，或者通过串行端口从网络外连接到交换机，或者通过本地网络中的终端或工作站连接到交换机。

要想预防交换机接受未授权的访问，用户应该配置以下安全特性之一：

- 最低限度，用户应该为每个交换机端口配置密码和特权级别。这些密码是储存在交换机本地的。当用户尝试通过一个端口或线路访问交换机时，他/她们必须输入这个端口或线路上指定的密码，才能获得交换机的访问权限；
- 为了实施更高一层的安全性，用户也可以配置用户名和密码对，这些信息也是储存在交换机本地的。用户可以把这些信息对分配给线路或端口，并在用户访问交换机之前对其

进行认证。如果用户定义了特权级别，还能够为每个用户名和密码对分配特定的特权级别（关联着权利和特权）；

- 如果用户希望使用用户名和密码对，但希望集中把这些信息储存在服务器上，而不是保存在交换机本地，用户可以把它们储存到安全服务器的数据库中。之后多种网络设备都可以使用相同的数据库来获得用户认证（如果需要的话，还可以获得授权信息）信息；
- 用户还可以启用登录高级特性，它会记录失败的和未成功的登录尝试。登录高级特性也可以用来在用户进行了一定次数的未成功尝试后，阻止它未来一段时间的登录尝试。更多信息用户可以参考 Inspur INOS Login Enhancements 文档。

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

使用密码和特权来控制交换机访问的限制条件

使用密码和特权来控制交换机访问具有以下限制条件：

- 如果用户使用全局配置命令 **boot manual** 手动启动交换机的话，禁用密码发现功能将不会生效。这条命令会在交换机重新加电后，进入引导加载程序（*switch:*）；

默认的密码和特权级别配置

在用户网络中提供终端访问控制的一种简单的方法是使用密码并分配特权级别。用户可以使用密码保护来对网络或网络设备的访问实施限制。特权级别定义了用户在登录到网络设备后，能够使用的命令。

下面这个表格中展示了默认的密码和特权级别配置。

表 122：默认的密码和特权级别

特性	**默认设置**
启用密码和特权级别	未定义密码。默认级别是级别 15（特权 EXEC 级别）。密码在配置文件中是未加密的
启用秘密密码和特权级别	未定义密码。默认级别是级别 15（特权 EXEC 级别）。密码在配置文件中是加密的
线路密码	未定义密码

其他密码安全特性

要想提供更高层次的安全性，尤其是涉及在网络中传输的密码，以及储存在简单文件传输协议（TFTP）服务器上的密码，用户可以使用全局配置命令 **enable password** 或 **enable secret**。这两条命令都提供了相同的功能：也就是用户必须输入一个加密密码，才能访问特权 EXEC 模式（默认）或任意特权级别。

我们建议用户使用 **enable secret** 命令，因为它使用了增强的加密算法。

如果用户配置了 **enable secret** 命令，这条命令的优先级会高于 **enable password** 命令；这两条命令不能同时生效。

如果用户启用了密码加密特性，这个特性会应用在所有密码上，其中包括用户密码、加密密钥密码、特权命令密码，以及控制和虚拟终端线路密码。

密码恢复

默认情况下，任意终端用户通过物理的方式介入到交换机上，都可以通过在交换机加电过程

中，打断启动进程并输入新密码，来恢复交换机密码。

密码恢复禁用特性能够通过禁用这个功能中的一部分，保护他人对交换机的访问。当用户启用了这个特性时，终端用户只有同意把系统恢复为默认配置，才可以打断启动进程。在密码恢复禁用后，用户仍可以打断启动进程并更改密码，但交换机的配置文件（config.text）和 VLAN 数据库文件（vlan.dat）都会被删除。

如果用户禁用了密码恢复特性，我们建议用户在一台安全服务器上保留配置文件的副本，以防终端用户打断了启动进程，而使系统恢复默认配置。不要在交换机上备份配置文件的副本。如果交换机运行在 VTP 透明模式的话，我们建议用户也要在安全服务器上备份 VLAN 数据库文件的副本。当交换机恢复默认系统配置后，用户可以使用 Xmodem 协议，把这些保存的文件下载到交换机上。

要想重新启用密码恢复特性，用户需要使用全局配置命令 **service password-recovery**。

终端线路 Telnet 配置

当用户第一次启动交换机时，可以使用一个自动设置程序，来分配 IP 地址并创建后续使用的默认配置。设置程序也会提示用户为通过 Telnet 访问交换机这种方式配置一个密码。如果用户没有在设置程序中配置这个密码，也可以在设置终端线路时设置 Telnet 密码。

用户名密码对

用户可以配置用户名和密码对，这些是储存在交换机本地的信息。用户可以把这些信息对分配给线路或端口，并在用户访问交换机之前对其进行认证。如果用户定义了特权级别，还能够为每个用户名和密码对分配特定的特权级别（关联着权利和特权）。

特权级别

Inspur 交换机（和其他设备）能够使用特权级别来为不同的交换机操作级别提供密码保护机制。默认情况下，Inspur INOS 软件运行在两种密码安全模式（特权级别）中：用户 EXEC（级别 1）和特权 EXEC（级别 15）。用户可以为每个模式配置 16 个命令层级。通过配置多个密码，用户可以允许不同的用户集合访问指定的命令。

线路上的特权级别

用户可以通过登录到线路中并启用不同的特权级别，来覆盖使用线路配置命令 **privilege level** 设置的特权级别。用户可以通过使用 **disable** 命令来降低特权级别。如果用户知道更高特权级别的密码，也可以使用密码来启用更高的特权级别。用户可能会为 Console 线路指定高级别或特权级别的访问权限，以此来限制对线路的使用。

举例来说，如果用户希望多个用户使用 **clear line** 命令，可以为其分配级别 2 安全等级，并广泛分发级别 2 密码。如果用户希望对 **configure** 命令实施更严格的访问限制，可以为其分配级别 3 安全等级，并把级别 3 密码严格限制在一组用户范围内。

命令特权级别

当用户把一条命令设置为特权级别时，如果这条命令的语法属于某个命令子集，则相应的命令也都会被设置为这个特权级别。举例来说，如果用户把 **show ip traffic** 命令设置为级别 15，那么 **show** 命令和 **show ip** 命令也会自动被设置为级别 15，除非用户分别把它们设置为不同的级别。

如何使用密码和特权级别来控制交换机访问

设置或更改静态 enable 密码

enable 密码用来控制用户访问特权 EXEC 模式的行为。用户可以按照以下步骤来设置或更改静态 enable 密码。

总步骤

1. enable

2. configure terminal

3. enable password *password*

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的

步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**enable password** *password* **示例：** `Device(config)# enable password secret321`	定义一个新密码或更改现有密码，来访问特权 EXEC 模式。 默认情况下没有定义密码。 在 *password* 部分指定一个长度为 1 至 25 的字母和数字字符串。字符串不能以数字开头、需要区分大小写，并且可以使用空格，但会忽略开头的空格。用户可以使用问号（?），但需要在问号前输入 Ctrl-v；举例来说，要想创建密码 abc?123，用户需要这样输入： **1** 输入 **abc** **2** 输入 **Ctrl-v** **3** 输入**?123** 在系统提示用户输入 enable 密码时，用户无需在问号前输入 Ctrl-v，只要在密码提示符后面输入 abc?123 就可以
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息

步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

使用加密特性保护 enable 密码和 enable 秘密密码

用户可以按照以下步骤建立一个加密密码，也就是要想进入特权 EXEC 模式（默认）或任何指定的特权级别，就必须输入这个密码：

总步骤：

1. enable

2. configure terminal

3. Use one of the following:

• **enable password** [**level** *level*]

{*password* | *encryption-type encrypted-password*}

• **enable secret** [**level** *level*]

{*password* | *encryption-type encrypted-password*}

4. service password-encryption

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 3	用户可以使用以下命令之一： • **enable password** [**level** *level*] {*password* | *encryption-type encrypted-password*} • **enable secret** [**level** *level*] {*password* | *encryption-type encrypted-password*} **示例：** `Device(config)#` **`enable password example102`** 或者 `Device(config)#` **`enable secret level 1 password secret123sample`**	• 定义一个新密码或更改一个已有密码，来访问特权 EXEC 模式。 • 定义一个秘密密码，这个密码是使用不可逆的加密模式保存在交换机中的。 · （可选）*level* 字段的取值范围是 0 至 15。级别 1 是通常用于用户 EXEC 模式的特权。默认级别是 15（特权 EXEC 模式的特权） · 在 *password* 部分指定一个长度为 1 至 25 的字母和数字字符串。字符串不能以数字开头、需要区分大小写，并且可以使用空格，但会忽略开头的空格。默认没有指定密码 · （可选）*encryption-type* 只有类型 5 可用，这是 Inspur 私有加密算法。如果用户指定了加密类型，就必须提供加密密码——从其他交换机配置中复制的加密密码 **注释：** 如果用户指定了加密类型，并输入了一个明文密码，就无法再次进入特权 EXEC 模式了。用户无法使用任何方式恢复丢失的加密密码
步骤 4	**service password-encryption** **示例：** `Device(config)#` **`service password-encryption`**	（可选）在定义密码是或重写配置时加密这个密码。 加密特性能够防止他人在配置中读取密码
步骤 5	**end**	返回特权 EXEC 模式

	示例： `Device(config)# end`	
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

禁用密码恢复

用户可以按照以下步骤来禁用密码恢复特性，以此保护交换机的安全：

在开始前

如果用户禁用了密码恢复特性，我们建议用户在一台安全服务器上保留配置文件的副本，以防终端用户打断了启动进程，而使系统恢复默认配置。不要在交换机上备份配置文件的副本。如果交换机运行在 VTP 透明模式的话，我们建议用户也要在安全服务器上备份 VLAN 数据库文件的副本。当交换机恢复默认系统配置后，用户可以使用 Xmodem 协议，把这些保存的文件下载到交换机上。

总步骤

1. enable

2. configure terminal

3. system disable password recovery switch {*all* | *<1-9>*}

4. end

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：**	进入特权 EXEC 模式。在提示时输入密码

	Device> **enable**	
步骤 2	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式
步骤 3	**system disable password recovery switch** {*all* | *<1-9>*} **示例：** Device(config)# **system disable password recovery switch all**	禁用密码恢复特性。 • *all*——设置堆栈中交换机上的配置 • *<1-9>*——设置所选交换机编号上的配置 这个设置保存在 Flash 中可由引导加载程序和浪潮 INOS 映像访问的区域中，但它不是文件系统的一部分，任何用户都无法访问
步骤 4	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式

接下来做什么？

要想删除 **disable password recovery** 命令，用户需要使用全局配置命令 **no system disable password recovery switch all**。

为终端线路设置 Telnet 密码

从用户 EXEC 模式开始，用户可以按照以下步骤为直连的终端线路配置 Telnet 密码：

在开始前

- 把安装有模拟软件的 PC 或工作站连接到交换机的 Console 端口，或把 PC 连接到以太网管理端口；
- Console 端口的默认数据特征是 9600、8、1、无隐私。用户可能需要多次按下回车键才能够看到命令行提示符。

总步骤

1. enable

2. configure terminal

3. line vty 0 15

4. password *password*

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	**注释：** 如果访问特权 EXEC 模式需要密码，系统会向用户进行提示。 进入特权 EXEC 模式
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**line vty 0 15** **示例：** `Device(config)# line vty 0 15`	配置 Telnet 会话（线路）的编号，并进入线路配置模式。 在支持命令的设备上共有 16 条会话。0 和 15 表示用户要配置所有 16 条 Telnet 会话
步骤 4	**password** *password* **示例：** `Device(config-line)# password abcxyz543`	为一条或多条线路设置 Telnet 密码。 在 *password* 部分指定一个长度为 1 至 25 的字母和数字字符串。字符串不能以数字开头、需要区分大小写，并且可以使用空格，但会忽略开头的空格。默认没有指定密码
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config**	检查用户输入的信息

	示例： `Device# show running-config`	
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置用户名和密码对

用户可以按照以下步骤来配置用户名和密码对：

总步骤

1. enable

2. configure terminal

3. username *name* [**privilege** *level*] {**password** *encryption-type password*}

4. 使用以下命令之一：

• line console 0

• line vty 0 15

5. login local

6. end

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 3	**username** *name* [**privilege** *level*] {**password** *encryption-type password*}
	
	示例：
	`Device(config)# username adamsample privilege 1 password secret456`
	`Device(config)# username 111111111111 mac attribute`
		为每个用户设置用户名、特权级别和密码。
		• 在 *name* 部分把用户 ID 设置为一个单词或 MAC 地址。不能使用空格和问号
		• 用户最多可以为用户名和 MAC 过滤器配置 12000 个客户端
		• （可选）*level* 字段定义的是用户获得访问权限后，能够使用的特权级别。取值范围是 0 至 15。级别 15 是特权 EXEC 模式的特权。级别 1 是用户 EXEC 模式的特权
		• 在 *encryption-type* 部分输入 0 指定未加密密码，输入 7 指定隐藏密码
		• （可选）*password* 部分指定的是用户必须用来获得设备访问权限的密码。密码长度必须为 1 至 25 个字符，其中可以包含空格，并且必须是 **username** 命令中指定的最后一个选项

步骤 4	用户可以使用以下命令之一：
	• **line console 0**
	• **line vty 0 15**
	
	示例：
	`Device(config)# line console 0`
	或者
	`Device(config)# line vty 15`
		进入线路配置模式，并配置 Console 端口（线路 0）或 VTY 线路（线路 0 至 15）

步骤 5	**login local** **示例：**	在登录时启用密码检查。基于步骤 3 中指定的用户名进行认证

	Device(config-line)# **login local**	
步骤 6	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 7	**show running-config** **示例：** Device# **show running-config**	检查用户输入的信息
步骤 8	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

为一条命令设置特权级别

用户可以使用以下命令来为一条命令设置特权级别：

总步骤

1. enable

2. configure terminal

3. privilege *mode* **level** *level command*

4. enable password level *level password*

5. end

6. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** Device> **enable**	进入特权 EXEC 模式。在提示时输入密码

步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**privilege** *mode* **level** *level command* **示例：** `Device(config)# privilege exec level 14 configure`	为一条命令设置特权级别： • 在 *mode* 部分输入 **configure** 指定全局配置模式，**exec** 指定 EXEC 模式，**interface** 指定接口配置模式，或者 **line** 指定线路配置模式 • 在 *level* 部分指定 0 至 15 之间的值。级别 1 表示普通用户 EXEC 模式的特权，级别 15 表示需要使用 **enable** 密码进入的特权模式 • 在 *command* 部分指定用户想要限制访问的命令
步骤 4	**enable password level** *level password* **示例：** `Device(config)# enable password level 14 SecretPswd14`	指定启用某个特权级别的密码。 • 在 *level* 部分指定 0 至 15 之间的值。级别 1 表示普通用户 EXEC 模式的特权 • 在 *password* 部分指定一个长度为 1 至 25 的字母和数字字符串。字符串不能以数字开头、需要区分大小写，并且可以使用空格，但会忽略开头的空格。默认没有指定密码
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config`	（可选）把输入的命令保存到配置文件中

	startup-config	

为线路改变默认的特权级别

用户可以按照以下步骤为指定线路更改默认的特权级别：

总步骤

1. enable

2. configure terminal

3. line vty *line*

4. privilege level *level*

5. end

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**line vty** *line* **示例：** `Device(config)# line vty 10`	选择用户想要限制访问的虚拟终端线路
步骤 4	**privilege level** *level* **示例：** `Device(config)# privilege level 15`	为线路更改默认的特权级别。 在 *level* 部分指定 0 至 15 之间的值。级别 1 表示普通用户 EXEC 模式的特权，级别 15 表示需要使用 **enable** 密码进入的特权模式
步骤 5	**end**	返回特权 EXEC 模式

	示例： `Device(config)# end`	
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

接下来做什么？

用户可以通过登录到线路中并启用不同的特权级别，来覆盖使用线路配置命令 **privilege level** 设置的特权级别。用户可以通过使用 **disable** 命令来降低特权级别。如果用户知道更高特权级别的密码，也可以使用密码来启用更高的特权级别。用户可能会为 Console 线路指定高级别或特权级别的访问权限，以此来限制对线路的使用。

登录和离开一个特权级别

从用户 EXEC 模式开始，用户可以按照以下步骤登录指定的特权级别，以及离开指定的特权级别。

总步骤

1. enable *level*

2. disable *level*

具体步骤

	命令或操作	**目的**
步骤 1	**enable** *level* **示例：** `Device> enable 15`	登录到指定特权级别中。 示例中 15 表示特权 EXEC 模式。 在 *level* 部分输入 0 至 15 之间的值
步骤 2	**disable** *level* **示例：** `Device# disable 1`	离开指定特权级别

监控交换机的访问

表 123：显示 DHCP 信息的命令

命令	目的
show privilege	显示特权级别的命令

设置密码和特权级别的配置示例

示例：设置或更改静态 enable 密码

这个示例展示了如何把 enable 密码更改为 *l1u2c3k4y5*。这个密码不加密，提供级别 15 的访问特权（传统特权 EXEC 模式的访问）：

```
Device(config)# enable password l1u2c3k4y5
```

示例：使用加密特性保护 enable 密码和 enable 秘密密码

这个示例展示了如何为特权级别 2 配置加密密码*1FaD0$Xyti5Rkls3LoyxzS8*：

```
Device(config)# enable secret level 2 5 $1$FaD0$Xyti5Rkls3LoyxzS8
```

示例：为终端线路设置 Telnet 密码

这个示例展示了如何把 Telnet 密码设置为 *let45me67in89*：

```
Device(config)# line vty 10
Device(config-line)# password let45me67in89
```

示例：为一条命令设置特权级别

这个示例展示了如何使用 **configure** 命令设置特权级别 14，并把用户用来进入特权级别 14 的密码设置为 *SecretPswd14*：

```
Device(config)# privilege exec level 14 configure
```

```
Device(config)# enable password level 14 SecretPswd14
```

其他参考资料

错误消息解码器

描述	链接
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	http://www.icntnetworks.com

配置 TACACS+

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置 TACACS+的先决条件

在使用 TACACS+来设置和配置交换机访问时，有以下先决条件（必须按顺序执行）：

1. 在交换机上配置 TACACS+服务器的地址；
2. 设置一个认证密钥；
3. 在 TACACS+服务器上配置步骤 2 指定的密钥；
4. 启用认证、授权和审计（AAA）；
5. 创建一个登录认证方式列表；
6. 在终端线路上应用列表；
7. 创建授权和审计方式列表。

在使用 TACACS+来控制交换机访问时，有以下先决条件：

- 用户必须有配置 TACACS+服务器的能力，才能在交换机上配置 TACACS+特性。并且用户必须能够访问（通常运行在 LINUX 或 Windows 工作站上）TACACS+守护程序上的数据库中维护的 TACACS+服务；
- 我们建议在交换机堆栈和 TACACS+服务器之间建立冗余连接。这是为了确保在交换机堆栈中的某个堆栈成员被移除后，堆栈仍能够访问 TACACS+服务器；
- 用户需要运行 TACACS+守护程序的系统，才能在交换机上使用 TACACS+；
- 要想使用 TACACS+，用户必须启用它；
- 用户必须启用授权，交换机才能使用授权；

- 用户必须首先成功完成 TACACS+认证，才能执行 TACACS+授权；
- 要想使用这部分或其他文档中列出的 AAA 命令，用户必须首先使用 **aaa new-model** 命令启用 AAA；
- 最起码用户必须对维护 TACACS+守护程序的一台或多台主机进行标识，并定义 TACACS+认证的方法列表。用户可以（可选的）定义 TACACS+授权和审计的方法列表；
- 方法列表中定义了要执行的认证的类型，以及执行它们的顺序；在它能够执行任何定义的身份验证方法之前，用户必须把它应用在特定的端口上。唯一的例外是默认方法列表（巧合的是，它就名为 ***default***）。默认方法列表会自动应用在所有端口上，除了已经明确定义了方法列表名称的端口。用户定义的方法列表会覆盖默认方法列表；
- 如用用户使用 TACACS+执行认证，也可以使用 TACACS+执行特权 EXEC 访问的授权；
- 如果没有使用 TACACS+执行认证，可以使用本地数据库执行认证。

相关主题

TACACS+的相关信息

TACACS+和交换机访问

这部分描述了 TACACS+。TACACS+提供了详细的审计信息，以及对认证和授权过程的灵活管理控制。它能够提供认证、授权、审计（AAA），并且只能通过 AAA 命令进行启用。

TACACS+概述

TACACS+是一项安全应用，以集中的方式对尝试获得交换机访问权限的用户进行认证。

TACACS+提供了分离和模块化的认证、授权和审计功能。TACACS+允许使用每台访问控制服务器（TACACS+守护程序）来独立地提供每一项服务——认证、授权和审计。每台服务器可以与自己的数据库相绑定，并根据守护程序的功能，利用服务器自身的服务，或网络上可用的其他服务。

TACACS+的目标是提供一种方法，用单个管理服务来管理多个网络接入点。用户交换机可以是网络访问服务器，或者其他 Inspur 路由器和访问服务器。

图 98：典型的 TACACS+网络配置

UNIX workstation	UNIX 工作站

(TACACS+ server 1)	（TACACS+ 服务器 1）
Catalyst 6500 series switch	Inspur 6500 系列交换机
UNIX workstation (TACACS+ server 2)	UNIX 工作站 （TACACS+ 服务器 2）
Workstations（共 2 处）	工作站
Configure the switches with the TACACS+ server addresses. Set an authentication key (also configure the same key on the TACACS+ servers). Enable AAA. Create a login authentication method list. Apply the list to the terminal lines. Create an authorization and accounting method list as required.	在交换机上配置 TACACS+服务器的地址 设置一个认证密钥（也要在 TACACS+服务器上配置相同的密钥） 启用 AAA 创建登录认证方法列表 在终端线路上应用列表 按需创建授权和审计方法列表

用户需要通过 AAA 安全服务对 TACACS+进行管理，TACACS+可以提供以下功能：

- 认证——通过登录和密码对话、质询和响应，以及消息传递，来提供完全受控的身份认证。

  认证功能可以与用户进行对话（举例来说，在提供了用户名和密码之后，使用以下问题来质询用户：比如家庭地址、母亲的姓氏、服务类型和身份证号码）。TACACS+认证服务还可以向用户屏幕发送消息。比如它可以通过消息来通知用户：由于公司的密码老化策略，用户必须更改自己的密码；
- 授权——在用户会话期间对用户功能提供细粒度的控制，这些控制包括但不限于：设置自动命令、访问控制、会话持续时间，或协议支持。用户还可以使用 TACACS+授权功能，限制用户所能够执行的命令；
- 审计——收集用于记帐、审计和报告的信息，并将其发送到 TACACS+守护进程。网络管理员可以使用审计功能来跟踪用于安全审计的用户活动，或者提供用于用户计费的信息。审计记录中包括用户身份、开始和停止时间、执行的命令（比如 PPP）、数据包数量，以及字节数。

TACACS+协议在交换机和 TACACS+守护程序之间提供了认证功能，并且它能够保证机密性，

因为交换机和 TACACS+守护程序之间的所有协议交换消息都进行了加密。

TACACS+的工作原理

当用户使用 TACACS+来对简单的 ASCII 登录尝试进行认证时，会发生以下过程：

1. 当连接建立时，交换机会联系 TACACS+守护程序，从而向用户显示出输入用户名的提示符。用户输入用户名后，交换机会再联系 TACACS+守护程序，从而向用户显示出输入密码的提示符。交换机向用户显示输入密码的提示符后，用户输入密码，之后交换机会把密码发送到 TACACS+守护程序。

   TACACS+支持守护程序和用户之间的对话，直到守护程序收到足够多的信息来认证用户为止。守护程序能够提示用户输入用户名和密码组合，但可以包括其他内容，比如用户母亲的姓氏；
2. 交换机最终会从 TACACS+守护程序那里收到以下响应之一：
   - ACCEPT（接受）——用户通过了认证，并开始获得服务。如果交换机上配置了授权，则现在开始进行授权；
   - REJECT（拒绝）——用户没有通过认证。并根据 TACACS+守护程序，拒绝用户的访问，或者为用户提示重试登录；
   - ERROR（错误）——在使用守护程序进行身份认证的过程中发生错误，或守护程序与交换机之间的网络连接发生错误。如果接收到了 ERROR 响应，交换机通常尝试使用替代方法来对用户进行认证；
   - CONTINUE（继续）——为用户提示其他认证信息。

   如果用户在交换机上启用了授权，那么在用户通过认证之后，会经历额外的授权阶段。用户必须首先成功完成 TACACS+身份验证，然后才能进行 TACACS+授权。
3. 如果需要使用 TACACS+授权，交换机会再次联系 TACACS+守护进程，并接收到 ACCEPT 或 REJECT 授权响应。如果交换机收到的是 ACCEPT 响应，则响应中包含了属性数据，指出用户提供的 EXEC 或 NETWORK 会话，以及用户可以访问的服务：
   - Telnet、安全壳（SSH）、远程登录，或特权 EXEC 服务；
   - 连接参数，其中包括主机或客户端 IP 地址、访问列表和用户超时时间。

方法列表

方法列表定义了为用户进行认证、授权或审计的序列和方法。用户可以使用方法列表来指定要使用的一个或多个安全协议，这样做可以在初始方法失败时，确保有一个备份系统。软件

使用列表中的第一个方法来对用户进行认证、授权或审计；如果该方法没有获得响应，软件会选择列表中的下一个方法。这个过程会持续直到使用列出的方法实现成功通信，或者持续到方法列表耗尽。

TACACS+的配置选项

用户可以配置交换机来使用单台 AAA 服务器或 AAA 服务器组，为现有的服务器主机进行身份认证。用户可以有选择地把一部分服务器主机设置为一组服务器，并将其用于提供特定服务。服务器组与全局服务器主机列表一起使用，还包含所选服务器主机的 IP 地址列表。

TACACS+登录认证

方法列表定义了为用户进行认证的序列和方法。用户可以使用方法列表来指定要使用的一个或多个安全协议，这样做可以在初始方法失败时，确保有一个备份系统。软件使用列表中的第一个方法来对用户进行认证、授权或审计；如果该方法没有获得响应，软件会选择列表中的下一个方法。这个过程会持续直到使用列出的方法实现成功通信，或者持续到方法列表耗尽。如果在这个周期中的任何时刻认证失败了——意味着安全服务器或本地用户名数据库发出了响应，拒绝了用户的访问——这时认证过程就停止了，并且不会再尝试其他认证方法。

为特权 EXEC 访问和网络服务使用 TACACS+授权

AAA 授权能够限制用户可以使用的服务。当用户启用了 AAA 授权后，交换机会使用从用户配置文件中检索的信息来配置用户的会话，这个配置文件位于本地用户数据库中，或位于安全服务器。只有当用户配置文件中的信息允许时，用户才能够访问所请求的服务。

TACACS+审计

AAA 审计功能会跟踪用户正在访问的服务，以及用户消耗的网络资源总量。当用户启用了 AAA 审计后，交换机以审计记录的形式向 TACACS+安全服务器报告用户的活动。每个审计记录中都包含了审计属性值（AV）对，并且这些信息存储在安全服务器上。之后用户可以使用这些数据来对网络管理、客户端计费或审计进行分析。

默认的 TACACS+配置

TACACS+和 AAA 默认都是禁用的。

为了防止安全性失效，用户不能通过网络管理应用程序来配置 TACACS+。当启用了 TACACS+时，TACACS+可以验证通过 CLI 访问交换机的用户。

注释： 虽然用户是通过 CLI 执行 TACACS+配置的，但是 TACACS+服务器认证会认证 HTTP 连接，并已经为这个连接配置了特权级别 15。

如何配置 TACACS+

这部分描述了如何配置交换机来支持 TACACS+。

标识 TACACS+服务器主机并设置认证密钥

用户可以按照以下步骤标识 TACACS+服务器主机并设置认证密钥：

总步骤

1. enable

2. configure terminal

3. tacacs-server host *hostname*

4. aaa new-model

5. aaa group server tacacs+ *group-name*

6. server *ip-address*

7. end

8. show running-config

9. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：**	进入特权 EXEC 模式。在提示时输入密码

	Device> **enable**	
步骤 2	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式
步骤 3	**tacacs-server host** *hostname* **示例：** Device(config)# **tacacs-server host yourserver**	标识维护 TACACS+服务器的一台或多台 IP 主机。用户需要多次输入这条命令，来创建相应的主机列表。软件会按照用户配置的顺序来搜索这些主机。 在 *hostname* 部分指定主机的名称或 IP 地址
步骤 4	**aaa new-model** **示例：** Device(config)# **aaa new-model**	启用 AAA
步骤 5	**aaa group server tacacs+** *group-name* **示例：** Device(config)# **aaa group server tacacs+ your_server_group**	（可选）使用组名定义一个 AAA 服务器组。 这条命令会让设备进入服务器组子配置模式
步骤 6	**server** *ip-address* **示例：** Device(config)# **server 10.1.2.3**	（可选）把指定的 TACACS+服务器关联到定义的服务器组中。用户需要重复配置这条命令，以便把所有相关的 TACACS+服务器都放入 AAA 服务器组中。 用户必须先通过步骤 3 来定义这些需要被放入组中的服务器
步骤 7	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式

步骤 8	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 9	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 TACACS 登录认证

用户可以按照以下步骤来配置 TACACS+登录认证：

在开始前

要想配置 AAA 认证，用户需要定义一个命名的认证方法列表，并把它应用给各种端口。

注释： 要想使用 AAA 方法保障 HTTP 访问的安全性，用户必须还得配置全局配置命令 **ip http authentication aaa**。只配置 AAA 认证功能并不能通过使用 AAA 方法来确保 HTTP 访问的安全性。

更多有关命令 **ip http authentication** 的信息，用户可以参考 *Inspur INOS Security Command Reference, Release 12.4*。

总步骤

1. enable

2. configure terminal

3. aaa new-model

4. aaa authentication login {**default** | *list-name*} *method1* [*method2...*]

5. line [**console** | **tty** | **vty**] *line-number* [*ending-line-number*]

6. login authentication {**default** | *list-name*}

7. end

8. show running-config

9. copy running-config startup-config

具体步骤

	命令或操作	**目的**

步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**aaa new-model** **示例：** `Device(config)# aaa new-model`	启用 AAA
步骤 4	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] **示例：** `Device(config)# aaa authentication login default tacacs+ local`	创建一个登录认证方法列表。 • 要想创建一个默认列表，当用户*没有*在 **login authentication** 命令中指定命名列表时就使用这个默认列表，用户需要在默认情况下使用的方法后面添加 **default** 关键字。默认方法列表会自动被应用到所有端口 • 在 *list-name* 部分指定一个字符串，用来命名用户创建的列表 • 在 *method1…*部分指定认证算法使用的实际方法。其他认证方法只有当前一个方法返回了错误响应消息时才会使用，而不是返回失败消息时使用。 用户可以选择以下方法之一： • *enable*——使用 enable 密码来进行认证。在用户可以使用这个认证方法前，必须使用全局配置命令

		enable *password* 来定义一个 enable 密码 • *group tacacs+*——使用 TACACS+认证。在用户可以使用这个认证方法前，必须配置 TACACS+服务器。更多信息用户可以参考标识 TACACS+服务器主机并设置认证密钥 • *line*——使用线路密码来进行认证。在用户可以使用这个认证方法前，必须先定义一个线路密码。用户可以使用线路配置命令 **password** *password* • *local*——使用本地用户名数据库进行认证。用户必须输入数据库中的用户名信息。需要使用全局配置命令 **username** *password* 进行配置 • *local-case*——使用区分大小写的本地用户名数据库进行认证。用户必须使用全局配置命令 **username** *name* **password**，把用户名信息输入到数据库中 • *none*——不为登录使用任何认证
步骤 5	**line** [**console** | **tty** | **vty**] *line-number* [*ending-line-number*] **示例：** `Device(config)# line 2 4`	进入线路配置模式，并对想要应用认证列表的线路进行配置
步骤 6	**login authentication** {**default** | *list-name*} **示例：** `Device(config-line)# login`	把认证列表应用在一条或多条线路上。 • 如果用户指定了 **default**，就使用命令 **aaa authentication login** 创建默认列表 • 在 *list-name* 部分指定 **aaa**

	authentication default	**authentication login** 命令中创建的列表
步骤 7	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 8	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 9	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

为特权 EXEC 访问和网络服务配置 TACACS+授权

用户可以在全局配置命令 **aaa authorization** 中使用 **tacacs+**关键字，来对用户访问特权 EXEC 模式的行为设置限制参数。

注释： 对于通过 CLI 登录且已通过了认证的用户，即使配置了授权，也会绕过授权。

用户可以按照以下步骤为特权 EXEC 访问和网络服务配置 TACACS+授权：

总步骤

1. enable

2. configure terminal

3. aaa authorization network tacacs+

4. aaa authorization exec tacacs+

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的

步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**aaa authorization network tacacs+** **示例：** `Device(config)# aaa authorization network tacacs+`	配置交换机为所有与网络相关的服务请求使用 TACACS+授权
步骤 4	**aaa authorization exec tacacs+** **示例：** `Device(config)# aaa authorization exec tacacs+`	配置交换机为特权 EXEC 的访问使用 TACACS+授权。 **exec** 关键字可能会返回用户配置文件信息（比如 **autocommand** 信息）
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

开始使用 TACACS+审计

用户可以按照以下步骤开始使用 TACACS+审计：

总步骤

1. enable

2. configure terminal

3. aaa accounting network start-stop tacacs+

4. aaa accounting exec start-stop tacacs+

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**aaa accounting network start-stop tacacs+** **示例：** `Device(config)# aaa accounting network start-stop tacacs+`	为所有与网络相关的服务请求启用 TACACS+审计
步骤 4	**aaa accounting exec start-stop tacacs+** **示例：** `Device(config)# aaa`	启用 TACACS+审计，在开始特权 EXEC 处理时发送开始记录（start-record）审计通知，在结束时发送停止记录（stop-record）

	accounting exec start-stop tacacs+	
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

接下来做什么？

如果在 AAA 服务器不可达时，要与路由器建立会话，用户可以使用 **aaa accounting system guarantee-first** 命令。这条命令可以确保系统审计为第一条记录，这也是默认的条件。在有些情况下，这种设置可能会阻止用户在 Console 或终端连接上启动会话，直到系统重启才能解决问题，这可能需要 3 分钟以上的时间。

如果路由器重启时 AAA 服务器不可达，要想与路由器建立 Console 或 Telnet 会话，用户可以使用 **no aaa accounting system guarantee-first** 命令。

在 AAA 服务器不可达时与路由器建立会话

如果在 AAA 服务器不可达时，要与路由器建立会话，用户可以使用 **aaa accounting system guarantee-first** 命令。这条命令可以确保系统审计为第一条记录，这也是默认的条件。在有些情况下，这种设置可能会阻止用户在 Console 或终端连接上启动会话，直到系统重启才能解决问题，这可能需要 3 分钟以上的时间。

如果路由器重启时 AAA 服务器不可达，要想与路由器建立 Console 或 Telnet 会话，用户可以使用 **no aaa accounting system guarantee-first** 命令。

监控 TACACS+

表 124：显示 TACACS+信息的命令

命令	目的
show tacacs	显示 TACACS+服务器的状态统计信息

MACsec 加密

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

MAC 加密的相关信息

本章描述了如何在 Inspur 6850 和 6650 交换机上配置媒介访问控制安全性（MACsec）加密特性。

MACsec 是 IEEE 802.1AE 标准，用来在两个具有 MACsec 功能的设备之间，为数据包提供认证和加密。Inspur 交换机能够在下行链路端口上支持使用 MACsec 密钥协定（MKA）的 802.1AE

加密，用来在交换机和主机设备之间进行加密。该交换机还支持使用 Inspur TrustSec 网络设备准入控制（NDAC）、安全关联协议（SAP）和基于 MKA 的密钥交换协议，为交换机到交换机（网络之间的设备）安全提供 MACsec 加密。链路层安全中也可以包含交换机之间的数据包认证，以及交换机之间的 MACsec 加密（加密是可选的）。

注释： NPE 许可或 LAN Base 服务镜像不支持 MACsec 特性。

表 125：交换机端口上支持的 MACsec

接口	**连接**	**对 MACsec 的支持**
下行链路端口	交换机到主机	MACsec MKA 加密
上行链路端口	交换机到交换机	MACsec MKA 加密 Inspur TrustSec NDAC Macsec

Inspur TrustSec 和 Inspur SAP 特性仅适用于交换机到交换机之间的链路，在连接终端主机（如 PC 或 IP 电话）的交换机端口上不支持这些特性。交换机到主机之间的链路（下行链路）以及交换机到交换机之间的链路（上行链路）上都支持 MKA 特性。面向主机的链路通常使用灵活的认证序列，来处理支持或不支持 IEEE 802.1x 的异构设备，并且还可以（可选地）使用基于 MKA 的 MACsec 加密。Inspur NDAC 和 SAP 特性，与网络边缘访问拓扑（NEAT）互不兼容，NEAT 用于紧凑型交换机，把安全性扩展到配线柜之外。

媒介访问控制安全性和 Macsec 密钥管理

MACsec 定义在 802.1AE 中，它通过对加密密钥使用带外方法，在有线网络上提供 MAC 层加密。MACsec 密钥协议（MKA）协议提供了需要使用的会话密钥，并且负责管理这个加密密钥。在使用 802.1x 可扩展认证协议（EAP-TLS）或预共享密钥（PSK）框架成功进行了认证后，便实施 MKA 和 MACsec。

根据与 MKA 对等体相关联的策略，使用 MACsec 特性的交换机能够接收 MACsec 帧或非 MACsec 帧。MACsec 帧是使用完整性校验值（ICV）进行加密和保护的。当交换机从 MKA 对等体接收到数据帧时，它会使用由 MKA 提供的会话密钥来对数据帧进行解密，并计算出正确的 ICV。交换机会把自己计算出的 ICV 与数据帧中携带的 ICV 进行比较。如果两者不相同，则交换机会丢弃这个数据帧。交换机还会使用当前的会话密钥，对通过安全端口（用来向 MKA 对等体提供安全 MAC 服务的接入点）发送的数据帧进行加密并添加 ICV。

MKA 协议会管理底层 MACsec 协议使用的加密密钥。MKA 的基本要求定义在 802.1x-REV 中。MKA 协议扩展了 802.1x，允许具有相互认证能力且共享 MACsec 密钥的对等体发现彼此，以此保护对等体之间交互的数据。

EAP 框架把 MKA 实现为新定义的 EAP-over-LAN（EAPOL）数据包。EAP 认证会产生一个主用会话密钥（MSK），数据交换中的两个对等体会共享这个 MSK。输入 EAP 会话 ID 会生成一个安全连接关联密钥名称（CKN）。交换机会对上行链路和下行链路进行认证；并充当下行链路的密钥服务器。它会生成一个随机安全关联密钥（SAK），并将其发送到客户端对等体。客户端永远不会是密钥服务器，并且只能与单个 MKA 实体（密钥服务器）进行交互。在派生和生成密钥后，交换机会以默认 2 秒钟的时间间隔周期性向对等体发送消息。

EAPOL 协议数据单元（PDU）数据包中的负载被称为 MACsec 密钥协商 PDU（MKPDU）。当 MKA 生命周期（6 秒钟）超时后，仍没有从对等体接收到 MKPDU，MKA 会话和对等体就会被删除。举例来说，如果 MKA 对等体断开了连接，交换机上的参与者会继续操作 MKA，直到从 MKA 对等体接收到最后一个 MKPDU 之后又过去了 6 秒钟。

MKA 策略

要想在接口上启用 MKA，用户应该在接口上应用一个已经定义好的 MKA 策略。用户可以配置以下选项：

- 策略名称，不超过 16 个 ASCII 字符；
- 为每个物理接口配置保密（加密）偏移 0、30 或 50 字节。

虚拟端口

用户可以使用虚拟端口为单个物理端口提供多个安全连接关联。每个连接关联（对）代表着一个虚拟端口。在上行链路中，每个物理端口上只能有一个虚拟端口。在下行链路中，每个物理端口上最多可以有两个虚拟端口，其中一个虚拟端口可以是数据 VLAN 的一部分；另一个必须为语音 VLAN 来标记数据包。用户不能在同一端口上的同一 VLAN 中，同时承载安全会话和不安全的会话。正因有此限制，所以不支持 802.1x 多重身份验证模式。

对于这个限制有一个例外：在多主机模式下，第一个 MACsec 请求方成功通过认证，并且它通过一个集线器与交换机相连。这时集线器上连接的其他非 MACsec 主机也可以发送流量，而无需身份验证，因为这时使用的是多主机模式。我们不建议使用多主机模式，因为在第一个客户端验证成功后，其他客户端都不需要进行身份验证了。

虚拟端口是一个用来代表连接关联的任意标识符，并且除非用于 MKA 协议，否则是没有意义的。虚拟端口对应着单独的逻辑端口 ID。虚拟端口的有效端口号范围是 0x0002 至 0xFFFF。每个虚拟端口都会接收唯一的安全信道标识符（SCI），SCI 是由物理接口的 MAC 地址和 16 位端口 ID 构成的。

MACsec 和堆栈

Inspur 6850 交换机堆栈中运行 MACsec 的主用设备会维护配置文件，配置文件中展示了成员交换机上的哪些端口支持 MACsec。堆栈主用设备会执行以下功能：

- 处理安全通道和安全关联的创建和删除；
- 向堆栈成员发送安全关联服务请求；
- 处理来自本地或远端端口的数据包编号和响应窗口信息，并向密钥关系协议发送通知；
- 向新添加到堆栈的交换机发送 MACsec 初始化请求和全局配置的选项；
- 向成员交换机发送基于每个端口的配置。

成员交换机会执行以下功能：

- 处理来自堆栈主用设备的 MACsec 初始化请求；
- 处理由堆栈主用设备的 MACsec 服务请求；
- 向堆栈主用设备发送有关本地端口的信息。

MACsec、MKA 和 802.1x 主机模式

用户可以结合使用 MACsec 和 MKA 协议，以及 802.1x 单主机模式、多主机模式，或多域认证（MDA）模式。不支持多认证模式。

单主机模式

下图展示了如何通过使用 MKA，由 MACsec 对单 EAP 认证会话提供保护的。

图 99：单主机模式的 MACsec 和受保护数据会话

Host	主机
Usecured	未受保护
Switch with MACsec configured	配置了 MACsec 的交换机
AAA Access-control system	AAA 访问控制系统

多主机模式

在标准的（非 802.1x REV）802.1x 多主机模式下，端口会基于单个认证进行打开或关闭。如果一个用户（主要受保护的客户端服务于客户端主机）通过了认证，交换机会向连接到同一端口的任意主机提供相同级别的网络访问。如果备用主机是 MACsec 请求方，则它不能被认

证，并且流量也不会被转发。如果备用主机是非 MACsec 主机的话，则它可以在不进行认证的情况下向网络发送流量，因为它处于多主机模式中。下图显示了标准多主机未受保护模式下的 MACsec。

图 100：多主机模式中的 MACsec——未受保护

Primary host	主用主机
Secondary host（共 2 处）	备用主机
Switch with MACsec configured	配置了 MACsec 的交换机
AAA Access-control system	AAA 访问控制系统

注释： 不建议使用多主机模式，因为在第一个客户端成功认证后，其他客户端都不需要进行认证了，这种方式并不安全。

在标准的（非 802.1x REV）802.1x 多域模式下，端口是基于单个认证进行打开或关闭的。如果主用户（数据域上的 PC）通过了认证，交换机会为连接到同一端口上的任意域提供相同级别的网络访问。如果备用用户是 MACsec 请求方，则它不能被认证，并且流量也不会被转发。备用用户是指语音域上的 IP 电话（即非 MACsec 主机），它可以向网络中发送流量，无需进行身份验证，因为它处于多域模式。

MKA 状态统计信息

有些 MKA 计数器是在全局收集的，而其他 MKA 计数器是在全局和每个会话中进行更新的。用户还可以查看有关 MKA 会话状态的信息。

以下为 **show mka statistics** 命令的输出示例：

```
Switch# show mka sessions
Total MKA Sessions. ..............  1
Secured Sessions... 1
Pending Sessions... 0
=========================================================================
==================================
Interface Local-TxSCI Policy-Name Inherited Key-Server
Port-ID Peer-RxSCI MACsec-Peers Status CKN
=========================================================================
==================================
Gi1/0/1 204c.9e85.ede4/002b p2 NO YES
```

```
43 c800.8459.e764/002a 1 Secured
0100000000000000000000000000000000000000000000000000000000000000
Switch#show mka sessions interface G1/0/1
Summary of All Currently Active MKA Sessions on Interface
GigabitEthernet1/0/1...
=====================================================================
===================================
Interface Local-TxSCI Policy-Name Inherited Key-Server
Port-ID Peer-RxSCI MACsec-Peers Status CKN
=====================================================================
===================================
Gi1/0/1 204c.9e85.ede4/002b p2 NO YES
43 c800.8459.e764/002a 1 Secured
0100000000000000000000000000000000000000000000000000000000000000
Switch#show mka sessions interface G1/0/1 de
MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec
Local Tx-SCI..... 204c.9e85.ede4/002b
Interface MAC Address.... 204c.9e85.ede4
MKA Port Identifier. ..... 43
Interface Name..... GigabitEthernet1/0/1
Audit Session ID.....
CAK                    Name                    (CKN).....
0100000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)... D46CBEC05D5D67594543CEAE
Message Number (MN)..... 89567
EAP Role..... NA
Key Server..... YES
MKA Cipher Suite..... AES-128-CMAC
Latest SAK Status..... Rx & Tx
Latest SAK AN. ..... 0
Latest SAK KI (KN)..... D46CBEC05D5D67594543CEAE00000001 (1)
```

```
Old SAK Status.......... FIRST-SAK
Old SAK AN. ................................ 0
Old SAK KI (KN).......... FIRST-SAK (0)
SAK Transmit Wait Time... 0s (Not waiting for any peers to respond)
SAK Retire Time.......... 0s (No Old SAK to retire)
MKA Policy Name.......... p2
Key Server Priority. ............. 2
Delay Protection......... NO
Replay Protection........ YES
Replay Window Size. ............... 0
Confidentiality Offset... 0
Algorithm Agility........ 80C201
SAK Cipher Suite......... 0080C20001000001 (GCM-AES-128)
MACsec Capability........ 3 (MACsec Integrity, Confidentiality, &
Offset)
MACsec Desired........... YES
# of MACsec Capable Live Peers. ........................... 1
# of MACsec Capable Live Peers Responded.. 1
Live Peers List:
MI MN Rx-SCI (Peer) KS Priority
-----------------------------------------------------------------
-----
38046BA37D7DA77E06D006A9 89555 c800.8459.e764/002a 10
Potential Peers List:
MI MN Rx-SCI (Peer) KS Priority
-----------------------------------------------------------------
-----
Dormant Peers List:
MI MN Rx-SCI (Peer) KS Priority
-----------------------------------------------------------------
-----
Switch#show mka sessions de
Switch#show mka sessions detail
```

MKA Detailed Status for MKA Session

==========

Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI..... 204c.9e85.ede4/002b

Interface MAC Address.... 204c.9e85.ede4

MKA Port Identifier. 43

Interface Name..... GigabitEthernet1/0/1

Audit Session ID.....

CAK Name (CKN).....

01000000000000000000000000000000000000000000000000000000000000000000

Member Identifier (MI)... D46CBEC05D5D67594543CEAE

Message Number (MN)..... 89572

EAP Role..... NA

Key Server..... YES

MKA Cipher Suite..... AES-128-CMAC

Latest SAK Status..... Rx & Tx

Latest SAK AN. 0

Latest SAK KI (KN)..... D46CBEC05D5D67594543CEAE00000001 (1)

Old SAK Status..... FIRST-SAK

Old SAK AN. 0

Old SAK KI (KN)..... FIRST-SAK (0)

SAK Transmit Wait Time... 0s (Not waiting for any peers to respond)

SAK Retire Time..... 0s (No Old SAK to retire)

MKA Policy Name..... p2

Key Server Priority. 2

Delay Protection..... NO

Replay Protection..... YES

Replay Window Size. 0

Confidentiality Offset... 0

Algorithm Agility..... 80C201

SAK Cipher Suite..... 0080C20001000001 (GCM-AES-128)

MACsec Capability..... 3 (MACsec Integrity, Confidentiality, &

Offset)

```
MACsec Desired..... YES
# of MACsec Capable Live Peers. ..... 1
# of MACsec Capable Live Peers Responded.. 1
Live Peers List:
MI MN Rx-SCI (Peer) KS Priority
-----------------------------------------------------------------
-----
38046BA37D7DA77E06D006A9 89560 c800.8459.e764/002a 10
Potential Peers List:
MI MN Rx-SCI (Peer) KS Priority
-----------------------------------------------------------------
-----
Dormant Peers List:
MI MN Rx-SCI (Peer) KS Priority
-----------------------------------------------------------------
-----
Switch#sh mka pol
MKA Policy Summary...
Policy KS Delay Replay Window Conf Cipher Interfaces
Name Priority Protect Protect Size Offset Suite(s) Applied
=================================================================
====================================
*DEFAULT POLICY* 0 FALSE TRUE 0 0 GCM-AES-128
p1 1 FALSE TRUE 0 0 GCM-AES-128
p2 2 FALSE TRUE 0 0 GCM-AES-128 Gi1/0/1
Switch#sh mka poli
Switch#sh mka policy p2
Switch#sh mka policy p2 ?
detail Detailed configuration/information for MKA Policy
sessions Summary of all active MKA Sessions with policy applied
| Output modifiers
<cr>
Switch#sh mka policy p2 de
```

```
MKA Policy Configuration ("p2")
========================
MKA Policy Name........ p2
Key Server Priority. ......... 2
Confidentiality Offset. 0
Cipher Suite(s)........ GCM-AES-128
Applied Interfaces...
GigabitEthernet1/0/1
Switch#sh mka policy p2
MKA Policy Summary...
Policy KS Delay Replay Window Conf Cipher Interfaces
Name Priority Protect Protect Size Offset Suite(s) Applied
================================================================
====================================
p2 2 FALSE TRUE 0 0 GCM-AES-128 Gi1/0/1
Switch#sh mka se?
sessions
Switch#sh mka ?
default-policy MKA Default Policy details
keychains MKA Pre-Shared-Key Key-Chains
policy MKA Policy configuration information
presharedkeys MKA Preshared Keys
sessions MKA Sessions summary
statistics Global MKA statistics
summary MKA Sessions summary & global statistics
Switch#sh mka statis
Switch#sh mka statistics ?
interface Statistics for a MKA Session on an interface
local-sci Statistics for a MKA Session identified by its Local Tx-SCI
| Output modifiers
<cr>
Switch#sh mka statistics inter
Switch#show mka statistics interface G1/0/1
```

```
MKA Statistics for Session
==========================
Reauthentication Attempts.. 0
CA Statistics
Pairwise CAKs Derived... 0
Pairwise CAK Rekeys. ..... 0
Group CAKs Generated. ..... 0
Group CAKs Received. ..... 0
SA Statistics
SAKs Generated. ..... 1
SAKs Rekeyed. ..... 0
SAKs Received. ..... 0
SAK Responses Received.. 1
MKPDU Statistics
Switch#show mka ?
default-policy MKA Default Policy details
keychains MKA Pre-Shared-Key Key-Chains
policy MKA Policy configuration information
presharedkeys MKA Preshared Keys
sessions MKA Sessions summary
statistics Global MKA statistics
summary MKA Sessions summary & global statistics
Switch#show mka summ
Switch#show mka summary
Total MKA Sessions. ..... 1
Secured Sessions... 1
Pending Sessions... 0
=====================================================================
==================================
Interface Local-TxSCI Policy-Name Inherited Key-Server
Port-ID Peer-RxSCI MACsec-Peers Status CKN
=====================================================================
==================================
```

Gi1/0/1 204c.9e85.ede4/002b p2 NO YES

43 c800.8459.e764/002a 1 Secured

0100000000000000000000000000000000000000000000000000000000000000

MKA Global Statistics

==========

MKA Session Totals Secured. ....

1

Reauthentication Attempts.. 0

Deleted (Secured). 0

Keepalive Timeouts. 0

CA Statistics

Pairwise CAKs Derived. 0

Pairwise CAK Rekeys. 0

Group CAKs Generated. 0

Group CAKs Received. 0

SA Statistics

SAKs Generated. 1

SAKs Rekeyed. 0

SAKs Received. 0

SAK Responses Received. 1

MKPDU Statistics

MKPDUs Validated & Rx..... 89589

"Distributed SAK". 0

"Distributed CAK". 0

MKPDUs Transmitted..... 89600

"Distributed SAK". 1

"Distributed CAK". 0

MKA Error Counter Totals

==========

Session Failures

Bring-up Failures..... 0

Reauthentication Failures. 0

Duplicate Auth-Mgr Handle. 0

```
SAK Failures
SAK Generation. ..................................... 0
Hash Key Generation. ............................... 0
SAK Encryption/Wrap. ............................... 0
SAK Decryption/Unwrap. ............................. 0
SAK Cipher Mismatch. ............................... 0
CA Failures
Group CAK Generation............................... 0
Group CAK Encryption/Wrap. ................... 0
Group CAK Decryption/Unwrap. ............. 0
Pairwise CAK Derivation. ...................... 0
CKN Derivation. ..................................... 0
ICK Derivation. ..................................... 0
KEK Derivation. ..................................... 0
Invalid Peer MACsec Capability... 0
MACsec Failures
Rx SC Creation. ..................................... 0
Tx SC Creation. ..................................... 0
Rx SA Installation. ................................ 0
Tx SA Installation. ................................ 0
MKPDU Failures
MKPDU Tx. ..........................................................
0
MKPDU Rx Validation. ............................... 0
MKPDU Rx Bad Peer MN.............................. 0
MKPDU Rx Non-recent Peerlist MN.. 0
Switch#
```

配置 MKA 和 MACsec

默认的 MACsec MKA 配置

默认 MACsec 是禁用的，也没有配置任何 MKA 策略。

配置 MKA 策略

总步骤

1. configure terminal

2. mka policy *policy name*

3. key-server *priority*

4. macsec-cipher-suite *gcm-aes-128*

5. confidentiality-offset *Offset value*

6. end

7. show mka policy

具体步骤

	命令或操作	目的
步骤 1	**configure terminal**	进入全局配置模式
步骤 2	**mka policy** *policy name*	标识一个 MKA 策略，并进入 MKA 策略配置模式。策略名称的长度最长为 16 字节
步骤 3	**key-server** *priority*	配置 MKA 密钥服务器选项并设置优先级（0 至 255）。 **注释：** 在用户把密钥服务器的优先级设置为 255 后，对等体就不会成为密钥服务器
步骤 4	**macsec-cipher-suite** *gcm-aes-128*	配置用于生成 128 比特加密 SAK 的加密套件
步骤 5	**confidentiality-offset** *Offset value*	为每个物理接口设置保密（加密）偏移。 **注释：** 偏移值可以是 0、30 或 50。

		如果用户在客户端上使用 Anyconnect 软件，建议设置偏移 0
步骤 6	**end**	返回特权 EXEC 模式
步骤 7	**show mka policy**	检查用户输入的信息

以下示例中展示了 MKA 策略的配置：

```
Switch(config)# mka policy mka_policy
Switch(config-mka-policy)# key-server priority 200
Switch(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Switch(config-mka-policy)# confidentiality-offset 30
Switch(config-mka-policy)# end
```

在接口上配置 MACsec

用户可以按照以下步骤在接口上配置 MACsec，一个 MACsec 会话用于语音，另一个用于数据：

总步骤

1. enable

2. configureterminal

3. interface *interface-id*

4. switchport access vlan*vlan-id*

5. switchport mode access

6. macsec

7. authentication event linksec fail action authorize vlan *vlan-id*

8. authentication host-mode multi-domain

9. authentication linksec policy must-secure

10. authentication port-control auto

11. authentication periodic

12. authentication timer reauthenticate

13. authentication violation protect

14. mka policy *policy name*

15. dot1x pae authenticator

16. spanning-tree portfast

17. end

18. show authentication session interface *interface-id*

19. show authentication session interface *interface-id* details

20. show macsec interface *interface-id*

21. show mka sessions

22. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Switch> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Swich# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id*	标识 MACsec 接口，并进入接口配置模式。这个接口必须是物理接口
步骤 4	**switchport access vlan***vlan-id*	为端口配置一个 Access VLAN
步骤 5	**switchport mode access**	把接口配置为 Access 端口
步骤 6	**macsec**	在接口上启用 802.1ae MACsec。macsec 命令至在交换机到主机之间的链路（下行端口）上启用 MKA MACsec
步骤 7	**authentication event linksec fail action authorize vlan** *vlan-id*	（可选）通过配置，让交换机在认证尝试失败后，为端口授权一个受限 VLAN，用来处理未授权用户的认证链路安全失败事件
步骤 8	**authentication host-mode multi-domain**	在端口上配置认真管理器模式，允许一台主机和一台语音设备在 802.1x 授权的端口上进行认证。如果没有配置这条命令，默认的主机模式是单主机模式
步骤 9	**authentication linksec policy must-secure**	设置 LinkSec 安全策略，在对等体可用时使用 MACsec 特性来保护会话的安

		全。如果没有设置这条命令，默认行为是*应该提供安全保护*
步骤 10	**authentication port-control auto**	在端口上启用 802.1x 认证。端口会根据交换机和客户端之间交换的认证结果，来改变授权或未授权状态
步骤 11	**authentication periodic**	为这个端口启用或禁用重认证功能
步骤 12	**authentication timer reauthenticate**	输入 1 至 65535（以秒为单位）之间的值。从服务器获得重认证超时时间。默认的重认证时间是 3600 秒
步骤 13	**authentication violation protect**	配置端口在发生以下事件时丢弃入站 MAC 地址：新设备连接到端口，或端口上已连接了最大数量的设备后又连接了新设备。如果没有配置这条命令，默认行为是关闭（shutdown）端口
步骤 14	**mka policy** *policy name*	在接口上应用现有的 MKA 协议策略，并在接口上启用 MKA。如果用户没有通过全局配置命令 **mka policy** 配置 MKA 策略的话，
步骤 15	**dot1x pae authenticator**	把端口配置为 802.1x 端口访问实体（PAE）认证器
步骤 16	**spanning-tree portfast**	在接口上为其关联的所有 VLAN 都启用生成树 Post Fast 特性。在启用了 Port Fast 特性后，接口会直接从阻塞状态进入转发状态，无需经历生成树中间状态
步骤 17	**end** **示例：** `Switch(config)#end`	返回特权 EXEC 模式
步骤 18	**show authentication session interface** *interface-id*	检查授权的会话安全状态
步骤 19	**show authentication session interface** *interface-id* **details**	检查授权会话的安全状态详情

步骤 20	**show macsec interface** *interface-id*	检查接口上的 MACsec 状态
步骤 21	**show mka sessions**	检查已建立的 MKA 会话
步骤 22	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

使用 PSK 配置 MACsec MKA

总步骤

1. configure terminal

2. key chain *key-chain-name* **macsec**

3. key *hex-string*

4. cryptographic-algorithm {*gcm-aes-128* | *gcm-aes-256*}

5. key-string { [*0*|*6*|*7*] *pwd-string* | *pwd-string*}

6. lifetime local [*start timestamp* {*hh::mm::ss* | *day* | *month* | *year*}] [**duration** *seconds* | *end timestamp* {*hh::mm::ss* | *day* | *month* | *year*}]

7. end

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal**	进入全局配置模式
步骤 2	**key chain** *key-chain-name* **macsec**	配置一个密钥链，并进入密钥链配置模式
步骤 3	**key** *hex-string*	为密钥链中的每个密钥配置唯一的识别符，并进入密钥链中的密钥配置模式。 **注释：** 对于 128 比特加密，使用 32 位十六进制数字密钥链。对于 256 比特加密，使用 64 位十六进制数字密钥链
步骤 4	**cryptographic-algorithm** {*gcm-aes-128*	*gcm-aes-256*}	使用 128 比特或 256 比特加密方式来设置加密认证算法

步骤 5	**key-string** { [*0*|*6*|*7*] *pwd-string* | *pwd-string*}	为密钥链设置密码。只能输入十六进制字符
步骤 6	**lifetime local** [*start timestamp* {*hh::mm::ss* | *day* | *month* | *year*}] [**duration** *seconds* | *end timestamp* {*hh::mm::ss* | *day* | *month* | *year*}]	设置预共享密钥的生存时间
步骤 7	**end**	返回特权 EXEC 模式

以下展示了一个示例：

```
Switch(config)# Key chain keychain1 macsec
Switch(config-key-chain)# key 1000
Switch(config-keychain-key)# cryptographic-algorithm gcm-aes-128
Switch(config-keychain-key)# key-string
12345678901234567890123456789012
Switch(config-keychain-key)# lifetime local 12:12:00 July 28 2016
12:19:00 July 28 2016
Switch(config-keychain-key)# end
```

在使用 PSK 的接口上配置 MACsec MKA

总步骤

1. configure terminal

2. interface *interface-id*

3. mka policy *policy-name*

4. mka pre-shared-key key-chain *key-chain name*

5. macsec replay-protection window-size *frame number*

6. end

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal**	进入全局配置模式
步骤 2	**interface** *interface-id*	进入接口配置模式
步骤 3	**mka policy** *policy-name*	配置一个 MKA 策略
步骤 4	**mka pre-shared-key key-chain** *key-chain*	配置一个 MKA 预共享密钥的密钥链名

	name	称。 **注释：** 用户可以在物理接口或子接口上配置 MKA 预共享密钥，但不能同时在接口和子接口上进行配置
步骤 5	**macsec replay-protection window-size** *frame number*	为重放保护设置 MACsec 窗口大小
步骤 6	**end**	返回特权 EXEC 模式

以下展示了一个示例：

```
Switch(config)# interface GigabitEthernet 0/0/0
Switch(config-if)# mka policy mka_policy
Switch(config-if)# mka pre-shared-key key-chain key-chain-name
Switch(config-if)# macsec replay-protection window-size 10
Switch(config-if)# end
```

理解使用 EAP-TLS 的 MACsec MKA

从 Inspur INOS 15.2(5)E 版本开始，Inspur 6850 和 6650 系列交换机能够在交换机到交换机之间的链路上支持 MACsec MKA。

通过使用 IEEE 802.1X 基于端口的认证和可扩展认证协议（EAP-TLS），用户可以在设备上行链路端口之间配置 MACsec MKA。EAP-TLS 能够实现相互认证，并且获得 MSK（主用会话密钥），MSK 会被 MKA 操作用来派生连接关联密钥（CAK）。EAP-TLS 会携带设备证书，以备 AAA 服务器认证。

使用 EAP-TLS 配置 MACsec MKA 的先决条件

- 用户要确保自己为网络配置了证书授权中心（CA）服务器；
- 生成一个 CA 证书；
- 用户要确保配置了 Inspur 身份服务引擎（ISE）2.0 版本；
- 用户要确保使用网络时间协议（NTP）对参与的设备、CA 服务器和 Inspur 身份服务引擎（ISE）进行同步。如果所有设备上的时间不同步，则无法验证证书；
- 用户要确保在设备上配置了 802.1x 认证和 AAA。

使用 EAP-TLS 配置 MACsec MKA 的限制条件

- Port-Channel 端口上不支持 MKA；
- 高可用性和本地认证功能与 MKA 不兼容。

配置使用 EAP-TLS 的 MACsec MKA

要想在点到点链路上配置 MACsec 和 MKA，用户需要执行以下工作：

- 配置证书注册
  - 生成密钥对
  - 配置 SECP 注册
  - 手动配置证书
- 配置一个认证策略
- 配置 EAP-TLS 配置文件和 IEEE 802.1x 证书
- 在接口上配置使用 EAP-TLS 的 MACsec MKA

生成密钥对

总步骤

1. configure terminal

2. crypto key generate rsa label *label name* **general-keys modulus** *size*

3. end

4. show authentication session interface *interface-id*

5. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal**	进入全局配置模式
步骤 2	**crypto key generate rsa label** *label name* **general-keys modulus** *size*	为信令交互和加密生成 RSA 密钥对。用户还可以使用 label 关键字为每个密钥对分配一个标签。使用密钥对的信任点会引用这个标签。如果不分配标签的

		话，密钥对会自动使用标签 <Default-RSA-Key> 如果不使用其他关键字，这条命令会生成一个通用的 RSA 密钥对。如果用户没有指定系数的话，默认的密钥系数为 1024。用户可以使用 modulus 关键字来指定其他系数大小。
步骤 3	**end**	返回特权 EXEC 模式
步骤 4	**show authentication session interface** *interface-id*	检查授权会话的安全状态
步骤 5	**copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

配置使用 SECP 进行注册

简单证书注册协议（SCEP）是 Inspur 开发的注册协议，使用 HTTP 来与证书授权中心（CA）或注册授权中心（RA）进行通信。SCEP 并不常用来发送和接收请求和证书。

总步骤

configure terminal

2. crypto pki trustpoint *server name*

3. enrollment url *url name pem*

4. rsakeypair *label*

5. serial-number none

6. ip-address none

7. revocation-check crl

8. auto-enroll *percent* **regenerate**

9. crypto pki authenticate *name*

10. exit

11. show crypto pki certificate *trustpoint name*

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal**	进入全局配置模式

步骤 2	**crypto pki trustpoint** *server name*	声明信任点并为其指定名称，并且进入 CA 信任点配置模式
步骤 3	**enrollment url** *url name pem*	指定 CA 的 URL，用户的设备应该向这个 CA 发送证书请求。 用户可以在URL中使用方括号配置IPv6地址。比如 http://[2001:DB8:1:1::1]:80。 用户需要使用 pem 关键字为证书请求添加隐私增强邮件（PEM）边界
步骤 4	**rsakeypair** *label*	指定与证书相关联的密钥对。 **注释： rsakeypair** 名称必须与信任点名称相匹配
步骤 5	**serial-number none**	**none** 关键字指定了不会包含在证书请求中的序列号
步骤 6	**ip-address none**	**none** 关键字指定了证书请求中不应该包含 IP 地址
步骤 7	**revocation-check crl**	把 CRL 作为方法，确保不会撤销对等体的证书
步骤 8	**auto-enroll** *percent* **regenerate**	启用自动注册特性，允许客户端自动向 CA 请求证书。 如果没有启用自动注册的话，客户端必须在证书过期时，手动在 PKI 中重新注册。 默认情况下，只有设备的域名系统（DNS）包含在证书中。 使用 percent 参数指定在当前证书的生命周期到达多少百分比后，要请求新的证书。 使用 regenerate 关键字为证书生成新的密钥，即使已经存在命名密钥。 如果滚动密钥对是可以导出的，则新密钥对也是可以导出的。用户会在信任点的配置中，看到以下注释，指示密钥对

		是否可以导出：“！RSA key pair associated with trustpoint is exportable.” 出于安全原因，建议生成新的密钥对
步骤 9	**crypto pki authenticate** *name*	检索 CA 证书并对其进行认证
步骤 10	**exit**	退出全局配置模式
步骤 11	**show crypto pki certificate** *trustpoint name*	显示与信任点相关的证书信息

手动配置注册

如果用户的 CA 不支持 SCEP，或者如果路由器和 CA 之间无法实现网络连接，用户可以按照以下步骤进行手动证书注册：

总步骤

1. configure terminal

2. crypto pki trustpoint *server name*

3. enrollment url *url name pem*

4. rsakeypair *label*

5. serial-number none

6. ip-address none

7. revocation-check crl

8. exit

9. crypto pki authenticate *name*

10. crypto pki enroll *name*

11. crypto pki import *name* **certificate**

12. exit

13. show crypto pki certificate *trustpoint name*

14. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal**	进入全局配置模式
步骤 2	**crypto pki trustpoint** *server name*	声明信任点并为其指定名称，并且进入

		CA 信任点配置模式
步骤 3	**enrollment url** *url name pem*	指定 CA 的 URL，用户的设备应该向这个 CA 发送证书请求。 用户可以在URL中使用方括号配置IPv6地址。比如 http://[2001:DB8:1:1::1]:80。 用户需要使用 pem 关键字为证书请求添加隐私增强邮件（PEM）边界
步骤 4	**rsakeypair** *label*	指定与证书相关联的密钥对。
步骤 5	**serial-number none**	**none** 关键字指定了不会包含在证书请求中的序列号
步骤 6	**ip-address none**	**none** 关键字指定了证书请求中不应该包含 IP 地址
步骤 7	**revocation-check crl**	把 CRL 作为方法，确保不会撤销对等体的证书
步骤 8	**exit**	退出全局配置模式
步骤 9	**crypto pki authenticate** *name*	检索 CA 证书并对其进行认证
步骤 10	**crypto pki enroll** *name*	生成证书请求，并显示证书服务器中复制和粘贴的请求。 在看到提示时输入注册信息。举例来说，指定是否要在证书请求中包含设备 FQDN 和 IP 地址。 用户还可以为 Console 终端提供显示和证书请求选项。 显示带有或不带有请求的 PEM 头部的基于 64 编码证书
步骤 11	**crypto pki import** *name* **certificate**	在 Console 终端上通过 HTTP 导入一个证书，它会检索授予的证书。 设备会尝试通过TFTP检索授予的证书，它会使用与发送请求所使用的文件名相同的文件名，但会把扩展名从“.req”更改为“.crt”。对于密钥证书的使用，它会使用扩展名“-sign.crt”和

		“-encr.crt”。 设备会对接收到的文件进行解析，对证书进行验证，并将证书插入到交换机的内部证书数据库中。 **注释：** 有些 CA 会忽略证书请求中使用的密钥信息，并发出通用目的的证书。如果 CA 忽略证书请求中使用的密钥信息，则只导入通用证书。路由器不会使用生成的两个密钥对中的任意一个
步骤 12	**exit**	退出全局配置模式
步骤 13	**show crypto pki certificate** *trustpoint name*	显示与信任点相关的证书信息
步骤 14	**copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

在接口应用 802.1x MACsec MKA 配置

用户可以按照以下步骤，在接口上应用使用了 EAP-TLS 的 MACsec MKA：

总步骤

1. configure terminal

2. interface *interface-id*

3. macsec network-link

4. authentication periodic

5. authentication timer reauthenticate interval

6. access-session host-mode multi-domain

7. access-session closed

8. access-session port-control auto

9. dot1x pae both

10. dot1x credentials profile

11. dot1x supplicant eap profile *name*

12. service-policy type control subscriber *control-policy name*

13. exit

14. show macsec interface

15. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal**	进入全局配置模式
步骤 2	**interface** *interface-id*	标识 MACsec 接口，并进入接口配置模式。这个接口必须是物理接口
步骤 3	**macsec network-link**	在接口上启用 MACsec
步骤 4	**authentication periodic**	为这个端口启用或禁用重认证功能
步骤 5	**authentication timer reauthenticate interval**	设置重认证时间间隔
步骤 6	**access-session host-mode multi-domain**	允许主机获得接口的访问权限
步骤 7	**access-session closed**	阻止接口上的预认证访问
步骤 8	**access-session port-control auto**	设置接口的授权状态
步骤 9	**dot1x pae both**	把端口配置为 802.1X 端口访问实体（PAE）请求方和认证方
步骤 10	**dot1x credentials profile**	为接口分配 802.1x 证书配置文件
步骤 11	**dot1x supplicant eap profile** *name*	为接口分配 EAP-TLS 配置文件
步骤 12	**service-policy type control subscriber** *control-policy name*	在接口上应用订阅者控制策略
步骤 13	**exit**	返回特权 EXEC 模式
步骤 14	**show macsec interface**	显示接口上的 MACsec 详情
步骤 15	**copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

Inspur TrustSec MACsec 的相关信息

下面这个表格中列出了 TrustSec 特性，这些特性是在 Inspur 交换机上最终实现的 TrustSec 功能。连续通用的 TrustSec 版本扩展了支持的交换机数量，以及每台交换机能够支持的 TrustSec 功能数量。

Inspur TrustSec 特性	**描述**
802.1 AE 标记（MACsec）	基于 IEEE 802.1AE 的有线速率逐跳二层加密协议。 在具有 MACsec 功能的设备之间，发送方设备在发送数据包时对其进行加密，接收方设备在接收到数据包时对其进行解密，在设备内数据包是明文的。 这个特性只能用于具有 TrustSec 硬件功能的设备之间。 **注释：** Inspur 2960x 交换机上不支持该特性
端点准入控制（EAC）	EAC 是当端点用户或设备在连接到 TrustSec 域时，对其进行身份验证过程。通常 EAC 是发生在接入层交换机上的行为。在 EAC 过程中成功的认证和授权后，用户或设备会获得安全组标记（Security Group Tag）。当前的 EAC 可以是 802.1X、MAC 认证旁路（MAB）和 Web 认证代理（WebAuth）
网络设备准入控制（NDAC）	NDAC 是一个认证过程，其中 TrustSec 域中的每台网络设备都可以验证其对等设备的证书和可信赖性。NDAC 会使用 IEEE 802.1X 基于端口认证的认证框架，并使用 EAP-FAST 作为它的 EAP 方法。在 NDAC 过程中认证和授权成功后，会为 IEEE 802.1AE 加密进行安全关联协议协商。 **注释：** Inspur 2960x 交换机上不支持该特性
安全关联协议（SAP）	在通过了 NDAC 认证后，安全关联协议（SAP）会为 TrustSec 对等体之间接下来的 MACSec 链路加密，自动协商密钥和加密套件。SAP 定义在 IEEE 802.11i 中
安全组标记（SGT）	安全组标记交换协议（SXP）。通过使用 SXP，

	不具备 TrustSec 硬件功能的设备能够为认证的用户和设备，从 Inspur 身份服务引擎（ISE）或 Inspur 安全访问控制系统（ACS）那里接收 SGT 属性。然后设备可以把源 IP 到 SGT 的绑定信息转发到具有 TrustSec 硬件功能的设备，继而为实施 SGACL 对源流量进行标记

当链路两端的设备都支持 802.1AE MACsec 时，它们会协商 SAP。请求方和认证方之间会进行 EAPOL 密钥交换，用来协商加密套件、交换安全参数和管理密钥。在这些任务的成功完成后，双方会建立安全关联（SA）。

根据用户软件版本和许可，以及链路硬件的支持，SAP 协商可以使用以下操作模式之一：

- Galois Counter Mode（GCM）——认证和加密
- GCM 认证（GMAC）——GCM 认证，无加密
- 无封装——无封装（明文）
- Null——封装，无认证，无加密

配置 Inspur TrustSec MACsec

在手动模式中配置 Inspur TrustSec 交换机到交换机链路安全

在开始前

当用户在接口上手动配置 Inspur TrustSec 特性时，需要考虑以下用法指导和限制条件：

- 如果没有定义 SAP 参数的话，Inspur TrustSec 特性就不会执行封装和加密；
- 如果选择 GCM 作为 SAP 运行模式的话，用户必须从 Inspur 获得 MACsec 加密软件许可。如果选择 GCM 但没有获得许可的话，接口会变为链路失效（link-down）状态；
- 在用户配置 SAP 成对的主密钥（sap pmk）时可以选择以下保护级别：
  - 未配置 SAP——无保护
  - **sap mode-list gcm-encrypt gmac no-encap**——期望提供保护，但并不强制
  - **sap mode-list gcm-encrypt gmac**——优选保密性和需要提供完整性。保护方式由请求方根据请求方的偏好进行选择
  - **sap mode-list gmac**——只提供完整性
  - **sap mode-list gcm-encrypt**——需要提供保密性

- **sap mode-list gmac gcm-encrypt——**需要并优选完整性，保密性是可选的

从特权 EXEC 模式开始，用户可以按照以下步骤，在接口上手动为另一台支持 Inspur TrustSec 的设备配置 Inspur TrustSec 特性：

总步骤

1. configure terminal

2. interface *interface-id*

3. cts manual

4. sap pmk *key* [**mode-list** *mode1* [*mode2* [*mode3* [*mode4*]]]]

5. no propagate sgt

6. exit

7. end

8. show cts interface [*interface-id* |**brief** |**summary**]

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Switch# configure terminal`	进入全局配置模式
步骤 2	**interface** *interface-id* **示例：** `Switch(config)# interface tengigabitethernet 1/1/2`	**注释：** 进入接口配置模式
步骤 3	**cts manual** **示例：** `Switch(config-if)# cts manual`	进入 Inspur TrustSec 手动配置模式
步骤 4	**sap pmk** *key* [**mode-list** *mode1* [*mode2* [*mode3* [*mode4*]]]] **示例：** `Switch(config-if-cts-manual)#`	（可选）配置 SAP 成对的主密钥（PMK）和运行模式。在 Inspur TrustSec 手动模式中，SAP 默认是禁用的。 • *key——*十六进制数值，字符长度

	sap pmk 1234abcdef mode-list gcm-encrypt null no-encap	为偶数，最长为 32 个字符 SAP 运行模式的选项： • **gcm-encrypt——**认证和加密 **注释：** 如果用户的软件许可支持 MACsec 加密的话，使用这个模式来实现 MACsec 认证和加密。 • **gmac——**认证，无加密 • **no-encap**——无封装 • **null**——封装，无认证，无加密 **注释：** 如果接口不支持数据链路加密的话，**no-cap** 就是默认设置，也是唯一可用的 SAP 运行模式。不支持 SGT
步骤 5	**no propagate sgt** **示例：** `Switch(config-if-cts-manual)# no propagate sgt`	如果对等体设备无法处理 SGT，用户需要使用这条命令的 **no** 形式。**no propagate sgt** 命令会阻止接口向对等体发送 SGT
步骤 6	**exit** **示例：** `Switch(config-if-cts-manual)# exit`	退出 Inspur TrustSec 802.1x 接口配置模式
步骤 7	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 8	**show cts interface** [*interface-id* **|brief |summary**]	（可选）通过查看与 TrustSec 相关的接口特征，检查用户输入的配置

以下示例展示了如何在接口上，以手动模式配置 Inspur TrustSec 认证特性：

```
Switch# configure terminal
Switch(config)# interface tengigabitethernet 1/1/2
Switch(config-if)# cts manual
```

```
Switch(config-if-cts-manual)# sap pmk 1234abcdef mode-list
gcm-encrypt null no-encap
Switch(config-if-cts-manual)# no propagate sgt
Switch(config-if-cts-manual)# exit
Switch(config-if)# end
```

配置示例

在接口上配置 MACsec

用户可以按照以下步骤在接口上配置 MACsec，一个 MACsec 会话用于语音，另一个用于数据：

总步骤

1. enable

2. configureterminal

3. interface *interface-id*

4. switchport access vlan*vlan-id*

5. switchport mode access

6. macsec

7. authentication event linksec fail action authorize vlan *vlan-id*

8. authentication host-mode multi-domain

9. authentication linksec policy must-secure

10. authentication port-control auto

11. authentication periodic

12. authentication timer reauthenticate

13. authentication violation protect

14. mka policy *policy name*

15. dot1x pae authenticator

16. spanning-tree portfast

17. end

18. show authentication session interface *interface-id*

19. show authentication session interface *interface-id* details

20. show macsec interface *interface-id*

21. show mka sessions

22. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Switch> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Swich# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id*	标识 MACsec 接口，并进入接口配置模式。这个接口必须是物理接口
步骤 4	**switchport access vlan***vlan-id*	为端口配置一个 Access VLAN
步骤 5	**switchport mode access**	把接口配置为 Access 端口
步骤 6	**macsec**	在接口上启用 802.1ae MACsec。macsec 命令至在交换机到主机之间的链路（下行端口）上启用 MKA MACsec
步骤 7	**authentication event linksec fail action authorize vlan** *vlan-id*	（可选）通过配置，让交换机在认证尝试失败后，为端口授权一个受限 VLAN，用来处理未授权用户的认证链路安全失败事件
步骤 8	**authentication host-mode multi-domain**	在端口上配置认真管理器模式，允许一台主机和一台语音设备在 802.1x 授权的端口上进行认证。如果没有配置这条命令，默认的主机模式是单主机模式
步骤 9	**authentication linksec policy must-secure**	设置 LinkSec 安全策略，在对等体可用时使用 MACsec 特性来保护会话的安全。如果没有设置这条命令，默认行为

		是*应该提供安全保护*
步骤 10	**authentication port-control auto**	在端口上启用 802.1x 认证。端口会根据交换机和客户端之间交换的认证结果，来改变授权或未授权状态
步骤 11	**authentication periodic**	为这个端口启用或禁用重认证功能
步骤 12	**authentication timer reauthenticate**	输入 1 至 65535（以秒为单位）之间的值。从服务器获得重认证超时时间。默认的重认证时间是 3600 秒
步骤 13	**authentication violation protect**	配置端口在发生以下事件时丢弃入站 MAC 地址：新设备连接到端口，或端口上已连接了最大数量的设备后又连接了新设备。如果没有配置这条命令，默认行为是关闭（shutdown）端口
步骤 14	**mka policy** *policy name*	在接口上应用现有的 MKA 协议策略，并在接口上启用 MKA。如果用户没有通过全局配置命令 **mka policy** 配置 MKA 策略的话，
步骤 15	**dot1x pae authenticator**	把端口配置为 802.1x 端口访问实体（PAE）认证器
步骤 16	**spanning-tree portfast**	在接口上为其关联的所有 VLAN 都启用生成树 Post Fast 特性。在启用了 Port Fast 特性后，接口会直接从阻塞状态进入转发状态，无需经历生成树中间状态
步骤 17	**end** **示例：** `Switch(config)#end`	返回特权 EXEC 模式
步骤 18	**show authentication session interface** *interface-id*	检查授权的会话安全状态
步骤 19	**show authentication session interface** *interface-id* **details**	检查授权会话的安全状态详情
步骤 20	**show macsec interface** *interface-id*	检查接口上的 MACsec 状态

步骤 21	**show mka sessions**	检查已建立的 MKA 会话
步骤 22	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

Inspur TrustSec 交换机到交换机链路安全配置示例

这个示例展示了为实现 Inspur TrustSec 交换机到交换机安全性，种子设备和非种子设备上所必需的配置。用户必须为链路安全性配置 AAA 和 RADIUS。在这个示例中，ACS-1 到 ACS-3 可以是任何名称的服务器，cts-radius 是 Inspur TrustSec 服务器。

种子设备上的配置：

```
Switch(config)#aaa new-model
Switch(config)#radius server ACS-1
Switch(config-radius-server)#address ipv4 10.5.120.12 auth-port 1812
acct-port 1813
Switch(config-radius-server)#pac key
inspur123 Switch(config-radius-server)#exit
Switch(config)#radius server ACS-2
Switch(config-radius-server)#address ipv4 10.5.120.14 auth-port 1812
acct-port 1813
Switch(config-radius-server)#pac key
inspur123 Switch(config-radius-server)#exit
Switch(config)#radius server ACS-3
Switch(config-radius-server)#address ipv4 10.5.120.15 auth-port 1812
acct-port 1813
Switch(config-radius-server)#pac key inspur123
Switch(config-radius-server)#exit
Switch(config)#aaa group server radius cts-radius
Switch(config-sg-radius)#server name ACS-1
Switch(config-sg-radius)#server name ACS-2
Switch(config-sg-radius)#server name ACS-3
```

```
Switch(config-sg-radius)#exit
Switch(config)#aaa authentication login default none
Switch(config)#aaa authentication dot1x default group cts-radius
Switch(config)#aaa authorization network cts-radius group cts-radius
Switch(config)#aaa session-id common
Switch(config)#cts authorization list cts-radius
Switch(config)#dot1x system-auth-control
Switch(config)#interface gi1/1/2
Switch(config-if)#switchport mode trunk
Switch(config-if)#cts dot1x
Switch(config-if-cts-dot1x)#sap mode-list gcm-encrypt gmac
Switch(config-if-cts-dot1x)#exit
Switch(config-if)#exit
Switch(config)#interface gi1/1/4
Switch(config-if)#switchport mode trunk
Switch(config-if)#cts manual
Switch(config-if-cts-dot1x)#sap pmk 033445AABBCCDDEEFF mode-list
gcm-encrypt gmac
Switch(config-if-cts-dot1x)#no propagate sgt
Switch(config-if-cts-dot1x)#exit
Switch(config-if)#exit
Switch(config)#radius-server vsa send authentication
Switch(config)#end
Switch#cts credentials id cts-36 password trustsec123
```

非种子设备上的配置：

```
Switch(config)#aaa new-model
Switch(config)#aaa session-id common
Switch(config)#dot1x system-auth-control
Switch(config)#interface gi1/1/2
Switch(config-if)#switchport mode trunk
Switch(config-if)#shutdown
Switch(config-if)#cts dot1x
Switch(config-if-cts-dot1x)#sap mode-list gcm-encrypt gmac
```

```
Switch(config-if-cts-dot1x)#exit
Switch(config-if)#exit
Switch(config)#interface gi1/1/4
Switch(config-if)#switchport mode trunk
Switch(config-if)#shutdown
Switch(config-if)#cts manual
Switch(config-if-cts-dot1x)#sap pmk 033445AABBCCDDEEFF mode-list
gcm-encrypt gmac
Switch(config-if-cts-dot1x)#no propagate sgt
Switch(config-if-cts-dot1x)#exit
Switch(config-if)#exit
Switch(config)#radius-server vsa send authentication
Switch(config)#end
Switch(config)#cts credentials id cts-72 password trustsec123
```

配置 RADIUS

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置 RADIUS 的先决条件

这一部分列出了使用 RADIUS 来控制设备访问行为的先决条件：

通常情况下：

- 要想使用本章中提到的任何配置命令，用户必须启用 RADIUS 和认证、授权和审计（AAA）；
- RADIUS 是通过 AAA 实现的，并且只能通过 AAA 命令进行启用；
- 用户需要使用全局配置命令 **aaa new-model** 来启用 AAA；
- 用户需要使用全局配置命令 **aaa authentication** 来为 RADIUS 认证定义方法列表；
- 用户需要使用 **line** 和 **interface** 命令，来启用定义好的方法列表；
- 最起码用户必须对运行 RADIUS 服务器软件的一台或多台主机进行标识，并定义 RADIUS 认证的方法列表。用户可以（可选的）为 RADIUS 授权和审计定义方法列表；
- 在用户的设备上配置 RADIUS 特性之前，用户应该能够访问 RADIUS 服务器，并且应该对其进行配置；
- RADIUS 主机通常是运行 RADIUS 服务器软件的多用户系统，这种软件可能会来自于 Inspur（Inspur 安全访问控制服务器 3.0 版本）、Livingston、Merit、Microsoft，或其他软件厂商。更多信息用户可以参考 RADIUS 服务器文档；
- 要想使用授权变更（CoA）接口，交换机上必须已经存在一个会话。CoA 可以用来标识一个会话，并执行断开连接请求。更新只会对指定会话带来影响；
- 建议在交换机堆栈和 RADIUS 服务器之间建立冗余的连接。这样做有助于在连接 RADIUS 服务器的堆栈成员从交换机堆栈中移除后，保障 RADIUS 服务器仍可访问。

对于 RADIUS 的运行：

- 用户必须首先成功完成 RADIUS 认证，才能继续使用 RADIUS 授权（如果启用了的话）。

配置 RADIUS 的限制条件

这个主题中包含了使用 RADIUS 来控制设备访问行为的限制条件：

通常情况下：

- 为了防止安全性中断，用户不能通过网络管理应用程序来配置 RADIUS。

RADIUS 不适用于下列网络安全环境：

- 多协议访问环境。RADIUS 不支持 AppleTalk 远程访问（ARA）、NetBINOS 数据帧控制协

议（NBFCP）、NetWare 异步服务接口（NASI）或 X.25 PAD 连接；

- 交换机到交换机或路由器到路由器的环境。RADIUS 不提供双向认证。如果非 Inspur 设备请求进行身份验证，RADIUS 可以提供从 Inspur 设备到非 Inspur 设备的身份验证；
- 使用各种服务的网络。RADIUS 通常会把用户绑定到一个服务模型。

RADIUS 的相关信息

RADIUS 和交换机访问

这一部分介绍了如何启用和配置 RADIUS。RADIUS 为认证和授权过程，提供了详细的审计信息和灵活的管理控制。

RADIUS 概述

RADIUS 是一种分布式客户端/服务器系统，它可以保护网络免遭未经授权的访问。RADIUS 客户端运行在支持的 Inspur 路由器和交换机上。客户端会向中心 RADIUS 服务器发送认证请求，其中包含所有用户认证和网络服务访问的相关信息。

用户可以在以下这些需要访问安全性的网络环境中使用 RADIUS：

- 部署了多厂商访问服务器的网络，其中所有服务器都支持 RADIUS。举例来说，来自不同厂商的访问服务器可以使用单个 RADIUS 服务器的安全数据库。在部署了多厂商接入服务器的 IP 网络中，使用 RADIUS 服务器对拨入用户进行认证，并且这台 RADIUS 服务器已经通过自定义，与 Kerberos 安全系统一起工作；
- 一站式网络安全环境，其中的应用支持 RADIUS 协议，比如使用*智能卡*访问控制系统的接入环境。在一种情况下，用户可以把 RADIUS 与 Enigma 安全卡一起使用，来验证用户并授予其对网络资源的访问权限；
- 已经在使用 RADIUS 的网络。用户可以将包含 RADIUS 客户端的 Inspur 设备添加到网络中。这可能是用户切换为使用 TACACS+服务器的第一步，用户可以参考下文中的图 2：从 RADIUS 切换到 TACACS+服务；
- 限制用户只能访问单个服务的网络。使用 RADIUS 可以控制让用户只访问单台主机、只访问单个应用（比如 Telnet），或者通过诸如 IEEE 802.1x 之类的协议来访问网络。有关这个协议的详细信息，用户可以参考第 11 章“配置 IEEE 802.1x 基于端口的认证”；

- 需要进行资源审计的网络。用户可以在 RADIUS 身份认证或授权之外，独立使用 RADIUS 审计功能。RADIUS 审计功能允许在服务的开始和结束时发送数据，并显示会话期间使用的资源（例如时间、数据包、字节等）。Internet 服务提供商可能会使用免费软件版本的 RADIUS 访问控制和审计软件，来满足特殊的安全性和审计需求。

图 101：从 RADIUS 切换到 TACACS+服务

RADIUS server（共 2 处）	RADIUS 服务器
TACACS+ server（共 2 处）	TACACS+ 服务器
Remote PC	远端 PC
Workstation	工作站

RADIUS 的工作原理

当用户尝试登录一台设备并进行认证时，设备如果是由 RADIUS 服务器提供访问控制的，就会发生以下事件：

1. 用户会看到输入用户名和密码的提示；
2. 通过网络把用户名和加密密码发送到 RADIUS 服务器的；
3. 用户从 RADIUS 服务器接收到以下响应之一：
    - ACCEPT（接受）——用户通过认证
    - REJECT（拒绝）——用户没有通过认证，并看到再次输入用户名和密码的提示，或者访问被拒绝
    - CHALLENGE（质询）——从用户质询请求其他数据
    - CHALLENGE PASSWORD——这个响应是要求用户选择一个新密码

ACCEPT（接受）或 REJECT（拒绝）响应是与用来进行特权 EXEC 或网络授权的其他数据捆绑在一起的。ACCEPT（接受）或 REJECT（拒绝）数据包中包含以下其他数据：

- Telnet、SSH、远程登录，或者特权 EXEC 服务
- 连接参数，其中包括主机或客户端 IP 地址、访问列表，以及用户超时时间

RADIUS 授权变更

RADIUS 授权变更（CoA）提供了一种机制，能够在认证成功后更改认证、授权和审计（AAA）

的属性。在 AAA 中为用户或用户组变更策略时，管理员会从 AAA 服务器发送 RADIUS CoA 数据包，以此来初始化认证并执行信息的策略，Inspur 安全访问控制服务器（ACS）就可以充当 AAA 服务器。这一部分概述了 RADIUS 接口，其中包括可用的原函数，以及如何在 CoA 中使用它们。

- 授权变更请求
- CoA 请求响应代码
- CoA 请求命令
- 会话重认证
- 有关会话终结的堆叠指导

标准的 RADIUS 接口通常用于拉取模型中，在这种模型中，请求源自于联网设备，响应来自于查询服务器。Inspur 支持 RFC 5176 中定义的 RADIUS CoA 扩展，它通常用于推送模型中，允许从外部 AAA 服务器或策略服务器进行动态的会话重配置。

交换机支持以下每会话 CoA 请求：

- 会话重认证
- 会话终结
- 会话终结及端口关闭（shutdown）
- 会话终结及端口反弹（bounce）

这个特性集成在 Inspur 安全访问控制服务器（ACS）5.1 中。

RADIUS 接口在 Inspur 交换机上是默认启用的。但用户仍需为以下属性执行基本配置：

- 安全和密码——参考这个指南中“在交换机上预防未授权访问”一节
- 审计——参考这个指南中配置基于交换机的认证一章中“开始使用 RADIUS 审计”一节

Inspur INOS 软件支持 RFC 5176 中定义的 RADIUS CoA 扩展，它通常用于推送模型中，允许从外部 AAA 服务器或策略服务器进行动态的会话重配置。它为每个会话支持 CoA 请求，能够实现会话标识、会话终结、主机重认证、端口关闭和端口反弹。这个模型中包含一个请求（CoA 请求）和两个可能用到的响应代码：

- CoA 确认（ACK）[CoA-ACK]
- CoA 未确认（NAK）[CoA-NAK]

请求是从 CoA 客户端（通常是 AAA 服务器或策略服务器）发起的，并发送给充当侦听方的设备。

下面这个表格中展示了基于身份识别的网络服务所支持的 RADIUS CoA 命令和厂商指定的属性（VSA）。所有 CoA 命令中都必须包含设备和 CoA 客户端之间的会话标识。

表 126：基于身份识别的网络服务所支持的 RADIUS CoA 命令

CoA 命令	Inspur VSA
激活服务	Inspur:Avpair="subscriber:command=activate-service" Inspur:Avpair="subscriber:service-name=<*service-name*>" Inspur:Avpair="subscriber:precedence=<*precedence-number*>" Inspur:Avpair="subscriber:activation-mode=replace-all"
停用服务	Inspur:Avpair="subscriber:command=deactivate-service" Inspur:Avpair="subscriber:service-name=<service-name>"
反弹主机端口	Inspur:Avpair="subscriber:command=bounce-host-port"
禁用主机端口	Inspur:Avpair="subscriber:command=disable-host-port"
会话查询	Inspur:Avpair="subscriber:command=session-query"
会话重认证	Inspur:Avpair="subscriber:command=reauthenticate" Inspur:Avpair="subscriber:reauthenticate-type=last"或者 Inspur:Avpair="subscriber:reauthenticate-type=rerun"
会话终结	这是标准的断开连接请求，并不需要 VSA
接口模版	Inspur:AVpair="interface-template-name=<*interfacetemplate*>"

授权变更请求

授权变更（CoA）请求定义在 RFC 5176 文档中，它通常用于推送模型中，实现了会话身份识别、主机重认证和会话终结。这个模型中包含一个请求（CoA 请求）和两个可能用到的响应代码：

- CoA 确认（ACK）[CoA-ACK]
- CoA 未确认（NAK）[CoA-NAK]

请求是从 CoA 客户端（通常是 AAA 服务器或策略服务器）发起的，并发送给充当侦听方的设备。

RFC 5176 规范

连接断开请求消息也称为数据包断开连接（POD），交换机支持使用它来完成会话终结工作。下面这个表格中展示了这个特性中支持的 IETF 属性。

表 127：支持的 IETF 属性

属性编号	属性名称
24	状态
31	Calling-Station-Id

44	Acct-Session-ID
80	消息认证器
101	错误原因

下面这个表格中展示了可能出现的错误原因属性值。

表 128：错误原因值

值	解释
201	残留的会话上下文已删除
202	无效的 EAP 数据包（已忽略）
401	不支持的属性
402	丢失的属性
403	NAS 身份不匹配
404	无效的请求
405	不支持的服务
406	不支持的扩展
407	无效的属性值
501	行政上禁止
502	请求无法路由（代理）
503	未找到会话上下文
504	未删除会话上下文
505	其他代理处理错误
506	资源不可用
507	请求已初始
508	不支持多会话选择

CoA 请求响应代码

CoA 请求响应代码可以用来向交换机传达一个命令。

CoA 请求响应代码使用的数据包格式定义在 RFC 5176 文档中，由以下字段构成：代码、识别符、长度、认证器和属性，使用的是类型:长度:值（TLV）格式。属性字段用来承载 Inspur 厂商定义的属性（VSA）。

会话身份识别

为了针对某个会话断开连接和发送 CoA 请求，交换机会根据以下属性之一来定位这个会话：

- Acct-Session-Id（IETF 属性 44）
- Audit-Session-Id（Inspur VSA）
- Calling-Station-Id（IETF 属性 31，其中包含主机 MAC 地址）
- IPv6 属性，可以是以下属性之一：
  - Framed-IPv6-Prefix（IETF 属性 97）和 Framed-Interface-Id（IETF 属性 96），它们加载一起构成了完整的 IPv6 地址，定义在 RFC 3162 文档中
  - Framed-IPv6-Address
- Plain IP Address（IETF 属性 8）

除非 CoA 消息中的所有会话标识属性都与会话相匹配，否则交换机会返回携带“无效属性值”错误代码属性的 Disconnect-NAK 或 CoA-NAK。

如果消息中包含多个会话标识属性，则所有属性必须都与会话相匹配，否则交换机会返回携带错误代码“无效属性值”的断开否定确认（NAK）或 CoA-NAK。

RFC 5176 中定义了 CoA 请求代码的数据包格式，其中包含以下字段：代码、身份标识、长度、认证器、以及属性，并使用类型:长度:值（TLV）格式。

Code	代码
Identifier	识别符
Length	长度
Authenticator	认证器
Attributes	属性

属性字段用来承载 Inspur 厂商指定的属性（VSA）。

对于针对特定策略的 CoA 请求，如果消息中包含上述任意会话标识属性，设备就会返回携带错误代码“无效属性值”的 CoA-NAK。

CoA ACK 响应代码

如果授权状态成功地改变了，则发送肯定确认（ACK）。在 CoA ACK 中返回的属性，根据 CoA 请求的变化而变化，这些内容会单独在 CoA 命令中进行讨论。

CoA NAK 响应代码

否定确认（NAK）标识无法改变授权状态，并且其中可以包含指示出失败原因的属性。用户可以使用 **show** 命令验证 CoA 的成功。

CoA 请求命令

表 129：交换机上支持的 CoA 命令

命令[8]	Inspur VSA
重认证主机	Inspur:Avpair="subscriber:command=reauthenticate"
终结会话	这是标准的断开连接请求，并不需要 VSA
反弹主机端口	Inspur:Avpair="subscriber:command=bounce-host-port"
禁用主机端口	Inspur:Avpair="subscriber:command=disable-host-port"

[8] 所有 CoA 命令中都必须包含交换机和 CoA 客户端之间的会话识别符。

会话重认证

当未知身份的主机加入网络，并且与受限的访问授权配置文件（例如访客 VLAN）相关联时，AAA 服务器通常生成一个会话重认证请求。当证书已知时，重新认证请求能够把主机放置在适当的授权组中。

为了初始化会话认证，AAA 服务器会发送一个标准 CoA 请求消息，其中会包含一个以下形式的 Inspur VSA：*Inspur:Avpair ="subscriber:command = reauthenticate"*，以及一个或多个会话标识属性。

当前会话的状态决定了交换机对消息的响应。如果会话当前通过了 IEEE 802.1x 认证，则交换机会通过向服务器发送一个 EAPoL（基于局域网的可扩展认证协议）请求 ID 消息来进行响应。

如果会话当前是通过 MAC 认证旁路（MAB）进行认证的，则交换机会向服务器发送访问请求，向其传递在初始认证成功时使用的相同身份属性。

如果交换机接收到该命令时会话认证仍在进行中，则交换机会终止认证过程，并重新开始认证序列，从配置中第一个尝试的方法重新开始。

如果会话尚未进行授权，或者被授权为访客 VLAN、重要 VLAN，或类似的策略，则重认证消息会重新开始执行访问控制方法，从配置中第一个尝试的方法重新开始。交换机会维持会话的当前授权状态，直到重新认证得出了不同的授权结果。

交换机堆栈中的会话重认证

当交换机堆栈接收到一个会话重认证消息后：

- 它会在返回确认（ACK）之前，检查是否需要重认证；
- 它会为适当的会话初始化重认证进程；
- 如果认证的结果是成功或失败，则触发重新认证的信号会从堆叠成员上移除；
- 如果堆叠主用设备在认证完成之前失效了，则在主用设备切换之后，会根据原始命令（随后会被移除）初始化重认证进程；
- 如果堆叠主用设备在发送 ACK 之前失效了，则新的主用设备会把它作为新命令进行重传。

会话终结

有三种类型的 CoA 请求可以触发会话终结进程。CoA Disconnect-Request 会终止会话，并且不禁用主机端口。这个命令会为指定主机重新初始化身份认证器状态机，但不会对主机访问网络进行限制。

要想限制主机对网络的访问，要使用一个 CoA 请求，以及 Inspur:Avpair="subscriber:command=disable-host-port" VSA。当已知主机在网络上引发问题，且用户需要立即阻止主机访问网络时，这个命令就很有用。用户要想恢复端口的网络访问功能，要使用非 RADIUS 机制重新启用端口。

当设备不是请求方（比如打印机）时，若需要获取新的 IP 地址（比如在 VLAN 发生变更后），用户要使用端口反弹特性来终止主机端口上的会话（暂时禁用，然后再重新启用端口）。

CoA 连接断开请求

这个命令是一个标准的 Disconnect-Request（断开连接请求）。如果无法定位会话的话，交换机会返回携带“未找到会话上下文”错误代码属性的 Disconnect-NAK 消息。如果定位了会话，交换机就会终止会话。在完全删除会话后，交换机会返回 Disconnect-ACK。

如果交换机在向客户端返回 Disconnect-ACK 之前，因故障切换到备用交换机，则当客户端重新发送请求时，在新的活跃交换机上会重复同样的过程。如果在重新发送之后并没有找到会话，则交换机会发送携带“未找到会话上下文”错误代码属性的 Disconnect-ACK。

CoA 请求：禁用主机端口

RADIUS 服务器发送的 CoA 禁用端口命令，能够把已经建立了会话的认证端口变为管理关闭状态，并导致会话被终止。当已知主机在网络上引发问题，且用户需要立即阻止主机访问网络时，这个命令就很有用。用户要想恢复端口的网络访问功能，要使用非 RADIUS 机制重新启用端口。这个命令承载在一个标准 CoA 请求消息中，其中包含以下这个新的厂商指定属性（VSA）：

Inspur:Avpair="subscriber:command=disable-host-port"

因为这个命令是面向会话的，所以它必须与“会话标识”部分中描述的一个或多个会话标识属性结合在一起使用。如果无法定位会话的话，交换机会返回携带“未找到会话上下文”错误代码属性的 CoA-NAK 消息。如果定位了会话，交换机就会禁用主机端口并返回 CoA-ACK 消息。

如果交换机在向客户端返回 CoA-ACK 之前发生故障，则当客户端重新发送请求时，在新的活跃交换机上会重复同样的过程。如果交换机在向客户端返回 CoA-ACK 之后，但整个过程还未结束之前发生故障，则新的活跃交换机上会重新开始同样的过程。

注释： 在命令重新发送后的断开连接请求失败事件，可能是由切换前成功的会话终结事件导致的（如果未发送 Disconnect-ACK），或者是由其他方式（比如链路故障）实现的会话终

结事件导致的；这些事件发生在发出原始命令之后，且在备用交换机变为活跃状态之前。

CoA 请求：反弹端口

从 RADIUS 服务器发送的 RADIUS 服务器 CoA 反弹端口消息，可能会导致认证端口上的链路发生翻动，从而触发连接到此端口的一个或多个主机重新进行 DHCP 协商。当有 VLAN 发生变更，且端点设备（如打印机）无法检测此认证端口上的变更时，就可能会发生这个事件。

CoA 反弹端口承载在标准 CoA 请求消息中，其中包含以下 VSA：

Inspur:Avpair="subscriber:command=bounce-host-port"

因为这个命令是面向会话的，所以它必须与一个或多个会话标识属性结合在一起使用。如果无法定位会话的话，交换机会返回携带“未找到会话上下文”错误代码属性的 CoA-NAK 消息。如果定位了会话，交换机就会禁用主机端口 10 秒钟的时间，之后重新启用它（端口反弹）并返回 CoA-ACK 消息。

如果交换机在向客户端返回 CoA-ACK 之前发生故障，则当客户端重新发送请求时，在新的活跃交换机上会重复同样的过程。如果交换机在向客户端返回 CoA-ACK 之后，但整个过程还未结束之前发生故障，则新的活跃交换机上会重新开始同样的过程。

有关会话终结的堆栈指导

不需要对交换机堆栈中的 CoA Disconnect-Request 消息执行特殊处理。

有关 CoA 请求反弹端口的堆栈指导

因为 **bounce-port** 命令针对的是会话而不是端口，所以如果找不到会话的话，就无法执行这个命令。

当堆叠主用设备上的 Auth Manager（认证管理器）命令处理程序，接收到一个有效的 **bounce-port** 命令时，它会在返回 CoA-ACK 消息之前检查以下信息：

- 是否需要端口反弹
- 端口 ID（在本地会话上下文中查找）

交换机初始化端口反弹行为（禁用端口 10 秒钟，然后重新启用它）。

如果端口反弹操作成功，触发端口反弹的信号就会从备用堆叠中的主用设备中删除。

如果在端口反弹完成之前堆栈主用设备失效了，则在堆栈主用设备切换后，会根据原始命令（其随后被移除）初始化端口反弹。

如果在发送 CoA-ACK 消息之前堆叠主用设备失效了，则新的主用设备会把它作为新命令进行重传。

有关 CoA 请求禁用端口的堆栈指导

因为 **disable-port** 命令针对的是会话而不是端口，所以如果找不到会话的话，就无法执行这

个命令。

当堆叠主用设备上的 Auth Manager（认证管理器）命令处理程序，接收到一个有效的 **disable-port** 命令时，它会在返回 CoA-ACK 消息之前检查以下信息：

- 是否需要端口禁用
- 端口 ID（在本地会话上下文中查找）

交换机尝试禁用端口。

如果端口禁用操作成功，触发端口禁用的信号就会从备用堆叠中的主用设备中删除。

如果在端口禁用完成之前堆栈主用设备失效了，则在堆栈主用设备切换后，会根据原始命令（其随后被移除）初始化端口禁用。

如果在发送 CoA-ACK 消息之前堆叠主用设备失效了，则新的主用设备会把它作为新命令进行重传。

默认的 RADIUS 配置

RADIUS 和 AAA 默认都是禁用的。

为了防止安全性失效，用户不能通过网络管理应用程序来配置 RADIUS。当启用了 RADIUS 时，RADIUS 可以验证通过 CLI 访问交换机的用户。

RADIUS 服务器主机

交换机到 RADIUS 服务器的通信中涉及多个组成部分：

- 主机或 IP 地址
- 认证目的端口
- 审计目的端口
- 密钥字符串
- 超时周期
- 重传值

用户可以通过主机名或 IP 地址、主机名和特定 UDP 端口号，或 IP 地址和特定 UDP 端口号来标识 RADIUS 安全服务器。IP 地址和 UDP 端口号的组合会创建出唯一的标识符，使不同的端口能够被单独定义为提供特定 AAA 服务的 RADIUS 主机。通过使用这个唯一的标识符，使 RADIUS 请求能够发送到服务器（IP 地址相同）上的多个 UDP 端口。

如果用户把同一台 RADIUS 服务器上的两个不同主机条目配置为相同的服务（比如审计），

则用户配置的第二个主机条目会充当第一个主机条目的故障切换备份设备。举例来说，如果第一主机条目无法提供记帐服务了，则会显示%RADIUS-4-RADIUS_DEAD 消息，然后交换机会尝试使用同一台设备上提供审计服务的第二主机条目（按照配置的顺序尝试 RADIUS 上的主机条目）。

RADIUS 服务器和交换机会使用共享秘密文本字符串来加密密码和交换的响应消息。要想配置 RADIUS 来使用 AAA 安全命令，用户必须指定运行 RADIUS 服务器守护程序的主机，以及与交换机共享的秘密文本（密钥）字符串。

用户可以为所有 RADIUS 服务器在全局配置超时、重传和加密密钥值，可以以每个服务器为基础进行配置，也可以使用全局设置和服务器设置的不同组合。

RADIUS 登录认证

要想配置 AAA 认证，用户需要定义一个命名的认证方法列表，并把它应用给各种端口。方法列表中定义了要执行的认证的类型，以及执行它们的顺序；在它能够执行任何定义的身份验证方法之前，用户必须把它应用在特定的端口上。唯一的例外是默认方法列表。默认方法列表会自动应用在所有端口上，除了已经明确定义了方法列表名称的端口。

方法列表定义了为用户进行认证、授权或审计的序列和方法。用户可以使用方法列表来指定要使用的一个或多个安全协议，这样做可以在初始方法失败时，确保有一个备份系统。软件使用列表中的第一个方法来对用户进行认证、授权或审计；如果该方法没有获得响应，软件会选择列表中的下一个方法。这个过程会持续直到使用列出的方法实现成功通信，或者持续到方法列表耗尽。如果在这个周期中的任何时刻认证失败了——意味着安全服务器或本地用户名数据库发出了响应，拒绝了用户的访问——这时认证过程就停止了，并且不会再尝试其他认证方法。

AAA 服务器组

用户可以配置交换机来使用单台 AAA 服务器或 AAA 服务器组，为现有的服务器主机进行身份认证。用户可以有选择地把一部分服务器主机设置为一组服务器，并将其用于提供特定服务。服务器组与全局服务器主机列表一起使用，还包含所选服务器主机的 IP 地址列表。

如果每个条目都拥有一个唯一的标识符（IP 地址和 UDP 端口号的组合），服务器组中还可以为同一台服务器包含多个主机条目，并且能够把不同端口单独定义为提供特定 AAA 服务的 RADIUS 主机。通过使用这个唯一的标识符，使 RADIUS 请求能够发送到服务器（IP 地址相同）上的多个 UDP 端口。如果用户把同一台 RADIUS 服务器上的两个不同主机条目配置为相同的

服务（比如审计），则用户配置的第二个主机条目会充当第一个主机条目的故障切换备份设备。举例来说，如果第一主机条目无法提供记帐服务了，则交换机会尝试使用同一台设备上提供审计服务的第二主机条目（按照配置的顺序尝试 RADIUS 上的主机条目）。

AAA 授权

AAA 授权能够限制用户可以使用的服务。当用户启用了 AAA 授权后，交换机会使用从用户配置文件中检索的信息来配置用户的会话，这个配置文件位于本地用户数据库中，或位于安全服务器。只有当用户配置文件中的信息允许时，用户才能够访问所请求的服务。

RADIUS 审计

AAA 审计功能会跟踪用户正在访问的服务，以及用户消耗的网络资源总量。当用户启用了 AAA 审计后，交换机以审计记录的形式向 RADIUS 安全服务器报告用户的活动。每个审计记录中都包含了审计属性值（AV）对，并且这些信息存储在安全服务器上。之后用户可以使用这些数据来对网络管理、客户端计费或审计进行分析。

厂商指定的 RADIUS 属性

Internet 工程任务组（IETF）草案标准中指定了一种通过使用厂商特定的属性（属性 26），在交换机和 RADIUS 服务器之间传送厂商特定信息的方法。厂商特定属性（VSA）能够使厂商支持自己的扩展属性，但不适合一般使用。Inspur RADIUS 通过使用规范中推荐的格式，来支持厂商特定的选项。Inspur 的厂商 ID 为 9，支持的选项具有供应商类型 1，名为 *inspur-avpair*。这个值是使用以下格式的字符串：

protocol : attribute sep value *

protocol 是用于特定授权类型的 Inspur 协议属性。*attribute* 和 *value* 是定义在 Inspur TACACS+ 规范中的适当属性值（AV）对，*sep* 是=，用来强制执行属性，*是可选属性。TACACS+授权中可用的完整特性集，都可用于 RADIUS 中。

举例来说，下面这个 AV 对会在 IP 授权期间（在 PPP 的 Internet 协议控制协议（IPCP）地址分配期间）激活 Inspur 的“多命名 IP 地址池”功能：

inspur-avpair= ”ip:addr-pool=first“

如果用户插入一个“*”，AV 对“ip:addr-pool=first”就会成为选项。需要注意的是，任何 AV 对都可以成为选项：

inspur-avpair= ”ip:addr-pool*first“

下面这个示例展示了如何让通过网络访问服务器登录用户，直接获得访问 EXEC 命令的权限：

inspur-avpair= ”shell:priv-lvl=15“

其他厂商也有它们各自唯一的厂商 ID、选项，以及相关联的 VSA。更多厂商 ID 和 VSA 的相关信息，用户可以参考 RFC 2138“Remote Authentication Dial-In User Service (RADIUS)”。

属性 26 中包含以下三个元素：

- 类型
- 长度
- 字符串（也称为数据）
  - Vendor-Id
  - Vendor-Type
  - Vendor-Length
  - Vendor-Data

下图展示了在属性 26“后面”封装的 VSA 数据包格式。

图 102：属性 26 后面封装的 VSA

Type	类型
Length	长度
Attributes-specific… (vendor-data)	指定属性 （厂商数据）

注释： 厂商负责指定它们的 VSA 所使用的格式。与属性相关的字段（也称为厂商数据）也取决于厂商对于属性的定义。

下面这个表格中描述了厂商指定 RADIUS IETF 属性表（下面第二个表格）中的重要字段，其中列出了支持的厂商指定 RADIUS 属性（IETF 属性 26）。

表 130：厂商指定的属性表字段描述

字段	**描述**
编号	下面表格中列出的所有属性都是 IETF 属性 26 的扩展
厂商指定的命令代码	用来标识具体厂商的代码。代码 9 定义了 Inspur VSA，311 定义了 Microsoft VSA，529 定义了 Ascend VSA
子类型编号	属性 ID 编号。这个编号与 IETF 属性的 ID 编号非常类似，只是它是封装在属性 26 后面的

	“第 2 层”ID 编号
属性	属性的 ASCII 字符串名称
描述	属性的描述

表 131：厂商指定的 RADIUS IETF 属性

编号	厂商指定的公司代码	子类型编号	属性	描述
MS-CHAP 属性				
26	311	1	MSCHAP-Response	在用于质询的响应消息中包含由 PPP MS-CHAP 用户提供的响应值。它仅用于 Access-Request（访问请求）数据包。这个属性与 PPP CHAP 标识符相同（RFC 2548）
26	311	11	MSCHAP-Challenge	包含由网络接入服务器向 MS-CHAP 用户发送的质询消息。它会用于 Access-Request（访问请求）和 Access-Challenge（访问质询）数据包中（RFC 2548）
VPDN 属性				
26	9	1	2tp-cm-local-window-size	为 L2TP 控制消息指定最大的接收窗口大小。这个值会在隧道建立期间通告给对等体
26	9	1	l2tp-drop-out-of-order	通过丢弃接收到的失序数据包，来遵从数据包的序列号。这并不保证数据包中一定会发送序列号，只是在接收到的时候进行控制
26	9	1	l2tp-hello-interval	为 Hello 存活间隔指定秒钟数。Hello 数据包会按照在这

				里配置的秒钟数，在没有数据发送时发送到隧道上
26	9	1	l2tp-hidden-avp	在启用后，L2TP 控制消息中的敏感 AVP 会被干扰或隐藏
26	9	1	l2tp-nosession-timeout	指定一个秒钟数，让隧道在这个时间值超时并关闭前，保持活跃
26	9	1	tunnel-tos-reflect	从每个负载数据包的 IP 头部复制 IP ToS 字段，并在 LNS 上进入隧道的数据包中，写入隧道数据包的 IP 头部
26	9	1	l2tp-tunnel-authen	如果设置了这个属性，就会执行 L2TP 隧道认证
26	9	1	l2tp-tunnel-password	用于 L2TP 隧道认证和 AVP 隐藏的共享秘密
26	9	1	l2tp-udp-checksum	这是一个授权属性，定义了 L2TP 是否应该为数据包执行 UDP 校验和。有效值是“yes”和“no”。默认值是 no
储存和转发传真属性				
26	9	3	Fax-Account-Id-Origin	指出系统管理员为 **mmoip aaa receive-id** 或 **mmoip aaa send-id** 命令定义的帐户 ID 源
26	9	4	Fax-Msg-Id=	指出由储存和转发传真分配的唯一传真消息识别编号
26	9	5	Fax-Pages	指出在这个传真会话中传输或接收的页数。页数中包含封面
26	9	6	Fax-Coverpage-Flag	指出出站网关是否为这个传真会话生成了封面。True 标

				识生成了封面；False 标识没有生成封面
26	9	7	Fax-Modem-Time	以秒为单位指出调制解调器发送传真数据（x）的时间，以及发送传真会话的总时长，其中包括传真邮件和 PSTN 时间，格式为 x/y。举例来说，10/15 标识传输时间花费了 10 秒钟，总传输会话花费了 15 秒钟
26	9	8	Fax-Connect-Speed	指出这个传真邮件初始化传输或接收时的调制解调器速率。可能的值包括 1200、4800、9600 和 14400
26	9	9	Fax-Recipient-Count	指出这个传真传输的接收方数量。直到电子邮件服务器支持会话模式，编号应该为 1
26	9	19	Fax-Process-Abort-Flag	指出传真会话的状态是终止还是成功。True 标识会话被终止；False 标识会话成功
26	9	11	Fax-Dsn-Address	指出要发送的 DSN 的地址
26	9	12	Fax-Dsn-Flag	指出是否启用了 DSN。True 表示已启用 DSN；False 表示未启用 DSN
26	9	13	Fax-Mdn-Address	指出要发送的 MDN 的地址
26	9	14	Fax-Mdn-Flag	指出是否启用了消息传递通知（MDN）。True 表示已启用 MDN；False 表示未启用 MDN
26	9	15	Fax-Auth-Status	指出这个传真会话的认证是否成功。这个字段的值可能

				是成功、失败、旁路或未知
26	9	16	Email-Server-Address	指出处理出站传真邮件消息的电子邮件服务器 IP 地址
26	9	17	Email-Server-Ack-Flag	指出入站网关已经从接收传真邮件消息的电子邮件服务器那里接收到了肯定的确认
26	9	18	Gateway-Id	指出处理传真会话的网关名称。名称显示为以下格式：主机名.域名
26	9	19	Call-Type	描述传真活动的类型：传真接收或传真发送
26	9	20	Port-Used	指出传输或接收这个传真邮件所使用的 Inspur AS5300 上的插槽/端口编号
26	9	21	Abort-Cause	如果传真会话终结，则指出发出终结信令的系统组成部分。能够触发终结事件的系统组成部分包括 FAP（传真应用进程）、TIFF（TIFF 阅读器或 TIFF 写入器）、传真邮件客户端、传真邮件服务器、ESMTP 客户端或 ESMTP 服务器
H323 属性				
26	9	23	Remote-Gateway-ID（H323 远端地址）	指出远端网关的 IP 地址
26	9	24	Connection-ID（h323 会议 ID）	标识会议 ID
26	9	25	Setup-Time（h323 建立时间）	指出这个连接建立的时间，标识为协调世界时间（UTC），以前称为格林威治时间（GMT）和 Zulu 时间

26	9	26	Call-Origin（h323 呼叫源）	指出与网关相关联的呼叫源。可能出现的值是发起和终结（应答）
26	9	27	Call-Type（h323 呼叫类型）	指出呼叫线路类型。可能出现的值是 **telephony** 和 **VoIP**
26	9	28	Connect-Time（h323 连接时间）	以 UTC 时间指出这条呼叫线路的连接时间
26	9	29	Disconnect-Time（h323 断开连接时间）	以 UTC 时间指出这条呼叫线路断开的时间
26	9	30	Disconnect-Cause（h323 连接断开原因）	根据 Q.931 定义，指定连接离线的原因
26	9	31	Voice-Quality（h323 语音质量）	指定影响一通呼叫语音质量的损伤因子（ICPIF）
26	9	32	Gateway-ID（h323 网关 ID）	指出底层网关的名称
大范围拨出属性				
26	9	1	callback-dialstring	为回拨定义一组拨号字符串
26	9	1	data-service	无描述信息可用
26	9	1	dial-number	定义拨叫号码
26	9	1	force-56	确定网络访问服务器是否只使用隧道的 56 K 部分，即使所有 65 K 都是可用的
26	9	1	map-class	让用户配置文件能够调用配置在 map-class 中的信息，网络访问服务器使用相同名称的 map-class 进行拨出行为
26	9	1	send-auth	定义 CLID 认证后，用户名密码认证使用的协议（PAP 或 CHAP）
26	9	1	send-name	PPP 名称认证。要想应用 PAP，不能在接口配置 **ppp**

				pap sent-name password 命令。对于 PAP 来说，“ preauth:send-name ”和“preauth:send-secret”会作为出向认证的 PAP 用户名和 PAP 密码。对于 CHAP 来说，“preauth:send-name”不仅会被用于出向认证，还会被用于入向认证。在 CHAP 入向环境中，NAS 会使用发往主叫方的质询数据包中，“preauth:send-name”中定义的名称。 **注释：** send-name 属性会发生变化：初始时，它提供现在由 send-name 和 remote-name 属性提供的功能。由于已经添加 remote-name 属性，因此 send-name 属性被限制为它当前的行为
26	9	1	send-secret	PPP 密码认证。厂商指定的属性（VSA）“ preauth:send-name ”和“preauth:send-secret”会被用于出向认证的 PAP 用户名和 PAP 密码。在 CHAP 出向环境中，“ preauth:send-name ”和“preauth:send-secret”会被用于响应数据包中
26	9	1	remote-name	提供用于大规模拨出的远程

				主机的名称。拨号程序会检查大规模拨出使用的远程名称是否与通过认证的名称相匹配，以防止发生用户 RADIUS 错误配置（举例来说，拨打有效的电话号码，但连接到错误的设备）
其他属性				
26	9	2	Inspur-NAS-Port	为NAS端口审计定义其他厂商指定的属性（VSA）信息。要想以属性值对（AV 对）格式指定其他 NAS 端口信息，用户要使用全局配置命令 **radius-server vsa send** **注释：** 这个 VSA 通常用于审计，但也可以用在认证（Access-Request）数据包中
26	9	1	min-links	为 MLP 设置最少链路数量
26	9	1	proxyacl#<n>	允许用户通过认证代理特性，配置下载用户配置文件（动态 ACL），使用户能够配置授权，来放行穿过指定接口的流量。
26	9	1	spi	承载归属代理（Home Agent）所需的认证信息，在注册期间对移动节点进行认证。这个信息与 **ip mobile secure host <addr>**配置命令的语法相同。基本上，它包含该字符串之后配置命令的其余部分。它提供了安全参数索引（SPI）、密钥、认证

				算法、认证模式和重放保护时间戳范围。

厂商私有的 RADIUS 服务器通信

尽管 IETF 在 RADIUS 标准草案中规定了交换机和 RADIUS 服务器之间传递厂商专有信息的方法，但一些厂商仍会以一种独特的方式对 RADIUS 属性集进行扩展。Inspur INOS 软件支持厂商私有 RADIUS 属性中的一个子集。

如前所述，要想配置 RADIUS（无论是厂商私有的或是符合 IETF 草案的），用户必须指定运行 RADIUS 服务器守护程序的主机及其与交换机共享的秘密文本字符串。用户可以使用全局配置命令 **radius server** 指定 RADIUS 主机和秘密文本字符串。

如何配置 RADIUS

标识 RADIUS 服务器主机

要想为与设备通信的所有 RADIUS 服务器设置全局配置，用户需要使用以下三个全局配置命令：**radius-server timeout**、**radius-server retransmit** 和 **radius-server key**。

用户可以配置设备，通过使用 AAA 服务器组，将现有的服务器主机汇总起来用于身份认证。更多详细信息，用户可以参考下面的相关主题。

用户还需要在 RADIUS 服务器上配置一些设置。这些设置包括设备的 IP 地址，以及服务器和设备共享的密钥字符串。更多详细信息，用户可以参考 RADIUS 服务器文档。

用户可以按照以下步骤配置基于服务器的 RADIUS 服务器通信。

在开始前

如果用户在设备上配置了全局功能，并针对每台服务器服务器配置了功能（超时、重传和密钥命令），那么针对每台服务器设置的定时器、重传和密钥值命令，会覆盖全局配置的定时器、重传和密钥值命令。在所有 RADIUS 服务器上配置这些设置的信息，用户可以参考下面的相关主题。

总步骤

1. enable

2. configure terminal

3. radius server *server name*

4. address {**ipv4 | ipv6**}*ip address*{ **auth-port** *port number* | **acct-port** *port number*}

5. key *string*

6. retransmit *value*

7. timeout *seconds*

8. end

9. show running-config

10. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式 • 在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**radius server** *server name* **示例：** `Device(config)# radius server rsim`	
步骤 4	**address** {**ipv4	ipv6**}*ip address*{ **auth-port** *port number*	**acct-port** *port number*} **示例：** `Device(config-radius-server)# address ipv4 124.2.2.12 auth-port 1612`	（可选）指定 RADIUS 服务器参数。 在 **auth-port** *port-number* 部分为认证请求指定 UDP 目的端口。默认值是 1645，取值范围是 0 至 65535 在 **acct-port** *port-number* 部分指定认证请求的 UDP 目的端口。默认值是 1646
步骤 5	**key** *string*	（可选）在 **key** *string* 部分指定设备和

	示例： `Device(config-radius-server)#` **`key rad123`**	运行了 RADIUS 守护程序的 RADIUS 服务器之间所使用的认证和加密密钥。 **注释：** 密钥的文本字符串必须与 RADIUS 服务器上使用的加密密钥相匹配。用户始终要把密钥配置为 **radius server** 命令中的最后一项。字符串前面的空格会被忽略，但密钥结尾处可以设置空格。如果在密钥中使用了空格，用户不要把密钥括在引号中，除非引号是密钥的一部分
步骤 6	**retransmit** *value* **示例：** `Device(config-radius-server)#` **`retransmit 10`**	（可选）指定当服务器没有响应或响应较慢时，重新发送 RADIUS 请求的次数。取值范围是 1 至 100。这个设置会覆盖全局配置命令 **radius-server retransmit** 中的设置
步骤 7	**timeout** *seconds* **示例：** `Device(config-radius-server)#` **`timeout 60`**	（可选）指定设备等待 RADIUS 服务器的响应，等待指定的时间间隔后重新发送请求。取值范围是 1 至 1000。这个设置会覆盖全局配置命令 **radius-server timeout** 中的设置
步骤 8	**end** **示例：** `Device(config)#` **`end`**	返回特权 EXEC 模式
步骤 9	**show running-config** **示例：** `Device#` **`show running-config`**	检查用户输入的信息
步骤 10	**copy running-config startup-config** **示例：** `Device#` **`copy running-config`**	（可选）把输入的命令保存到配置文件中

	startup-config	

配置 RADIUS 登录认证

用户可以按照以下步骤来配置 RADIUS 登录认证：

在开始前

要想使用 AAA 方法来保护设备上的 HTTP 访问，用户必须使用全局配置命令 **ip http authentication aaa** 来配置设备。配置 AAA 认证并不会使用 AAA 方法来保护设备上的 HTTP 访问。

总步骤

1. enable

2. configure terminal

3. aaa new-model

4. aaa authentication login {**default** | *list-name*} *method1* [*method2...*]

5. line [**console** | **tty** | **vty**] *line-number* [*ending-line-number*]

6. login authentication {**default** | *list-name*}

7. end

8. show running-config

9. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**aaa new-model** **示例：**	启用 AAA

	Device(config)# **aaa new-model**	
步骤 4	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] **示例：** Device(config)# **aaa authentication login default local**	创建一个登录认证方法列表。 • 要想创建一个默认列表，当用户*没有*在 **login authentication** 命令中指定命名列表时就使用这个默认列表，用户需要在默认情况下使用的方法后面添加 **default** 关键字。默认方法列表会自动被应用到所有端口 • 在 *list-name* 部分指定一个字符串，用来命名用户创建的列表 • 在 *method1…*部分指定认证算法使用的实际方法。其他认证方法只有当前一个方法返回了错误响应消息时才会使用，而不是返回失败消息时使用。 用户可以选择以下方法之一： • *enable*——使用 enable 密码来进行认证。在用户可以使用这个认证方法前，必须使用全局配置命令 **enable** *password* 来定义一个 enable 密码 • *group radius*——使用 RADIUS 认证。在用户可以使用这个认证方法前，必须配置 RADIUS 服务器 • *line*——使用线路密码来进行认证。在用户可以使用这个认证方法前，必须先定义一个线路密码。用户可以使用线路配置命令 **password** *password*

		• *local*——使用本地用户名数据库进行认证。用户必须输入数据库中的用户名信息。需要使用全局配置命令 **username** *name* **password** 进行配置 • *local-case*——使用区分大小写的本地用户名数据库进行认证。用户必须使用全局配置命令 **username** *password*，把用户名信息输入到数据库中 • *none*——不为登录使用任何认证
步骤 5	**line** [**console** | **tty** | **vty**] *line-number* [*ending-line-number*] **示例：** `Device(config)# line 1 4`	进入线路配置模式，并对想要应用认证列表的线路进行配置
步骤 6	**login authentication** {**default** | *list-name*} **示例：** `Device(config-line)# login authentication default`	把认证列表应用在一条或多条线路上。 • 如果用户指定了 **default**，就使用命令 **aaa authentication login** 创建默认列表 • 在 *list-name* 部分指定 **aaa authentication login** 命令中创建的列表
步骤 7	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 8	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息

步骤 9	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

定义 AAA 服务器组

用户可以使用 **server** 服务器组配置命令，把指定服务器关联到用户定义的服务器组中。用户可以使用服务器的 IP 地址来标识服务器，或使用可选关键字 **auth-port** 和 **acct-port** 来标识多个主机实例或条目。

用户可以按照以下步骤来定义 AAA 服务器组：

总步骤

1. enable

2. configure terminal

3. radius server *name*

4. address {**ipv4** | **ipv6**} {*ip-address* | *hostname*} **auth-port** *port-number* **acct-port** *port-number*

5. key *string*

6. end

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**radius server** *name*	为受保护访问证书（PAC）的部署指

	示例： `Device(config)# radius server ISE`	定 RADIUS 服务器的名称，并进入 RADIUS 服务器配置模式。 设备也为 IPv6 支持 RADIUS
步骤 4	**address** {**ipv4** | **ipv6**} {*ip-address* | *hostname*} **auth-port** *port-number* **acct-port** *port-number* **示例：** `Device(config-radius-server)# address ipv4 10.1.1.1 auth-port 1645 acct-port 1646`	为 RADIUS 服务器审计和认证参数配置 IPv4 地址
步骤 5	**key** *string* **示例：** `Device(config-radius-server)# key inspur123`	指定设备和 RADIUS 服务器之间所使用的认证和加密密钥。
步骤 6	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 7	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 8	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

为用户特权访问和网络服务配置 RADIUS 授权

注释： 对于通过 CLI 登录且已通过了认证的用户，即使配置了授权，也会绕过授权。

用户可以按照以下步骤为特权访问和网络服务配置 RADIUS 授权：

总步骤

1. enable

2. configure terminal

3. aaa authorization network radius

4. aaa authorization exec radius

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**aaa authorization network tacacs+** **示例：** `Device(config)# aaa authorization network radius`	配置交换机为所有与网络相关的服务请求使用 RADIUS 授权
步骤 4	**aaa authorization exec tacacs+** **示例：** `Device(config)# aaa authorization exec radius`	配置交换机为特权 EXEC 的访问使用 RADIUS 授权。 **exec** 关键字可能会返回用户配置文件信息（比如 **autocommand** 信息）

步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

接下来做什么？

用户可以使用全局配置命令 **aaa authorization** 和 **radius** 关键字设置指定参数，来限制用户访问特权 EXEC 模式的网络访问行为。

aaa authorization exec radius local 命令中可以设置以下三个授权参数：

- 如果使用 RADIUS 执行认证的话，用户可以使用 RADIUS 来提供特权 EXEC 访问的授权；
- 如果没有使用 RADIUS 执行认证的话，用户可以使用本地数据库来进行授权。

开始使用 RADIUS 审计

用户可以按照以下步骤开始使用 RADIUS 审计。

总步骤

1. enable

2. configure terminal

3. aaa accounting network start-stop radius

4. aaa accounting exec start-stop radius

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**aaa accounting network start-stop radius** **示例：** `Device(config)# aaa accounting network start-stop radius`	为所有与网络相关的服务请求启用 RADIUS 审计
步骤 4	**aaa accounting exec start-stop radius** **示例：** `Device(config)# aaa accounting exec start-stop radius`	启用 RADIUS 审计，在开始特权 EXEC 处理时发送开始记录（start-record）审计通知，在结束时发送停止记录（stop-record）
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

	示例： `Device# copy running-config startup-config`	

接下来做什么？

如果在 AAA 服务器不可达时，要与路由器建立会话，用户可以使用 **aaa accounting system guarantee-first** 命令。这条命令可以确保系统审计为第一条记录，这也是默认的条件。在有些情况下，这种设置可能会阻止用户在 Console 或终端连接上启动会话，直到系统重启才能解决问题，这可能需要 3 分钟以上的时间。

如果路由器重启时 AAA 服务器不可达，要想与路由器建立 Console 或 Telnet 会话，用户可以使用 **no aaa accounting system guarantee-first** 命令。

为所有 RADIUS 服务器配置相关设置

从特权 EXEC 模式开始，用户可以按照以下步骤为所有 RADIUS 服务器配置相关设置：

总步骤

1. configure terminal

2. radius-server key *string*

3. radius-server retransmit *retries*

4. radius-server timeout *seconds*

5. radius-server deadtime *minutes*

6. end

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**radius-server key** *string* **示例：**	指定交换机和所有 RADIUS 服务器之间共享的秘密文本字符串。 **注释：** 密钥的文本字符串必须与

	Device(config)# **radius-server key your_server_key** Device(config)# **key your_server_key**	RADIUS 服务器上使用的加密密钥相匹配。字符串前面的空格会被忽略，但密钥结尾处可以设置空格。如果在密钥中使用了空格，用户不要把密钥括在引号中，除非引号是密钥的一部分
步骤 3	**radius-server retransmit** *retries* **示例：** Device(config)# **radius-server retransmit 5**	指定交换机发送 RADIUS 请求的次数，超出指定次数后交换机会放弃发送。默认值是 2；取值范围是 1 至 1000
步骤 4	**radius-server timeout** *seconds* **示例：** Device(config)# **radius-server timeout 3**	指定交换机在重新发送请求之前，等待 RADIUS 请求响应的时间。默认值是 5 秒钟；取值范围是 1 至 1000
步骤 5	**radius-server deadtime** *minutes* **示例：** Device(config)# **radius-server deadtime 0**	当 RADIUS 服务器么有响应认证请求时，用户使用这条命令来指定服务器停止发送请求的时间。这样做可以避免在尝试下一个配置的服务器之前，等待请求就超时了。默认值是 0；取值范围是 1 至 1440 分钟
步骤 6	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 7	**show running-config** **示例：** Device# **show running-config**	检查用户输入的信息
步骤 8	**copy running-config startup-config** **示例：**	（可选）把输入的命令保存到配置文件中

	Device# **copy running-config startup-config**	

配置设备来使用厂商指定的 RADIUS 属性

用户可以按照以下步骤，配置设备来使用厂商指定的 RADIUS 属性：

总步骤

1. enable

2. configure terminal

3. radius-server vsa send [**accounting** | **authentication**]

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** Device> **enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式
步骤 3	**radius-server vsa send** [**accounting**	**authentication**] **示例：** Device(config)# **radius-server vsa send accounting**	启用设备来识别并使用 RADIUS IETF 属性 26 中定义的 VSA。 • （可选）使用 **accounting** 关键字把一部分厂商指定的属性设置为只发送审计属性 • （可选）使用 **authentication** 关键字把一部分厂商指定的属性设置为只发送认证属性 如果用户在输入这条命令时没有设置

		关键字的话，会同时使用审计和认证厂商指定的属性
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置设备来使用厂商私有的 RADIUS 服务器通信

用户可以按照以下步骤，来配置设备使用厂商私有的 RADIUS 服务器通信。

总步骤

1. enable

2. configure terminal

3. radius server *server name*

4. address { **ipv4 | ipv6** }*ip address*

5. non-standard

6. key *string*

7. end

8. show running-config

9. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入

	示例： `Device> enable`	密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**radius server** *server name* **示例：** `Device(config)# radius server rsim`	指定 RADIUS 服务器
步骤 4	**address** { **ipv4 | ipv6** }*ip address* **示例：** `Device(config-radius-server)# address ipv4 172.24.25.10`	（可选）指定 RADIUS 服务器的 IP 地址
步骤 5	**non-standard** **示例：** `Device(config-radius-server)# non-standard`	使用厂商私有的 RADIUS 部署方式来标识 RADIUS 服务器
步骤 6	**key** *string* **示例：** `Device(config-radius-server)# key rad123`	指定设备和厂商私有 RADIUS 服务器之间所使用的共享秘密文本字符串。设备和 RADIUS 服务器会使用这个文本字符串来加密密码和交换响应信息
步骤 7	**end** **示例：** `Device(config-radius-server)# end`	返回特权 EXEC 模式

步骤 8	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 9	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

在设备上配置 CoA

用户可以按照以下步骤在设备上配置 CoA，用户需要按序配置。

总步骤

1. enable

2. configure terminal

3. aaa new-model

4. aaa server radius dynamic-author

5. client {*ip-address* | *name*} [**vrf** *vrfname*] [**server-key** *string*]

6. server-key [**0** | **7**] *string*

7. port *port-number*

8. auth-type {**any** | **all** | **session-key**}

9. ignore session-key

10. ignore server-key

11. authentication command bounce-port ignore

12. authentication command disable-port ignore

13. end

14. show running-config

15. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密

	示例： `Device> enable`	码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**aaa new-model** **示例：** `Device(config)# aaa new-model`	启用 AAA
步骤 4	**aaa server radius dynamic-author** **示例：** `Device(config)# aaa server radius dynamic-author`	把设备配置为认证、授权和审计（AAA）服务器，用来与外部策略服务器协同工作
步骤 5	**client** {*ip-address* | *name*} [**vrf** *vrfname*] [**server-key** *string*]	进入动态授权本地服务器配置模式，并指定 RADIUS 客户端，设备会从这个 RADIUS 客户端接受 CoA 请求和断开连接请求
步骤 6	**server-key** [**0** | **7**] *string* **示例：** `Device(config-sg-radius)# server-key your_server_key`	配置设备和 RADIUS 客户端之间共享的 RADIUS 密钥
步骤 7	**port** *port-number* **示例：** `Device(config-sg-radius)# port 25`	在设备上指定端口，让设备通过这个端口侦听 RADIUS 客户端发来的 RADIUS 请求
步骤 8	**auth-type** {**any** | **all** | **session-key**}	指定设备为 RADIUS 客户端使用的授权

	示例： Device(config-sg-radius)# **auth-type any**	类型。 客户端必须匹配用户配置的所有属性，才能获得授权
步骤 9	**ignore session-key**	（可选）配置设备来忽略会话密钥。 有关 **ignore** 命令的更多信息，用户可以参考 icntnetworks.com 上的 *Inspur INOS Intelligent Services Gateway Command Reference*
步骤 10	**ignore server-key** **示例：** Device(config-sg-radius)# **ignore server-key**	（可选）配置设备来忽略服务器密钥。 有关 **ignore** 命令的更多信息，用户可以参考 icntnetworks.com 上的 *Inspur INOS Intelligent Services Gateway Command Reference*
步骤 11	**authentication command bounce-port ignore** **示例：** Device(config-sg-radius)# **authentication command bounce-port ignore**	（可选）配置设备来忽略 CoA 请求，以便暂时禁止在端口上发起会话。暂时禁用端口的目的是为了当 VLAN 发生变化且终端设备无法检测这个变化时，触发 DHCP 重配置
步骤 12	**authentication command disable-port ignore** **示例：** Device(config-sg-radius)# **authentication command disable-port ignore**	（可选）配置设备来忽略非标准命令：该命令请求发起会话的端口变为管理失效模式。关闭端口的结果是会话终结。 用户需要使用标准 CLI 或 SNMP 命令来重新启用端口
步骤 13	**end** **示例：** Device(config-sg-radius)#	返回特权 EXEC 模式

	end	
步骤 14	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 15	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

监控 CoA 功能

表 132：特权 EXEC show 命令

命令	目的
show aaa attributes protocol radius	显示 RADIUS 命令的 AAA 属性

表 133：全局排错命令

命令	目的
debug radius	显示有关 RADIUS 排错的信息
debug aaa coa	显示有关 CoA 进程排错的信息
debug aaa pod	显示有关 POD 数据包排错的信息
debug aaa subsys	显示有关 POD 数据包排错的信息
debug cmdhd [**detail** | **error** | **events**]	显示排错命令的头部信息

有关这些命令显示信息中各个字段的详细信息，用户可以参考这个版本的命令参考文档。

其他参考资料

相关文档

相关主题	文档名称
为会话感知类网络配置身份控制策略和身份	Session Aware Networking Configuration

服务模版	Guide, Inspur INOS (Inspur 6850 Switches) http://www.cisco.com/en/US/docs/ios-xml/ios/san/configuration/xe-3se/3850/san-svc-temp.html
配置 RADIUS、TACACS+、SSH、802.1X 和 AAA	Securing User Services Configuration Guide Library, Inspur INOS (Inspur 6850 Switches) http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/security/config_library/xe-3se/3850/secuser-xe-3se-3850-library.html

错误消息解码器

描述	链接
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

标准和 RFC

标准/RFC	标题
无	-

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密	http://www.icntnetworks.com

码。	

配置 Kerberos

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置 Kerberos 控制交换机访问的先决条件

配置 Kerberos 来控制交换机访问的先决条件如下所示：

- 为了使远程用户能够向网络服务进行身份验证，用户必须在 Kerberos 域中配置主机和 KDC，以便使用户和网络服务能够进行通信和相互认证。为此，用户必须让它们能够识别出彼此。用户需要把主机的条目添加到 KDC 上的 Kerberos 数据库中，并将由 KDC 生成的 KEYTAB 文件添加到 Kerberos 域中的所有主机上。用户还可以在 KDC 数据库中为用户创建条目；
- Kerberos 服务器可以是一台交换机，用户需要把它配置为网络安全服务器，这样它就可以使用 Kerberos 协议来对用户进行认证了。

用户在为主机和用户添加或创建条目时，需要遵从以下指导：

- Kerberos 规则名称*必须*全都是小写字符；
- Kerberos 实例名称*必须*全都是小写字符；
- Kerberos 域名*必须*全都是大写字符。

Kerberos 的相关信息

这一部分提供了 Kerberos 的相关信息。

Kerberos 和交换机访问

这一部分描述了如何启用和配置 Kerberos 安全系统，它能够使用一个受信任的第三方，来为网络资源进行请求认证。

注释： 在 Kerberos 配置示例中个，受信任的第三方可以是任何支持 Kerberos 的交换机，这太交换机会被用户配置为网络安全服务器，并且它能够使用 Kerberos 协议来对用户进行认证。

Kerberos 概述

Kerberos 是一项秘密密钥网络认证协议，由麻省理工学院（MIT）开发。它使用数据加密标准（DES）加密算法进行加密和认证，并为网络资源的请求提供认证。Kerberos 使用受信任第三方的概念来对用户和服务提供安全认证。这个受信任的第三方称为*密钥分发中心*（KDC）。Kerberos 会验证用户是他们所声称的用户，以及他们所使用的网络服务是他们所声称要使用的服务。为此，KDC 或受信任的 Kerberos 服务器会向用户发送票据（ticket）。这些票据具有有限的生命周期，并储存在用户证书缓存中。Kerberos 服务器会使用这些票据，而不是使用用户名和密码来验证用户和网络服务。

注释： Kerberos 服务器可以是任何被用户配置为网络安全服务器的交换机，并且它能够使用 Kerberos 协议来对用户进行认证。

Kerberos 的证书机制使用称为*单一登录*（Single Logon）的进程。这个进程会一次性对用户进行认证，然后根据接受的用户证书范围，在这个范围中的任何地方用户都会通过安全验证（无需加密另一个密码）。

这个软件版本支持 Kerberos 5，已经在使用 Kerberos 5 的组织机构可以在 KDC 上，使用已在其他网络主机（比如 UNIX 服务器和 PC）上使用的相同 Kerberos 身份认证数据库。

Kerberos 支持以下网络服务：

- Telnet
- rlogin
- rsh

下面这个表格中列出了常见的 Kerberos 相关术语和定义。

表 134：Kerberos 术语

术语	**定义**
认证	用户或服务向另一个服务验证自己身份的过程。举例来说，客户端可以向交换机验证自己的身份，或者交换机可以向另一台交换机验证自己的身份
授权	交换机用来识别用户拥有网络中哪些特权的方式，或者交换机用来识别用户能够执行哪些行为的方式
证书	表示认证票据的通用术语，比如 TGT[9] 和服务证书。Kerberos 的证书验证了用户或服务器的身份。如果一个网络服务决定信任发放票据的 Kerberos 服务器，它就可以使用证书来代替用户名和密码。证书的默认生命周期是 8 个小时
实例	Kerberos 规则的授权等级标签。大多数 Kerberos 规则的格式都是 *user@REALM*（比如 smith@EXAMPLE.COM ）。Kerberos 规则和 Kerberos 实例一起表示为 *user/instance@REALM*（比如 smith/admin@EXAMPLE.COM）。Kerberos 示例可以用来指定授权等级，如果用户认证成功了的话。提供每个网络服务的服务器可能会实施并使用 Kerberos 示例的授权映射，但这并不是强制行为。 **注释：** Kerberos 规则和实例的名称*必须*全都是小写字符

	注释： Kerberos 域名*必须*全都是大写字符
KDC[10]	密钥分发中心，由 Kerberos 服务器和运行在一台网络主机上的数据库程序构成
Kerberos 化的	这个术语用来描述已经被变更为能够支持 Kerberos 证书架构的应用和服务
Kerberos 域	由用户、主机和网络服务构成的域，这些组成元素都会注册到 Kerberos 服务器上。Kerberos 服务器是受信任的设备，用来向用户或网络服务认证另一个用户或网络服务的身份。 **注释：** Kerberos 域名*必须*全都是大写字符
Kerberos 服务器	运行在网络主机上的一个守护进程。用户和网络服务把它们各自的身份注册到 Kerberos 服务器上。网络服务向 Kerberos 服务器进行查询，以此来认证其他网络服务
KEYTAB[11]	网络服务与 KDC 共享的密码。在 Kerberos 5 和后续的 Kerberos 版本中，网络服务在认证一个加密的服务证书时，会使用 KEYTAB 对其进行解密。在 Kerberos 5 之前的版本中，KEYTAB 称为 SRVTAB[12]
规则	也称为 Kerberos 实体，这是根据 Kerberos 服务器指定的设备身份或服务器身份 **注释：** Kerberos 规则名称*必须*全都是小写字符
服务证书	网络服务的证书。KDC 颁发后，这个证书是使用网络服务和 KDC 之间共享的密码进行加密的。这个密码也会与用户 TGT 共享
SRVTAB	网络服务与 KDC 共享的密码。在 Kerberos 5 或后续的 Kerberos 版本中，SRVTAB 也称为 KEYTAB
TGT	承认的票据，这是 KDC 颁发给通过认证的用户的证书。用户接收到 TGT 时，它们可以在

	KDC 代表的 Kerberos 域中对网络服务进行认证

9 承认的票据

10 密钥分发中心

11 密钥表

12 服务器表

Kerberos 的工作原理

Kerberos 服务器可以是配置为网络安全服务器的任何设备，它可以使用 Kerberos 协议来认证远端用户。尽管用户可以使用多种方式来自定义 Kerberos，但远端用户在尝试访问网络服务时，必须通过三层安全防范措施，才能访问网络服务。

要想使用 Kerberos 服务器设备来对网络服务进行认证，远端用户必须遵从以下步骤：

向边界交换机进行认证

这部分描述了远端用户必须通过的第一层安全防范措施。用户必须首先向边界交换机进行认证。这时会发生以下事件：

1. 用户向边界交换机开启未 Kerberos 化的 Telnet 连接；
2. 交换机向用户提示输入用户名和密码；
3. 交换机为用户向 KDC 请求 TGT；
4. KDC 向交换机发送加密的 TGT，其中包含用户的身份；
5. 交换机尝试使用用户输入的密码来解密 TGT：
   - 如果解密成功的话，交换机上的用户认证就成功了；
   - 如果解密没有成功的话，用户会重复步骤 2：重新输入用户名和密码（如果大写键或数字键已开启或关闭的话），或输入另一个用户名和密码。

初始化非 Kerberos 化的 Telnet 会话并向边界交换机进行身份认证的远程用户位于防火墙内，但在访问网络服务之前，用户仍必须直接向 KDC 进行身份验证。用户必须向 KDC 进行身份验证，因为交换机上储存着 KDC 颁发的 TGT，并且这个 TGT 无法在用户登录到交换机前，用于其他认证。

从 KDC 获得 TGT

这一部分介绍了远程用户必须通过的第 2 层安全防范措施。用户现在必须向 KDC 进行认证，并从 KDC 获得 TGT 来访问网络服务。

如何向 KDC 进行认证的指导，用户可以参考 *Inspur INOS Security Configuration Guide, Release 12.4* 中，“Security Server Protocols”一章中的“Obtaining a TGT from a KDC”部分。

向网络服务进行认证

这部分介绍了远程用户必须通过的第 3 层安全防范措施。持有 TGT 的用户现在必须向 Kerberos 域中的网络服务进行身份验证。

如何向网络服务进行认证的指导，用户可以参考 *Inspur INOS Security Configuration Guide, Release 12.4* 中，“Security Server Protocols”一章中的“Authenticating to Network Services”部分。

如何配置 Kerberos

要想建立一个由 Kerberos 进行认证的服务器-客户端系统，用户需要按照以下步骤进行配置：

- 使用 Kerberos 命令来配置 KDC；
- 配置交换机来使用 Kerberos 协议。

监控 Kerberos 的配置

要想查看 Kerberos 的配置，用户可以使用以下命令：

- **show running-config**
- **show kerberos creds**：列出当前用户证书缓存中的证书
- **clear kerberos creds**：清除当前用户证书缓存中的所有证书，其中包括转发的证书

其他参考资料

相关文档

相关主题	文档名称
Kerberos 命令	*Inspur INOS Security Command Reference*

错误消息解码器

描述	链接
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	http://www.icntnetworks.com

配置本地认证和授权

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

如何配置本地认证和授权

配置交换机来执行本地认证和授权

用户可以通过设置交换机实施本地模式的 AAA，以此实现不使用服务器的 AAA 操作。交换机会负责处理认证和授权事宜。这种配置不支持审计。

注释： 为了使用 AAA 方式来确保 HTTP 访问交换机的安全性，用户必须使用全局配置命令 **ip http authentication aaa** 来配置交换机。配置 AAA 认证并不会对使用 AAA 方法的交换机提供 HTTP 访问保护。

用户可以按照以下步骤来配置 AAA 操作，以本地的方式（而不是用服务器）来设置交换机实施 AAA：

总步骤

1. enable

2. configure terminal

3. aaa new-model

4. aaa authentication login default local

5. aaa authorization exec local

6. aaa authorization network local

7. username *name* [**privilege** *level*] {**password** *encryption-type password*}

8. end

9. show running-config

10. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**aaa new-model** **示例：** `Device(config)# aaa new-model`	启用 AAA
步骤 4	**aaa authentication login default local** **示例：** `Device(config)# aaa authentication login default local`	设置使用本地用户名数据库来执行登录认证。使用 **default** 关键字为所有端口使用本地用户数据库进行认证
步骤 5	**aaa authorization exec local** **示例：** `Device(config)# aaa authorization exec local`	配置用户 AAA 授权、检查本地数据库，并允许用户使用 EXEC 命令
步骤 6	**aaa authorization network local** **示例：**	为所有与网络相关的服务请求配置 AAA 授权

	Device(config)# **aaa authorization network local**	
步骤 7	**username** *name* [**privilege** *level*] {**password***encryption-type password*} **示例：** Device(config)# **username your_user_name privilege 1 password 7 secret567**	进入本地数据库，并建立基于用户名的认证系统。 用户需要为每个用户重复配置以下命令： • 在 *name* 部分指定用户 ID，可以指定一个单词。不能使用空格和引号 • （可选）在 *level* 部分指定用户能够获得的特权等级。取值范围是 0 至 15。等级 15 是特权 EXEC 模式的访问权限。等级 0 是用户 EXEC 模式的访问权限 • 在 *encryption-type* 部分输入 0 来指定未加密的密码。输入 7 来指定隐藏密码 • 在 *password* 部分指定用户必须输入并获得交换机访问权限的密码。密码必须在 1 至 25 字符之间，可以包含空格，并且必须是 **username** 命令中配置的最后一个选项
步骤 8	**end** **示例：** Device(config-sg-radius)# **end**	返回特权 EXEC 模式
步骤 9	**show running-config** **示例：** Device# **show running-config**	检查用户输入的信息
步骤 10	**copy running-config startup-config**	（可选）把输入的命令保存到配置文件

	示例： `Device# copy running-config startup-config`	中

监控本地认证和授权

要想查看本地认证和授权的配置，用户可以使用特权 EXEC 命令 **show running-config**。

其他参考资料

错误消息解码器

描述	链接
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	http://www.icntnetworks.com

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置安全壳的先决条件

在交换机上配置安全壳（SSH）拥有以下先决条件：

- 要想让 SSH 正常工作，交换机上需要有 Rivest、Shamir 和 Adleman（RSA）公钥/私钥对。这与安全复制协议（SCP）相同，SCP 依赖于 SSH 来提供传输安全性；
- 在启用 SCP 之前，用户必须先在交换机上正确配置了 SSH、认证和授权；
- 因为 SCP 依赖于 SSH 来提供传输安全性，因此路由器上必须拥有 Rivest、Shamir 和 Adleman（RSA）公钥/私钥对；
- SCP 依赖于 SSH 来提供安全性；
- SCP 需要认证、授权和审计（AAA）的授权配置，这样路由器才能够确定用户应该获得的正确特权等级；
- 用户必须有正确的授权，才能使用 SCP；
- 拥有能够使用 SCP 正确授权的用户，可以使用 **copy** 命令把交换机 Inspur INOS 文件系统（IFS）中的文件复制进来或复制出去。授权的管理员也可以从工作站执行相同的工作；
- 安全壳（SSH）服务器需要使用 IPsec（数据加密标准[DES]或 3DES）加密软件镜像；SSH 客户端需要使用 IPsec（DES 或 3DES）加密软件镜像；
- 用户可以使用全局配置模式的命令 **hostname** 和 **ip domain-name** 来为设备配置用户名和主机域名。

配置安全壳的限制条件

在设备上配置安全壳拥有以下限制条件：

- 交换机需要支持 Rivest、Shamir 和 Adleman（RSA）认证；
- SSH 只支持可执行 Shell 应用；
- 只有数据加密标准（DES，56 比特）和 3DES（168 比特）数据加密软件能够支持 SSH 服务器和 SSH 客户端。在 DES 软件镜像中，DES 是唯一可用的加密算法。在 3DES 软件镜像中，可以使用 DES 和 3DES 加密算法；
- 设备支持高级加密算法（AES）加密算法：128 比特密钥、192 比特密钥或 256 比特密钥。但不支持使用对称密码 AES 来加密密钥；
- 这个软件版本不支持 IP 安全（IPsec）；
- 在使用 SCP 时，用户不能在 **copy** 命令中输入密码。用户必须在看到提示时输入密码；
- 安全壳版本 1 中不支持登录旗标消息。安全壳版本 2 中可以支持；
- 在配置 Console 访问的反向 SSH 备用方法时，-l 分隔符关键字和用户 ID：{number} {ip-address}参数是必需的。

SSH 的相关信息

安全壳（SSH）是一个用来为设备提供安全远程连接的协议。SSH 比 Telnet 为远程连接提供了更多的安全性，它能够在设备认证时提供强健的加密措施。这个软件版本支持 SSH 版本 1（SSHv1）和 SSH 版本 2（SSHv2）。

SSH 和交换机访问

安全壳（SSH）是一个用来为设备提供安全远程连接的协议。SSH 比 Telnet 为远程连接提供了更多的安全性，它能够在设备认证时提供强健的加密措施。这个软件版本支持 SSH 版本 1（SSHv1）和 SSH 版本 2（SSHv2）。

SSH 能够为 IPv6 提供与 IPv4 相同的功能。对于 IPv6 来说，SSH 能够支持 IPv6 地址，并且能够为使用 IPv6 传输的远端 IPv6 节点，提供启用了安全加密的连接。

SSH 服务器、集成客户端和支持的版本

安全壳（SSH）集成客户端特性是运行在 SSH 协议上的应用，用来提供设备认证和加密。SSH 客户端能够使 Inspur 设备与其他 Inspur 设备之间建立安全加密的连接，或者与其他运行 SSH 服务器的设备之间建立安全加密的连接。这条连接提供的功能与带外 Telnet 连接提供的功能类似，只不过这条连接是加密的。通过使用认证和加密，SSH 客户端可以在不安全的网络上提供安全通信。

SSH 服务器和 SSH 集成客户端都是运行在交换机上的应用。SSH 服务器与这个版本中支持的 SSH 客户端和非 Inspur SSH 客户端一起工作。SSH 客户端能够与可用的公开和商业 SSH 服务器一起工作。SSH 客户端能够支持数据加密标准（DES）、3DES 和密码认证。

交换机上支持 SSHv1 或 SSHv2 服务器。

交换机上支持 SSHv1 客户端。

注释： 只有当用户启用了 SSH 服务器后，才能使用 SSH 客户端功能。

为用户提供的用户认证功能与 Telnet 会话中提供的用户认证类似。SSH 也支持下列用户认证方式：

- TACACS+
- RADIUS
- 本地认证和授权

SSH 的配置指导

在把交换机配置为 SSH 服务器或 SSH 客户端时，用户需要遵从以下指导：

- SSHv1 服务器生成的 RSA 密钥对，也可以由 SSHv2 服务器使用，反之亦然；
- 如果在堆栈主用设备上运行 SSH 服务器，并且堆栈主用设备失效了，新的堆栈主用设备会使用前一个堆栈主用设备所生成的 RSA 密钥对；
- 如果用户在输入全局配置命令 **crypto key generate rsa** 后看到了 CLI 错误消息，并且 RSA 密钥对没有生成。那么用户需要重新配置用户名和域名，然后再次输入 **crypto key generate rsa** 命令。更多信息，用户可以参考相关主题部分；
- 在生成 RAS 密钥对时，交换机可能会显示出没有指定主机名的消息。如果看到了这条消息，用户必须使用全局配置命令 **hostname** 来配置主机名；
- 生成 RAS 密钥对时，交换机可能会显示出没有指定域名的消息。如果看到了这条消息，用户必须使用全局配置命令 **ip comain-name** 来配置 IP 域名；

- 在配置本地认证和授权的认证方法时，用户要确保在 Console 端口上禁用了 AAA。

安全复制协议概述

安全复制协议（SCP）特性为复制交换机配置文件或交换机镜像文件提供了一种安全且能够执行认证的方法。SCP 依赖于安全壳（SSH），SSH 是能够为 Berkeley r-tools 提供安全性的应用和协议。

要想让 SSH 正常工作，交换机上需要有 RSA 公钥/私钥对。这与 SCP 相同，SCP 依赖于 SSH 来提供传输安全性。

由于 SSH 也依赖于 AAA 认证，因此 SCP 也依赖于 AAA 授权，因此用户需要正确配置相关信息。

- 在启用 SCP 之前，用户必须先在交换机上正确配置了 SSH、认证和授权；
- 因为 SCP 依赖于 SSH 来提供传输安全性，因此路由器上必须拥有 Rivest、Shamir 和 Adleman（RSA）公钥/私钥对

注释： 在使用 SCP 时，用户不能在 **copy** 命令中输入密码。用户必须在看到提示时输入密码。

安全复制协议

安全复制协议（SCP）特性为复制设备配置文件或交换机镜像文件提供了一种安全，且能够进行认证的方法。SCP 的行为与远程复制（rcp）类似，后者来自于 Berkeley 远程工具集，只不过 ACP 依赖于 SSH 提供安全性。SCP 也需要配置认证、授权和审计（AAA）的授权功能，这样设备才能够确定用户应该使用的正确特权等级。要想配置安全复制特性，用户应该理解 SCP 的概念。

如何配置 SSH

设置设备来运行 SSH

用户可以按照以下步骤，设置设备来运行 SSH：

在开始前

为本地或远程访问配置用户认证功能。用户需要按顺序进行配置。更多信息用户可以参考相关主题部分。

总步骤

1. enable

2. configure terminal

3. hostname *hostname*

4. ip domain-name *domain_name*

5. crypto key generate rsa

6. end

7. show running-config

8. copy running-config startup-config

具体配置

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**hostname** *hostname* **示例：** `Device(config)# hostname your_hostname`	为用户的设备配置主机名和 IP 域名。 **注释：** 只有在把设备配置为 SSH 服务器时，才使用这个步骤
步骤 4	**ip domain-name** *domain_name* **示例：** `Device(config)# ip domain-name your_domain`	为用户的设备配置一个域名
步骤 5	**crypto key generate rsa**	在设备上为本地和远程认证启用 SSH

	示例： `Device(config)# crypto key generate rsa`	服务器功能，并生成一个 RSA 密钥对。为设备生成 RSA 密钥对会自动启用 SSH。 我们建议使用的最小系数大小是 1024 比特。 在生成 RSA 密钥时，用户会看到输入系数长度的提示。更长的系数会提供更高的安全性，但也会需要更长的时间来进行生成和使用。 **注释：** 只有在把设备配置为 SSH 服务器时，才使用这个步骤
步骤 6	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 7	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 8	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 SSH 服务器

用户可以按照以下步骤来配置 SSH 服务器：

注释： 只有在把设备配置为 SSH 服务器时，才使用这个步骤。

总步骤

1. enable

2. configure terminal

3. ip ssh version [**1** | **2**]

4. ip ssh {**timeout** *seconds* | **authentication-retries** *number*}

5. 使用以下命令之一：

• **line vty***line_number*[*ending_line_number*]

• **transport input ssh**

6. end

7. show running-config

8. copy running-config startup-config

具体配置

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip ssh version** [**1**	**2**] **示例：** `Device(config)# ip ssh version 1`	（可选）配置设备来运行 SSH 版本 1 或 SSH 版本 2。 • **1**——配置设备来运行 SSH 版本 1 • **2**——配置设备来运行 SSH 版本 2 如果用户没有输入这条命令，或者没有指定关键字，SSH 服务器会选择 SSH 客户端所支持的最新 SSH 版本。举例来说，如果 SSH 客户端支持 SSHv1 和 SSHv2，那么 SSH 服务器会选择使用 SSHv2
步骤 4	**ip ssh** {**timeout** *seconds* | **authentication-retries** *number*} **示例：** `Device(config)# ip ssh`	配置 SSH 控制参数： • 以秒为单位指定超时值；默认值为 120 秒钟。取值范围是 0 至 120 秒。这个参数应用在 SSH 协商阶段。在连接建立后，设备会使用基于 CLI

	timeout 90 **authentication-retries 2**	会话的默认超时值 默认情况下，设备为网络中多条 CLI 会话同时支持 5 条加密的 SSH 连接（会话 0 至会话 4）。在开始执行 Shell 后，基于 CLI 的会话超时值会返回到默认的 10 分钟 • 指定客户端可以向服务器进行重认证的次数。默认值是 3；取值范围是 0 至 5 用户需要重复这个步骤来同时配置两个参数

步骤 5	使用以下命令之一：
	• **line vty** *line_number* [*ending_line_number*]
	• **transport input ssh**
	示例：
	`Device(config)# line vty 1 10`
	或者
	`Device(config-line)# transport input ssh`
		（可选）配置虚拟终端线路设置：
		• 进入线路配置模式，来配置虚拟终端线路设置。在 *line_number* 和 *ending_line_number* 部分指定一对线路，取值范围是 0 至 5
		• 指定设备阻止非 SSH 的 Telnet 连接。限制路由器只支持 SSH 连接

步骤 6	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 7	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 8	**copy running-config startup-config** **示例：**	（可选）把输入的命令保存到配置文件中

	Device# **copy running-config startup-config**	

监控 SSH 的配置和状态

下面这个表格中的命令显示了 SSH 服务器的配置和状态。

表 135：显示 SSH 服务器配置和状态的命令

命令	目的
show ip ssh	显示 SSH 服务器的版本和配置信息
show ssh	显示 SSH 服务器的状态

其他参考资料

相关主题

相关主题	文档名称
为会话感知类网络配置身份控制策略和身份服务模版	Session Aware Networking Configuration Guide, Inspur INOS (Inspur 6850 Switches) Http://www.icntnetworks.com
配置 RADIUS、TACACS+、SSH、802.1X 和 AAA	Securing User Services Configuration Guide Library, Inspur INOS (Inspur 6850 Switches) http://www.icntnetworks.com

错误消息解码器

描述	链接
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

标准和 RFC

标准/RFC	标题
无	-

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	http://www.icntnetworks.com

SSH 的特性信息

版本	特性信息
Inspur INOS 11.3.1	引入该特性

用于 SSH 认证的 X.509v3 证书

用于 SSH 认证的 X.509v3 证书

SSH 认证的 X.509v3 证书特性在安全壳（SSH）服务器侧的服务器和用户认证使用 X.509v3 数字证书。

这部分描述了如何为数字证书配置服务器和用户证书配置文件。

查询特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置用于 SSH 认证的 X.509v3 证书的先决条件

用于 SSH 认证的 X.509v3 证书特性引入了 **ip ssh server algorithm authentication** 命令，来代替 **ip ssh server authenticate user** 命令。如果用户使用了 **ip ssh server authenticate user** 命令，设备上会显示以下信息。

Warning: SSH command accepted but this CLI will be deprecated soon. Please move to new CLI “ip ssh server algorithm authentication”. Please configure “default ip ssh server authenticate user” to make CLI ineffective.

- 用户可以使用命令 **default ip ssh server authenticate user** 来移除 **ip ssh server authenticate user** 命令的作用。然后使用 **ip ssh server algorithm authentication** 命令来启

用 INOS 安全壳（SSH）。

配置用于 SSH 认证的 X.509v3 证书的限制条件

- 用于 SSH 认证的 X.509v3 证书特性只能实施在 INOS 安全壳（SSH）服务器侧；
- INOS SSH 服务器在 INOS SSH 服务器侧的服务器和用户认证上，只支持基于 x509v3-ssh-rsa 算法的认证。

用于 SSH 认证的 X.509v3 证书的相关信息

数字证书

认证的有效性取决于公共签名密钥和签名者身份之间的联系强度。X.509v3 格式（RFC5280）的数字证书用来提供身份管理。受信任的根证书机构及其中间证书机构的签名链，会把指定的公共签名密钥绑定到指定的数字身份。

公钥基础设施（PKI）信任点有助于管理数字证书。证书和信任点之间的关联有助于跟踪证书。信任点包含有关证书颁发机构（CA）、不同的身份参数和数字证书的信息。用户可以创建多个信任点来与不同的证书相关联。

使用 X.509v3 的服务器和用户认证

对于服务器认证来说，INOS 安全壳（SSH）服务器会把自己的证书发送到 SSH 客户端进行验证。这个服务器证书与服务器证书配置文件（配置在 ssh-server-cert-profile-server 配置模式中）中配置的信任点相关联。

对于用户认证来说，SSH 客户端会把用户的证书发送到 INOS SSH 服务器进行验证。SSH 服务器会使用服务器证书配置文件（配置在 ssh-server-cert-profile-user 配置模式中）中配置的公钥基础设施（PKI）信任点来验证入站的用户证书。

默认情况下，用户需要在 INOS SSH 服务器端为服务器和用户启用基于证书的身份认证。

如何配置用于 SSH 认证的 X.509v3 证书

配置 INOS SSH 服务器来为服务器认证使用数字证书

总步骤

1. enable

2. configure terminal

3. ip ssh server algorithm hostkey {**x509v3-ssh-rsa** [**ssh-rsa**] | **ssh-rsa** [**x509v3-ssh-rsa**]}

4. ip ssh server certificate profile

5. server

6. trustpoint sign *PKI-trustpoint-name*

7. ocsp-response include

8. end

具体步骤

	命令或操作	目的
步 骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步 骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步 骤 3	**ip ssh server algorithm hostkey** {**x509v3-ssh-rsa** [**ssh-rsa**]	**ssh-rsa** [**x509v3-ssh-rsa**]} **示例：** `Device(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa`	定义主机密钥算法的顺序。只有配置的算法会用于与安全壳（SSH）客户端的协商中。 **注释：** INOS SSH 服务器上必须至少配置一个主机密钥算法： • ssh-rsa——基于公钥

		的算法 • x509v3-ssh-rsa——基于证书的认证
步骤 4	**ip ssh server certificate profile** **示例：** `Device(config)# ip ssh server certificate profile`	配置服务器证书配置文件和用户证书配置文件，并进入 SSH 证书配置文件配置模式
步骤 5	**server** **示例：** `Device(ssh-server-cert-profile)# server`	配置服务器证书配置文件，并进入 SSH 服务器证书配置文件服务器配置模式
步骤 6	**trustpoint sign** *PKI-trustpoint-name* **示例：** `Device(ssh-server-cert-profile-server)# trustpoint sign trust1`	把公钥基础设施（PKI）信任点关联到服务器证书配置文件。SSH 服务器会使用这个 PKI 信任点关联的证书来对服务器进行认证
步骤 7	**ocsp-response include** **示例：** `Device(ssh-server-cert-profile-server)# ocsp-response include`	（可选）随服务器证书发送在线证书状态协议（OCSP）响应或 OCSP 闭合（Stapling）。 **注释：** 默认情况下配置的是这条命令的“no”形式，也就是不会随服务器证书发送 OCSP 响应
步骤 8	**end** **示例：** `Device(ssh-server-cert-profile-server)# end`	离开 SSH 服务器证书配置文件服务器配置模式，并进入特权 EXEC 模式

配置 INOS SSH 服务器为用户认证验证用户数字证书

总步骤

1. enable

2. configure terminal

3. ip ssh server algorithm authentication {**publickey** | **keyboard** | **password**}

4. ip ssh server algorithm publickey {**x509v3-ssh-rsa** [**ssh-rsa**] | **ssh-rsa** [**x509v3-ssh-rsa**]}

5. ip ssh server certificate profile

6. user

7. trustpoint verify *PKI-trustpoint-name*

8. ocsp-response required

9. end

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip ssh server algorithm authentication** {**publickey** | **keyboard** | **password**} **示例：** `Device(config)# ip ssh server algorithm authentication publickey`	定义用户认证算法的顺序。只有配置的算法会用于与安全壳（SSH）客户端的协商中。 **注释：** INOS SSH 服务器上必须至少配置一个用户认证算法 **注释：** 要想为用户认证使用证书方式，用户必须配置 **publickey** 关键字 **注释：** 使用 **ip ssh server algorithm authentication** 命令

		代替 **ip ssh server authenticate user** 命令
步骤 4	**ip ssh server algorithm publickey** {**x509v3-ssh-rsa** [**ssh-rsa**] | **ssh-rsa** [**x509v3-ssh-rsa**]} **示例：** `Device(config)# ip ssh server algorithm publickey x509v3-ssh-rsa`	定义公钥算法的顺序。只有配置的算法可以被 SSH 客户端接受并用于用户认证。 **注释：** INOS SSH 客户端上必须至少配置一个公钥算法： • ssh-rsa——基于公钥的算法 • x509v3-ssh-rsa——基于证书的认证
步骤 5	**ip ssh server certificate profile** **示例：** `Device(config)# ip ssh server certificate profile`	配置服务器证书配置文件和用户证书配置文件，并进入 SSH 证书配置文件配置模式中
步骤 6	**user** **示例：** `Device(ssh-server-cert-profile)# user`	配置用户证书配置文件，并进入 SSH 服务器证书配置文件用户配置模式
步骤 7	**trustpoint verify** *PKI-trustpoint-name* **示例：** `Device(ssh-server-cert-profile-user)# trustpoint verify trust2`	配置公钥基础设施（PKI）信任点，以此用来验证入站的用户证书。 **注释：** 用户需要多次执行相同的命令来配置多个信任点。用户最多可以配置 10 个信任点
步骤 8	**ocsp-response required** **示例：** `Device(ssh-server-cert-profile-user)# ocsp-response required`	（可选）使用在线证书状态协议（OCSP）来对入站用户的证书做出响应。 **注释：** 默认的配置是这条命令的“no”格式，无需 OCSP

		响应就会接受用户证书
步骤 9	**end** **示例：** `Device(ssh-server-cert-profile-user)# end`	离开 SSH 服务器证书配置文件用户配置模式，并进入特权 EXEC 模式

验证使用数字证书的服务器和用户认证配置

总步骤

1. enable

2. show ip ssh

具体步骤

步骤 1 enable

进入特权 EXEC 模式

- 在提示时输入密码

示例：

```
Device> enable
```

步骤 2 show ip ssh

显示当前配置的认证方法。来确认使用的基于证书的认证，并确保配置的主机密钥算法是 x509v3-ssh-rsa 算法。

示例：

```
Device# show ip ssh
SSH Enabled - version 1.99
Authentication methods:publickey,keyboard-interactive,password
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa
Hostkey Algorithms:x509v3-ssh-rsa,ssh-rsa
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
```

用于 SSH 认证的 X.509v3 证书配置示例

示例：配置 INOS SSH 服务器来为服务器认证使用数字证书

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa
Device(config)# ip ssh server certificate profile
Device(ssh-server-cert-profile)# server
Device(ssh-server-cert-profile-server)# trustpoint sign trust1
Device(ssh-server-cert-profile-server)# exit
```

示例：配置 INOS SSH 服务器来为用户认证验证用户的数字证书

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm authentication publickey
Device(config)# ip ssh server algorithm publickey x509v3-ssh-rsa
Device(config)# ip ssh server certificate profile
Device(ssh-server-cert-profile)# user
Device(ssh-server-cert-profile-user)# trustpoint verify trust2
Device(ssh-server-cert-profile-user)# end
```

用于 SSH 认证的 X.509v3 证书的其他参考资料

相关文档

相关主题	文档名称
Inspur INOS 命令	Inspur INOS 主命令列表，所有版本
安全命令	• Inspur INOS 安全命令参考：命令 A 至 C • Inspur INOS 安全命令参考：命令 D 至 L • Inspur INOS 安全命令参考：命令 M 至 R • Inspur INOS 安全命令参考：命令 S 至 Z
SSH 认证	*SecureShellConfigurationGuide*中的“Secure Shell-Configuring User Authentication Methods”一章
公钥基础设施（PKI）信任点	*Public Key Infrastructure ConfigurationGuide*中的“Configuring and Managing a Inspur INOS Certificate Server for PKI Deployment”一章

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	http://www.icntnetworks.com

用于 SSH 认证的 X.509v3 证书的特性信息

下面这个表格提供了这部分内容中描述的特性版本信息。这个表格中只列出了指定软件版本系列中，引入该特性的软件版本。除非另行说明，否则这个软件版本的后续版本也支持该特性。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

表 136：用于 SSH 认证的 X.509v3 证书的特性信息

特性名称	版本	特性信息
用于 SSH 认证的 X.509v3 证书	Inspur INOS XE 3.14S 版本	用于 SSH 认证的 X.509v3 证书特性在安全壳（SSH）服务器侧，在服务器和用户认证中使用 X.509v3 数字证书。 这个版本中引入或更改了以下命令：**ip ssh server algorithm hostkey**、**ip ssh**

		server algorithm authentication 和 ip ssh server certificate profile

配置安全套接字层 HTTP

查寻特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

安全套接字层（SSL）HTTP 的相关信息

安全 HTTP 服务器和客户端概述

在安全的 HTTP 连接上，去往和来自 HTTP 服务器的数据在通过 Internet 进行传输之前，会先进行加密。使用 SSL 加密的 HTTP 能够提供安全连接，从而实现一些功能，比如从 Web 浏览器对交换机进行配置。Inspur 安全 HTTP 服务器和安全 HTTP 客户端的实现利用了提供应用层加密的 SSL 3.0 版本。HTTP over SSL 缩写为 HTTPS；提供安全连接的 URL 以 https://而不是 http://开头。

注释： 1999 年 SSL 演变为传输层安全（TLS）协议，但有些环境中仍会使用 SSL。

HTTP 安全服务器（交换机）的主要作用是侦听指定端口（默认的 HTTPS 端口是 443）上的 HTTPS 请求，并将请求传递给 HTTP 1.1 Web 服务器。HTTP 1.1 服务器会负责处理这个请求，并把响应（页面）传递回到 HTTP 安全服务器，接着 HTTP 安全服务器会对原始请求做出响应。

HTTP 安全客户端（Web 浏览器）的主要作用是为 HTTPS 用户代理服务，做出 INOS 应用请求响应、为应用执行 HTTPS 用户代理服务，并将响应传递回给应用。

注释： 从 Inspur INOS 11.3.1 版本开始，用户能够为 HTTP 服务器关联 IPv6 ACL。在 Inspur INOS 11.3.1 版本之前，用户只能为安全 HTTP 服务器配置 IPv4 ACL。用户可以为安全 HTTP 服务器使用配置 CLI 命令，把已经配置好的 IPv6 和 IPv4 ACL 关联到 HTTP 服务器。

证书授权中心信任点

证书授权中心（CA）负责管理证书请求，并向参与网络行为的设备颁发证书。这些服务为参与网络行为的设备提供了安全密钥和证书的集中管理。指定的 CA 服务器称为信任点。

当用户在尝试连接时，HTTPS 服务器会通过向客户端颁发 X.509v3 证书的方式来为客户端提供安全连接，这个证书是从指定的 CA 信任点获取的，且已经经过了认证。接着客户端（通常是 Web 浏览器）会使用公钥来对这个证书进行认证。

为了确保 HTTP 连接的安全性，我们强烈建议用户配置一个 CA 信任点。如果用户没有为运行 HTTPS 服务器的设备配置 CA 信任点，则服务器会对自己进行认证，并生成所需的 RSA 密钥对。由于自我认证（自签名）证书无法提供足够的安全性，因此服务器所连接的客户端上会生成一个通知消息，表明这个证书是自我认证的，并让用户有机会接受或拒绝这个连接。这个选项在内部网络拓扑（比如测试）中很有用。

如果用户没有配置 CA 信任点的话，则在启用安全 HTTP 连接时，设备会自动生成为安全 HTTP 服务器（或客户端）使用的临时或永久自签名证书。

- 如果交换机上没有配置主机名和域名，则它会生成一个临时的自签名证书。在交换机重新启动后，所有临时的自签名证书都会丢失，并且交换机会生成一个新的临时自签名证书；
- 如果交换机上已配置了主机和域名，则它会生成一个永久自签名证书。在交换机重新启动后，或者禁用了安全 HTTP 服务器，以便下次重新启用安全 HTTP 连接时，这个证书会保持可用状态。

注释： 用户必须在每个设备上单独配置证书授权中心和信任点信息。从其他设备上复制的信息是无效的。

在注册新证书时，新的配置变更不会立即应用于 HTTPS 服务器，直到服务器重新启动为止。用户可以使用 CLI 或者通过物理的方式，重新启动服务器。在重新启动服务器时，交换机将会使用新证书。

如果交换机生成了自签名证书，特权 EXEC 命令 **show running-config** 的输出内容中会包含以下信息。以下为命令输出中的部分内容，只显示了自签名证书的命令。

```
Device# show running-config
Building configuration...
<output truncated>
crypto pki trustpoint TP-self-signed-3080755072
enrollment selfsigned
subject-name cn=INOS-Self-Signed-Certificate-
3080755072 revocation-check none
rsakeypair TP-self-signed-3080755072
!
!
crypto ca certificate chain TP-self-signed-3080755072
certificate self-signed 01
3082029F  30820208  A0030201  02020101  300D0609  2A864886  F70D0101
04050030
59312F30  2D060355  04031326  494F532D  53656C66  2D536967  6E65642D
43657274
69666963  6174652D  33303830  37353530  37323126  30240609  2A864886
F70D0109
02161743  45322D33  3535302D  31332E73  756D6D30  342D3335  3530301E
170D3933
30333031  30303030  35395A17  0D323030  31303130  30303030  305A3059
312F302D
<output truncated>
```

要想删除这个自签名证书，用户可以禁用安全 HTTP 服务器，并且输入全局配置命令 **no crypto pki trustpoint TP-self-signed-30890755072**。如果用户之后再次启用安全 HTTP 服务器，交换机就会生成一个新的自签名证书。

注释： *TP-self-signed* 后面的数值取决于设备的序列号。

用户可以使用可选命令（**ip http secure-client-auth**）使 HTTPS 服务器能够从客户端请求

X.509v3 证书。认证客户端要比服务器的自身认证提供了更高的安全性。

更多有关认证授权中心的信息，用户可以查看*Inspur INOS Security Configuration Guide, Release 12.4*中的“Configuring Certification Authority Interoperability”一章。

加密套件

加密套件（CipherSuite）指定了在一个 SSL 连接上使用的加密算法和摘要算法。在与 HTTPS 服务器建立连接时，客户端 Web 浏览器会提供它所支持的加密套件列表，客户端和服务器会在它们都支持的列表中协商出最佳的加密算法来使用。举例来说，Netscape Communicator 4.76 能够支持使用 RSA 公钥加密的 U.S.安全性、MD2、MD5、RC2-CBC、RC4、DES-CBC 和 DES-EDE3-CBC。

为了实现尽可能好的加密措施，用户应该使用支持 128 比特加密算法的客户端浏览器，比如 Microsoft Internet Explorer V5.5（或更高版本）或 Netscape Communicator 4.76 版本（或更高版本）。SSL_RSA_WITH_DES_CBC_SHA 加密套件提供的安全性低于其他的加密套件，因为它不提供 128 比特加密。

使用更安全和更复杂的加密套件需要消耗更多的处理时间。这个列表定义了交换机所支持的加密套件，并按照路由器处理负载（速度），把它们按照从最快到最慢的顺序进行排列：

1. SSL_RSA_WITH_DES_CBC_SHA——在 RSA 密钥交换（RSA 公钥加密）中，为消息加密使用 DES-CBC 加密，为消息摘要使用 SHA；
2. SSL_RSA_WITH_NULL_SHA——在密钥交换中，为消息加密使用 NULL，为消息摘要使用 SHA（只用于 SSL 3.0）；
3. SSL_RSA_WITH_NULL_MD5——在密钥交换中，为消息加密使用 NULL，为消息摘要使用 MD5（只用于 SSL 3.0）；
4. SSL_RSA_WITH_RC4_128_MD5——在 RSA 密钥交换中，为消息加密使用 RC4 128 比特加密，为消息摘要使用 MD5；
5. SSL_RSA_WITH_RC4_128_SHA——在 RSA 密钥交换中，为消息加密使用 RC4 128 比特加密，为消息摘要使用 SHA；
6. SSL_RSA_WITH_3DES_EDE_CBC_SHA——在 RSA 密钥交换中，为消息加密使用 3DES 和 DES-EDE3-CBC 加密，为消息摘要使用 SHA；
7. SSL_RSA_WITH_AES_128_CBC_SHA——在 RSA 密钥交换中，为消息加密使用 AES 128 比特加密，为消息摘要使用 SHA（只适用于 SSL 3.0）；
8. SSL_RSA_WITH_AES_256_CBC_SHA——在 RSA 密钥交换中，为消息加密使用 AES 256 比特加密，为消息摘要使用 SHA（只适用于 SSL 3.0）；

9. SSL_RSA_WITH_DHE_AES_128_CBC_SHA——在 RSA 密钥交换中，为消息加密使用 AES 128 比特加密，为消息摘要使用 SHA（只适用于 SSL 3.0）；
10. SSL_RSA_WITH_DHE_AES_256_CBC_SHA——在 RSA 密钥交换中，为消息加密使用 AES 256 比特加密，为消息摘要使用 SHA（只适用于 SSL 3.0）

注释： 最新版本的 Chrome 浏览器不支持四个原始的加密套件，因此无法访问 Web GUI 和用户门户。

RSA（与指定的加密和摘要算法组合相结合）同时用于 SSL 连接上的密钥生成和认证。这种用法与用户是否配置了 CA 信任点无关。

默认的 SSL 配置

启用了标准 HTTP 服务器

启用了 SSL

未配置 CA 信任点

未生成自签名证书

SSL 的配置指导

当用户在交换机集群中使用 SSL 时，SSL 会话会终结在集群指挥官（Commander）上。集群成员交换机上必须运行标准 HTTP。

在用户配置 CA 信任点之前，应该确保已经设置了系统时钟。如果用户没有设置时钟的话，则交换机会因为日期不正确而拒绝证书。

在交换机堆栈中，SSL 会话会终结在堆栈主用设备上。

如何配置安全 HTTP 服务器和客户端

配置 CA 信任点

为了保障 HTTP 连接的安全性，我们建议用户配置一个官方的 CA 信任点。CA 信任点要比自签名证书更安全。

从特权 EXEC 模式开始，用户可以按照以下步骤来配置 CA 信任点：

总步骤

1. configure terminal

2. hostname *hostname*

3. ip domain-name *domain-name*

4. crypto key generate rsa

5. crypto ca trustpoint *name*

6. enrollment url *url*

7. enrollment http-proxy *host-name port-number*

8. crl query *url*

9. primary *name*

10. exit

11. crypto ca authentication *name*

12. crypto ca enroll *name*

13. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device# configure terminal`	进入全局配置模式
步骤 2	**hostname** *hostname* 示例： `Device(config)# hostname your_hostname`	指定交换机的主机名（只有当用户之前没有配置过主机名时才需要配置）。使用安全密钥和证书需要有主机名
步骤 3	**ip domain-name** *domain-name* 示例： `Device(config)# ip domain-name your_domain`	指定交换机的 IP 域名（只有当用户之前没有配置过 IP 域名时才需要配置）。使用安全密钥和证书需要有域名
步骤 4	**crypto key generate rsa**	（可选）生成一个 RSA 密钥对。交换机上先要拥有 RSA 密钥对，用户才能为交

	示例： `Device(config)# crypto key generate rsa`	换机获取一个证书。RSA 密钥对是自动生成的。用户可以使用这条命令来按需生成密钥
步骤 5	**crypto ca trustpoint** *name* 示例： `Device(config)# crypto ca trustpoint your_trustpoint`	为 CA 信任点指定一个本地配置名称，并进入 CA 信任点配置模式
步骤 6	**enrollment url** *url* 示例： `Device(ca-trustpoint)# enrollment url http://your_server:80`	指定一个 URL，也就是交换机向其发送证书请求的 URL
步骤 7	**enrollment http-proxy** *host-name port-number* 示例： `Device(ca-trustpoint)# enrollment http-proxy your_host 49`	（可选）配置交换机通过 HTTP 代理服务器，从 CA 那里获得证书。 • 在 *host-name* 部分指定用来访问 CA 的代理服务器 • 在 *port-number* 部分指定用来访问 CA 的端口号
步骤 8	**crl query** *url* 示例： `Device(ca-trustpoint)# crl query ldap://your_host:49`	配置交换机来请求一个证书撤销列表（CRL），来确保对等体的证书未被撤销
步骤 9	**primary** *name* 示例： `Device(ca-trustpoint)# primary your_trustpoint`	（可选）指定一个信任点，并将这个信任点用作处理 CA 请求的主用（默认）信任点。 • 在 *name* 部分指定用户刚配置的信任点
步骤 10	**exit**	退出 CA 信任点配置模式，并返回全局

	示例： `Device(ca-trustpoint)# exit`	配置模式
步骤 11	**crypto ca authentication** *name* **示例：** `Device(config)# crypto ca authentication your_trustpoint`	通过获得 CA 的公钥来对 CA 进行认证。使用与步骤 5 相同的名称
步骤 12	**crypto ca enroll** *name* **示例：** `Device(config)# crypto ca enroll your_trustpoint`	从指定的 CA 信任点获取证书。这条命令会为每个 RSA 密钥对请求一个签名证书
步骤 13	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式

配置安全 HTTP 服务器

从特权 EXEC 模式开始，用户可以按照以下步骤来配置安全 HTTP 服务器。

在开始前

如果用户使用证书授权中心来提供认证服务，则应该在启用 HTTP 服务器之前，按照上述步骤在交换机上配置 CA 信任点。如果用户没有配置 CA 信任点，则在首次启用安全 HTTP 服务器时，交换机会生成自签名证书。在用户配置了服务器之后，用户可以有选择地应用适用于标准和安全 HTTP 服务器的选项（路径、要应用的访问列表、最大连接数量，或超时策略）。用户要想验证使用 Web 浏览器的安全 HTTP 连接，可以输入 https://*URL*，其中 URL 是服务器交换机的 IP 地址或主机名。如果用户配置了默认端口之外的端口，就还必须在 URL 后面指定端口号。举例来说：

注释： 不支持 AES256_SHA2。

`https://209.165.129:1026`

或者

```
https://host.domain.com:1026
```

用来指定访问列表（只适用于 IPv4 ACL）的现有命令 **ip http access-class access-list-number** 将被弃用。用户仍然可以使用这条命令来指定访问列表，来放行访问 HTTP 服务器的流量。现在用户可以使用两个新命令来指定 IPv4 和 IPv6 ACL。

命令 **ip http access-class ipv4** *access-list-name* | *access-list-number* 用来指定 IPv4 ACL，命令 **ip http access-class ipv6** *access-list-name* 用来指定 IPv6 ACL。我们建议用户使用新的 CLI 命令，避免收到警告消息。

在指定访问列表时，用户需要考虑以下考量因素：

- 如果用户在指定一个不存在的访问列表，配置会生效，同时用户会收到以下警告消息：

  ```
  ACL being attached does not exist, please configure it
  ```

- 如果用户使用命令 **ip http access-class** 为 HTTP 服务器指定访问列表，用户会看到以下警告消息：

  ```
  This CLI will be deprecated soon, Please use new CLI ip http
  access-class ipv4/ipv6 <access-list-name>| <access-list-number>
  ```

- 如果用户使用命令 **ip http access-class ipv4** *access-list-name* | *access-list-number* 或命令 **ip http access-class ipv6** *access-list-name* 时，已经使用命令 **ip http access-class** 配置了访问列表，用户就会看到以下警告消息：

  ```
  Removing ip http access-class <access-list-number>
  ```

命令 **ip http access-class** *access-list-number* 和命令 **ip http access-class ipv4** *access-list-name* | *access-list-number* 拥有相同的功能。每条命令会覆盖之前命令的配置。这两条命令的下列组合会对运行配置带来以下影响：

- 如果用户已经配置命令 **ip http access-class** *access-list-number*，之后再尝试配置 **ip http access-class ipv4** *access-list-number* 命令，那么命令 **ip http access-class** *access-list-number* 的配置会被移除，命令 **ip http access-class ipv4** *access-list-number* 的配置会被放入运行配置中；
- 如果用户已经配置了命令 **ip http access-class** *access-list-number*，之后再尝试配置 **ip http access-class ipv4** *access-list-name* 命令，那么命令 **ip http access-class** *access-list-number* 的配置会被移除，命令 **ip http access-class ipv4** *access-list-name* 的配置会被添加到运行配置中；
- 如果用户已经配置了命令 **ip http access-class ipv4** *access-list-number*，之后再尝试配置 **ip http access-class** *access-list-name* 命令，那么命令 **ip http access-class ipv4** *access-list-number* 的配置会被移除，命令 **ip http access-class** *access-list-name* 的配置会被添加到运行配置中；

- 如果用户已经配置了命令 **ip http access-class ipv4** *access-list-name*，之后再尝试配置 **ip http access-class** *access-list-number* 命令，那么命令 **ip http access-class ipv4** *access-list-name* 的配置会被移除，命令 **ip http access-class** *access-list-number* 的配置会被添加到运行配置中。

总步骤

1. show ip http server status

2. configure terminal

3. ip http secure-server

4. ip http secure-port *port-number*

5. ip http secure-ciphersuite {[**3des-ede-cbc-sha**] [**rc4-128-md5**] [**rc4-128-sha**] [**des-cbc-sha**]}

6. ip http secure-client-auth

7. ip http secure-trustpoint *name*

8. ip http path *path-name*

9. ip http access-class { **ipv4** {*access-list-number* | *access-list-name*} | **ipv6** {*access-list-name*} }

10. ip http max-connections *value*

11. ip http timeout-policy idle *seconds* **life** *seconds* **requests** *value*

12. end

具体配置

	命令或操作	目的

步骤 1	**show ip http server status**
	
	示例：
	`Device# show ip http server status`
		（可选）显示 HTTP 服务器的状态，来确定软件中是否支持安全 HTTP 服务器特性。用户应该会在输出信息中看到以下输出内容：
		`HTTP secure server capability: Present`
		或者
		`HTTP secure server capability: Not present`

步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式

步骤 3	**ip http secure-server** **示例：** `Device(config)# ip http secure-server`	启用 HTTP 服务器（如果还未启用的话）。HTTPS 服务器默认是启用的
步骤 4	**ip http secure-port** *port-number* **示例：** `Device(config)# ip http secure-port 443`	（可选）指定 HTTPS 服务器使用的端口号。默认端口号是 443。有效选项是 443 或 1025 至 65535 之中的任意号码
步骤 5	**ip http secure-ciphersuite** {[**3des-ede-cbc-sha**] [**rc4-128-md5**] [**rc4-128-sha**] [**des-cbc-sha**]} **示例：** `Device(config)# ip http secure-ciphersuite rc4-128-md5`	（可选）指定在 HTTPS 连接上，为加密使用的加密套件（加密算法）。如果用户不想指定具体的加密套件，就应该允许服务器和客户端来协商出它们都支持的加密套件。这是默认设置
步骤 6	**ip http secure-client-auth** **示例：** `Device(config)# ip http secure-client-auth`	（可选）配置 HTTP 服务器在连接过程中，向客户端请求 X.509v3 证书用于认证。默认设置是客户端需要向服务器请求证书，但服务器无需向客户端请求证书用于认证
步骤 7	**ip http secure-trustpoint** *name* **示例：** `Device(config)# ip http secure-trustpoint your_trustpoint`	指定用来获得 X.509v3 安全证书的 CA 信任点，并用来认证客户端认证连接。 **注释：** 使用这条命令的前提是用户已经按照之前的步骤配置了 CA 信任点
步骤 8	**ip http path** *path-name* **示例：**	（可选）为 HTML 文件设置一个基本 HTTP 路径。路径定义了本地系统上的 HTTP 服务器文件的位置（通常位于系

	Device(config)# **ip http path /your_server:80**	统 flash 内存中）
步骤 9	**ip http access-class** { **ipv4** {*access-list-number* | *access-list-name*} | **ipv6** {*access-list-name*}} **示例：** Device(config)# **ip http access-class ipv4 4**	（可选）指定用来允许 HTTP 服务器访问的访问列表
步骤 10	**ip http max-connections** *value* **示例：** Device(config)# **ip http max-connections 4**	（可选）设置同时连接 HTTP 服务器的最大并发连接数量。我们建议设置不小于 10 的值。这可以保证 UI 功能正常运行
步骤 11	**ip http timeout-policy idle** *seconds* **life** *seconds* **requests** *value* **示例：** Device(config)# **ip http timeout-policy idle 120 life 240 requests 1**	（可选）指定 HTTP 服务器连接可以维持多长时间，并且定义以下情况： • **idle**——指定最大空闲时间周期，也就是没有收到数据，或者不能发送响应数据的时间段。取值范围是 1 至 600 秒。默认值为 180 秒钟（3 分钟） • **life**——指定连接建立后的最大时间周期。取值范围是 1 至 86400 秒（24 小时）。默认值为 180 秒钟 • **requests**——在持续连接上处理请求的最大数量。最大值为 86400，默认值为 1
步骤 12	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式

配置安全 HTTP 客户端

从特权 EXEC 模式开始，用户可以按照以下步骤来配置安全 HTTP 客户端：

在开始前

标准 HTTP 客户端和安全 HTTP 客户端总是启用的。证书授权中心需要为安全 HTTP 客户端提供证书。在配置以下命令的前提是用户已经在交换机上配置了 CA 信任点。如果用户没有配置 CA 信任点，并且远端 HTTPS 服务器请求客户端进行认证，那么与安全 HTTP 客户端之间的连接会失效。

总步骤

1. configure terminal

2. ip http client secure-trustpoint *name*

3. ip http client secure-ciphersuite {[**3des-ede-cbc-sha**] [**rc4-128-md5**] [**rc4-128-sha**] [**des-cbc-sha**]}

4. end

具体配置

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**ip http client secure-trustpoint** *name* **示例：** `Device(config)# ip http client secure-trustpoint your_trustpoint`	（可选）指定远端 HTTP 服务器在请求客户端认证时使用的 CA 信任点。使用这条命令的前提是用户已经按照之前的步骤配置了 CA 信任点。当无需进行客户端认证，或者当用户已经配置了主用信任点时，这条命令是可选的
步骤 3	**ip http client secure-ciphersuite** {[**3des-ede-cbc-sha**] [**rc4-128-md5**] [**rc4-128-sha**] [**des-cbc-sha**]} **示例：** `Device(config)# ip http`	（可选）指定在 HTTPS 连接上，为加密使用的加密套件（加密算法）。如果用户不想指定具体的加密套件，就应该允许服务器和客户端来协商出它们都支持的加密套件。这是默认设置

	client secure-ciphersuite rc4-128-md5	
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式

监控安全 HTTP 服务器和客户端状态

为了监控 SSL 安全服务器和客户端状态，用户可以使用以下表格中的特权 EXEC 命令。

表 137：显示 SSL 安全服务器和客户端状态的命令

命令	目的
show ip http client secure status	显示 HTTP 安全客户端的配置
show ip http server secure status	显示 HTTP 安全服务器的配置
show running-config	显示为安全 HTTP 连接生成的自签名证书

其他参考资料

相关主题

相关主题	文档名称
为会话感知类网络配置身份控制策略和身份服务模版	Session Aware Networking Configuration Guide, Inspur INOS XE Release 3SE (Inspur 6850 Switches) http://www.icntnetworks.com
配置 RADIUS、TACACS+、SSH、802.1X 和 AAA	Securing User Services Configuration Guide Library, Inspur INOS XE Release 3SE (Inspur 6850 Switches) http://www.icntnetworks.com

错误消息解码器

描述	链接
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

标准和 RFC

标准/RFC	标题
无	-

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	http://www.icntnetworks.com

配置 IPv4ACL

查询特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置 IPv4 访问控制列表的先决条件

本节列出了在使用访问控制列表（ACL）配置网络安全性时的一些先决条件。

- 在运行 LAN 基本特性集的交换机上，不支持 VLAN map 特性。

配置 IPv4 访问控制列表的限制条件

通用网络安全

使用 ACL 配置网络安全性的限制条件如下所示：

- 并不是所有适用于编号 ACL 的命令都适用于命名 ACL。在接口上进行包过滤和路由过滤时，可以使用命名 ACL。实施 VLAN map 也可以使用命名 ACL；
- 标准 ACL 与扩展 ACL 不可以使用相同的名字；
- 尽管在命令行的帮助信息中能看到 **appletalk**，但在 MAC 访问列表配置模式中，配置 **deny** 和 **permit** 语句时不能把 **appletalk** 作为匹配条件；
- 在下游客户端策略中，不支持 ACL 通配符。

网络接口上的 IPv4 ACL

在网络接口上应用 IPv4 ACL 的限制条件如下所示：

- 当用户希望对接口实施访问控制时，既可以使用命名 ACL，也可以使用编号 ACL；
- 如果用户在一个二层接口上应用了 ACL，且该接口是某个 VLAN 中的成员，那么这个二层（端口）ACL 优先于相应 VLAN 接口上应用的入向三层 ACL，也优先于这个 VLAN 上配置的 VLAN map；
- 如果用户在三层接口上应用了 ACL，但这个交换机并未启用路由功能，那么这个 ACL 仅过滤需要 CPU 处理的数据包，比如 SNMP、Telnet 或其他网页流量；
- 如果配置了 **preauth_ipv4_acl**ACL 用来过滤数据包，这个 ACL 会在身份验证后被清除；
- 用户在二层接口上应用 ACL 时，不必开启路由功能。

注释： 当三层接口上配置的访问控制列表拒绝了一个数据包时，路由器默认会发送 Internet 控制消息协议（ICMP）不可达消息。这些被访问控制列表拒绝了的数据包并不是在硬件中直接丢弃，而是交给交换机的 CPU 进行处理，因此设备会发送 ICMP 不可达消息。如果用户不希望交换机生成 ICMP 不可达消息，可以在使用 ACL 的同时实施接口配置命令 **no ipunreachables**，该命令能够禁用 ICMP 不可达消息。

二层接口上的 MAC ACL

用户在创建了一个 MAC ACL 后，就可以把它应用到二层接口上了，这种应用可以过滤进入该二层接口的非 IP 流量。当应用 MAC ACL 时，用户应该考虑如下的指导建议：

- 用户可以在一个二层接口上同时应用一个 IP 访问列表和一个 MAC 访问列表。IP 访问列表仅过滤 IP 数据包，但是 MAC 访问列表可以过滤所有非 IP 数据包；
- 一个二层接口上只能应用一个 MAC 访问列表。如果用户在一个已经配置了 MAC ACL 的二层接口上应用另一个新的 MAC 访问列表，那么这个新的 ACL 就会取代之前配置的那个 MAC ACL。

注释： 接口配置命令 **mac access-group** 仅在二层物理接口上有效。用户不能在 EtherChannel 端口上使用此命令。

IP 访问列表条目序号

- 这个特性不支持动态、自反、防火墙访问列表。

与 ACL 相关的网络安全信息

本章会介绍如何通过访问控制列表（ACL）在交换机上配置网络安全性，在一些命令和表格中，访问控制列表也被称为访问列表。

ACL 概述

实施数据包过滤有助于限制网络流量，同时还能够限制某些用户或设备使用网络的行为。ACL 能够过滤穿越路由器或交换机的流量，并允许或拒绝那些穿过指定接口或 VLAN 的数据包。ACL 会按顺序依次检查那些针对数据包设置的允许或拒绝的条件。当接口接收到数据包时，交换机会将数据包中的字段与访问列表中具体的条件进行比较，以判断是否转发此数据包。交换机会将数据包与访问列表中的条件逐一进行匹配。根据最先匹配的结果，交换机判断是否接收此数据包。因为一旦某个条件匹配成功，交换机就会停止检查，所以访问列表中条件的先后顺序是至关重要的。如果访问列表中的所有条件都没有匹配成功，那么交换机就会拒绝这个数据包。如果访问列表中没有匹配的限制条件，那么交换机就会转发这个数据包，否则就会丢弃数据包。交换机可以使用 ACL 过滤它转发的所有数据包，也包括那些在 VLAN 中的数据包。

用户可以通过在路由器或三层交换机上配置访问列表，为其网络提供基本的安全性。如果用户不配置 ACL，那么通过用户交换机的所有数据包都可以被传递到网络的各个部分。用户可以使用 ACL 来控制不同的主机访问网络不同的部分，或是使用 ACL 来决定路由器接口转发哪些类型的流量、阻塞哪些类型的流量。比如，用户可以允许转发电子邮件的流量，但阻塞 Telnet 的流量。用户可以配置 ACL 来阻塞入向流量、出向流量或同时阻塞双向的流量。

访问控制条目

一个 ACL 中包含一个按序排列的访问控制条目（ACE）列表。每条 ACE 中都会指定 *permit* 或 *deny* 行为，以及一组条件，数据包必须满足这些条件才能匹配这条 ACE。*permit* 或 *deny* 的对象取决于这个 ACL 当时所在的配置环境。

支持的 ACL 类型

交换机支持 IP ACL 和以太网（MAC）ACL：

- IP ACL 能够过滤 IPv4 流量，其中包括 TCP、用户数据报协议（UDP）、Internet 组管理协议（IGMP），以及 Internet 控制消息协议（ICMP）；
- 以太网 ACL 能够过滤非 IP 流量。

交换机还支持服务质量（QoS）分类 ACL。

支持的 ACL

交换机支持使用以下三种 ACL，来过滤流量：

- 端口 ACL 负责对进入二层接口的流量实施访问控制。用户仅可以应用一个 IP 访问列表和一个 MAC 访问列表；
- 路由器 ACL 负责对 VLAN 间的路由流量实施访问控制，用户可以把它应用在三层接口的某个方向上（入向或出向）；
- VLAN ACL 或 VLAN map 负责对所有（桥接的和路由的）数据包实施访问控制。用户可以使用 VLAN map 过滤同一个 VLAN 中不同设备之间的流量。配置 VLAN map 可以为基于三层地址的 IPv4 提供访问控制。要想对那些不受支持的协议实施访问控制，用户可以凭借 MAC 地址使用以太网 ACE。当一个 VLAN 中应用了 VLAN map 之后，所有（桥接的和路由的）数据包在进入这个 VLAN 时，都会由 VLAN map 进行检查。数据包可以通过交换端口进入这个 VLAN，也可以通过路由端口被路由到这个 VLAN 中。

ACL 优先级

当用户在一台交换机上同时配置了 VLAN map、端口 ACL 和路由器 ACL 时，入向流量的过滤优先级从高到低依次是：端口 ACL、VLAN map、路由器 ACL，出向流量的过滤优先级从高到低依次是：路由器 ACL、VLAN map、端口 ACL。

下列示例描述了一些简单的使用环境：

- 当用户在一台交换机上同时应用了入向端口 ACL 和入向 VLAN map 时，在那些应用了端口 ACL 的端口上，入站数据包会由端口 ACL 进行过滤。其他端口收到的入站数据包会由 VLAN map 进行过滤；
- 当用户在一个交换机虚拟接口（SVI）上同时应用了入向路由器 ACL 和入向端口 ACL 时，在那些应用了端口 ACL 的端口上，入站数据包会由端口 ACL 进行过滤。其他端口收到的入站路由 IP 数据包会由路由器 ACL 进行过滤。其他的数据包不会被过滤；
- 当用户在一个 SVI 接口上同时应用了出向路由器 ACL 和入向端口 ACL 时，在那些应用了端口 ACL 的端口上，入站数据包会由端口 ACL 进行过滤。出站路由 IP 数据包会由路由

器 ACL 进行过滤。其他的数据包不会被过滤；

- 当用户在一个 SVI 接口上同时应用了 VLAN map、入向路由器 ACL 和入向端口 ACL 时，在那些应用了端口 ACL 的端口上，入站数据包只会由端口 ACL 进行过滤。其他端口收到的入站路由 IP 数据包会由 VLAN map 和路由器 ACL 同时过滤。其他的数据包仅会由 VLAN map 进行过滤；
- 当用户在一个 SVI 接口上同时应用了 VLAN map、出向路由器 ACL 和入向端口 ACL 时，在那些应用了端口 ACL 的端口上，入站数据包仅会由端口 ACL 进行过滤。出站路由 IP 数据包会同时由 VLAN map 和路由器 ACL 进行过滤。其他的数据包仅会由 VLAN map 进行过滤。

端口 ACL

端口 ACL 是一种应用在交换机二层接口上的 ACL。它只能用在物理接口上，不能用于 EtherChannel 接口。用户可以将端口 ACL 分别应用在接口的出方向和接口的入方向上。端口 ACL 能够支持下列三种访问列表：

- 标准 IP 访问列表，针对源地址进行过滤；
- 扩展 IP 访问列表，针对源地址、目的地址，以及可选协议类型信息进行过滤；
- 扩展 MAC 访问列表，针对源 MAC 地址、目的 MAC 地址，以及可选协议类型信息进行过滤。

交换机会检查接口上应用的 ACL，并根据数据包与 ACL 条目匹配的情况，来决定允许或是拒绝转发这个数据包。这样，ACL 就能够对整个网络或网络的某部分，实施访问控制了。

下面这个示例中介绍了，当所有工作站都属于同一个 VLAN 的时候，端口 ACL 是如何对网络实施访问控制的。用户在图 103 中这台交换机的二层接口上，应用了入向端口 ACL，使得主机 A 能够正常地访问人力资源网络，但主机 B 却无法访问这个网络。在本示例中，用户仅将端口 ACL 应用在了二层接口的入方向上。

图 103：使用 ACL 来控制网络中的流量

Human Resouces Network	人力资源 网络
Research & Development Network	研究和发展 网络

ACL denying traffic from Host B and permitting traffic from Host A	ACL 拒绝从主机 B 发来的流量 允许从主机 A 发来的流量
Packet	数据包

当用户在一个 Trunk 端口上应用了端口 ACL 时，这个 ACL 会过滤该 Trunk 端口上所有 VLAN 的流量。

当用户在一个配置了语音 VLAN 的端口上应用了端口 ACL 时，这个 ACL 既可以过滤数据流量，也可以过滤语音 VLAN 的流量。

在端口 ACL 中，用户可以通过 IP 访问列表来过滤 IP 流量，通过 MAC 访问列表来过滤非 IP 流量。

用户可以在同一个二层接口上，同时应用一个 IP 访问列表和一个 MAC 访问列表，来同时过滤 IP 流量和非 IP 流量。

注释： 在一个二层接口上，用户只能应用一个 IP 访问列表和一个 MAC 访问列表。如果用户在一个已经配置了 IP 访问列表或 MAC 访问列表的二层接口上，应用另一个新的 IP 访问列表或 MAC 访问列表，那么这个新的 ACL 就会取代之前配置的那个 ACL。

路由器 ACL

用户可以分别在交换机虚拟接口（SVI；这是 VLAN 的三层接口）、三层物理接口，以及三层 EtherChannel 接口上应用路由器 ACL。用户可以在接口的某个方向上（入向或出向）应用路由器 ACL。在一个接口的每个方向上，用户只能应用一个路由器 ACL。

针对 IPv4 流量，交换机支持的访问列表如下所示：

- 标准 IP 访问列表，针对源地址进行匹配；
- 扩展 IP 访问列表，针对源地址、目的地址，以及可选的协议类型信息进行匹配。

与端口 ACL 一样，交换机会在查看 ACL 的同时，查看该接口上配置的特性。当数据包进入交换机接口的时候，交换机会检查该接口上配置的所有入向特性相关联的 ACL。那些被路由的数据包，在被转发至下一跳之前，交换机会检查与出接口上配置的所有出向特性相关联的 ACL。

根据数据包与 ACL 条目匹配的情况，ACL 来决定允许或是拒绝转发这个数据包，这样就能够对整个网络或网络的某部分，实施访问控制了。

VLAN Map

VLAN ACL 或 VLAN map 是一种用于控制 VLAN 内部网络流量的访问控制列表。用户可以把

VLAN map 应用在交换机或交换机堆栈中桥接的 VLAN 内部数据包上。VACL 能够严格地进行数据包的安全性过滤，并且可以将流量重定向到具体的物理接口上。VACL 是没有方向性的（入向或出向）。

所有非 IP 协议都是通过 MAC 地址和以太类型（EtherType）字段，使用 MAC VLAN map 进行访问控制的（IP 流量不是通过 MAC VLAN map 进行访问控制的）。用户只能对那些穿越交换机的数据包实施 VLAN map；用户不能对通过集线器相连的主机之间的流量，以及这台交换机直连的其他交换机的流量实施 VLAN map。

交换机会根据 VLAN map 中指定的动作，来决定允许或是拒绝转发这些数据包。

图 104 展示了一个应用示例，用户通过应用 VLAN map，不允许交换机转发来自 VLAN10 中主机 A 的特定流量。用户只能在一个 VLAN 上应用一个 VLAN map。

图 104：使用 VLAN Map 来控制流量

Host A	主机 A
Host B	主机 B
VLAN map denying specific type of traffic from Host A	VLAN map 拒绝了从主机 A 发来的指定类型流量
Packet	数据包

ACE 与分片和未分片流量

IP 数据包在穿越网络的时候可以进行分片处理。分片处理后，只有数据包的起始分片中会包含第 4 层信息，比如 TCP 或 UDP 的端口号、ICMP 类型和编码信息等。其他的所有分片中都不会包含上述信息。

有一些访问控制条目（ACE）不会检查四层信息，这样用户就可以把它们应用在数据包的所有分片上。而对于那些需要检查四层信息的 ACE，用户就不能直接把它们应用在 IP 数据包的非起始分片上了。当需要检查第 4 层信息的 ACE 遇到不包含四层信息的分片时，匹配规则会发生如下的改变：

- 那些检查分片中三层信息（包括协议类型，比如 TCP、UDP 等）的 ACEpermit 条目，对分片进行匹配时，并不检查该分片是否包含四层信息；
- 那些检查四层信息的 ACE 的 deny 条目，只会在分片中包含四层信息时，才会匹配这些分片。

ACE 与分片和未分片流量的示例

下列是配置在访问列表 102 中的命令，用户将访问列表 102 应用在三个数据包分片上：

```
Device(config)# access-list 102 permit tcp any host 10.1.1.1 eq smtp
Device(config)# access-list 102 deny tcp any host 10.1.1.2 eq telnet
Device(config)# access-list 102 permit tcp any host 10.1.1.2
Device(config)# access-list 102 deny tcp any any
```

注释： 在示例的第一个和第二个 ACE 中，目的地址后面的 *eq* 关键字表示要检查的 TCP 目的端口号。在这里，端口号分别由简单邮件传输协议（SMTP）和 Telnet 代替。

- 数据包 A 是一个 TCP 数据包，从主机 10.2.2.2 的 65000 端口发往主机 10.1.1.1 的 SMTP 端口。如果这个数据包经过了分片处理，那么第一个分片能够成功匹配上第一条 ACE（允许），因为第一个分片中包含了所有的四层信息，就像是一个完整的数据包一样。虽然其余的分片中并不包含 SMTP 端口信息，但是因为第一条 ACE 只检查这些分片的三层信息，所以这些分片也可以与第一条 ACE 成功匹配。本示例中匹配的信息是 TCP 数据包和目的地址 10.1.1.1；
- 数据包 B 是一个从主机 10.2.2.2 的 65001 端口发往主机 10.1.1.2 的 Telnet 端口的数据包。如果这个数据包经过了分片处理，那么第一个分片能够成功匹配上第二条 ACE（拒绝），因为第一个分片中包含了所有三层和四层的信息。因为其余的数据包分片中不包含四层信息，所以它们不会与第二条 ACE 相匹配，而是与第三条 ACE（允许）相匹配。

  因为第一个分片被拒绝，所有主机 10.1.1.2 无法把这些分片重新组合成数据包，所以数据包 B 被有效地拒绝了。但是，除了第一个分片之外的其余分片还是会被转发，这样不仅浪费了网络带宽资源，同时也会消耗主机 10.1.1.2 上的硬件资源来尝试重新组合这些分片；
- 数据包 C 是一个从主机 10.2.2.2 的 65001 端口发往主机 10.1.1.3 的 FTP 端口的数据包。如果这个数据包经过了分片处理，那么第一个分片能够成功匹配上第四条 ACE（拒绝）。其余所有的分片也可以成功匹配上第四条 ACE，因为这条 ACE 不会检查四层信息，而这些分片中的三层信息又表明它们的目的主机是 10.1.1.3，与之前几条 ACE 允许条目中的目的地址无法匹配。

ACL 和交换机堆栈

交换机堆栈对 ACL 的支持与对单台交换机对 ACL 的支持是一样的。ACL 的配置信息会被传播

到堆栈中的每一台交换机上。堆栈中的所有交换机，也包括主用交换机，都能够处理信息，以及管理它们的硬件功能。

主用交换机和 ACL 功能

主用交换机能够支持的 ACL 功能如下所示：

- 主用交换机可以处理 ACL 的配置，并将配置信息传播到堆栈中的每一台交换机；
- 主用交换机可以把 ACL 的配置信息发送给新加入到堆栈中的交换机；
- 如果由于某种原因（比如，硬件资源不足时），导致交换机必须通过软件来转发数据包，那么主用交换机仅在对这些数据包应用了 ACL 之后，才会转发它们；
- 主用交换机能够根据由自己处理过的 ACL 的配置信息，来配置自己的硬件功能。

堆栈成员和 ACL 功能

堆栈成员能够支持的 ACL 功能如下所示：

- 堆栈成员可以接收由主用交换机发来的 ACL 的配置信息，并根据这些信息来配置它们自己的硬件功能；
- 用户可以把一台堆栈成员配置为备用交换机，如果主用交换机发生了故障，这台备用交换机就会执行主用交换机的功能。

主用交换机故障和 ACL

主用交换机和备用交换机上都有 ACL 配置信息。当主用交换机发生故障时，备用交换机会接管主用交换机的角色。新的主用交换机会向所有堆栈成员发送 ACL 配置信息。

标准和扩展 IPv4 ACL

这一部分会介绍 IP ACL。

ACL 是一系列按顺序的，允许（permit）和拒绝（deny）条件的集合。交换机会把数据包与访问列表中的条件逐一进行匹配。根据最先匹配的结果，交换机会判断是接受还是拒绝这个数据包。因为一旦某个条件匹配成功，交换机就会停止检查，所以访问列表中匹配条件的先后顺序是至关重要的。如果访问列表中的所有匹配条件都没有匹配成功，那么交换机就会拒绝这个数据包。

交换机操作系统支持的 ACL 或 IPv4 访问列表类型如下所示：

- 标准 IP 访问列表，针对源地址进行匹配；
- 扩展 IP 访问列表，针对源地址、目的地址，以及可选的协议类型信息进行匹配。

交换机 IPv4 ACL 不支持的特性

在交换机上配置 IPv4 ACL，与在其他 Inspur 交换机和路由器上配置 IPv4 ACL 的方法是一样的。

交换机无法支持下列与 ACL 相关的特性：

- 非 IP 协议的 ACL
- IP 审计
- 自反 ACL 和动态 ACL

访问列表编号

用户用来标记 ACL 的编号可以指明用户创建的访问列表的类型。

表 138 中列出了访问列表的编号，以及与编号相对应的访问列表的类型，并且注明了哪些类型是交换机所支持的。交换机能够支持 IPv4 标准访问列表和扩展访问列表，标准访问列表的编号是从 1 至 199 号，扩展访问列表的编号是从 1300 至 2699 号。

表 138：访问列表编号

访问列表编号	类型	是否支持
1-99	IP 标准访问列表	支持
100-199	IP 扩展访问列表	支持
200-299	协议类型代码访问列表	不支持
300-399	DECnet 访问列表	不支持
400-499	XNS 标准访问列表	不支持
500-599	XNS 扩展访问列表	不支持
600-699	AppleTalk 访问列表	不支持
700-799	48 比特 MAC 地址访问列表	不支持
800-899	IPX 标准访问列表	不支持
900-999	IPX 扩展访问列表	不支持
1000-1099	IPX SAP 访问列表	不支持
1100-1199	扩展的 48 比特 MAC 地址访问列表	不支持

1200-1299	IPX 汇总地址访问列表	不支持
1300-1399	IP 标准访问列表（延伸范围）	支持
2000-2699	IP 扩展访问列表（延伸范围）	支持

用户除了创建编号的标准和扩展 ACL 外，还可以通过使用相应的编号，来创建命名的标准和扩展 IP ACL。也就是说，标准 IP ACL 的名字可以是 1 至 99 中的任意数字；扩展 IP ACL 的名字可以是 100 至 199 中的任意数字。相比于编号的访问列表，命名的 ACL 的优势在于，用户可以在不删除整个 ACL 的情况下，单独删除其中的某些列表条目。

编号的标准 IPv4 ACL

请切记，当用户在创建一个 ACL 的时候，在这个 ACL 的结尾会默认自动生成一条隐藏的拒绝所有数据包的语句，也就是说，如果数据包不能与 ACL 中前面配置的条件匹配成功，那么，默认就会被拒绝。在配置标准访问列表条目的时候，如果用户在 IP 地址后面没有配置通配符掩码，那么 ACL 会默认把这个通配符掩码假定为 0.0.0.0。

交换机会自动重新排列标准访问列表中条目的顺序，这样做是为了让那些与 **host** 匹配的条目和使用*无所谓*掩码 0.0.0.0 的条目位于列表的顶端，使它们位于使用非零*无所谓*掩码的条目的前面。因此，在 **show** 命令的输出中，以及在配置文件中，ACE 并不会以用户输入的顺序显示出来。

创建完成后，用户可以将这个编号的标准 IPv4 ACL 应用在 VLAN、终端线路或接口上。

编号的扩展 IPv4 ACL

由于标准 ACL 只能根据源地址进行匹配，为了满足更精准的控制需求，用户可以使用扩展 ACL，因为扩展 ACL 是根据源地址、目的地址，以及可选的协议类型信息进行匹配操作的。当用户创建了一个编号的扩展访问列表时，请切记，在这个 ACL 的结尾也会默认自动生成一条隐藏的，拒绝所有数据包的语句。此外，用户不能对编号的访问列表中的条目进行重新排序，也不能插入新的条目或单独移除已有条目。

交换机无法支持动态访问列表和自反访问列表，也无法支持基于服务类型（ToS）最低位的过滤操作。

当用户在 ACL 中配置某些协议的时候，需要为这些协议设置具体的参数和关键字。

用户可以定义一个扩展的 TCP、UDP、ICMP、IGMP 或其他 IP ACL。交换机能够支持下列的 IP 协议。

注释： ICMP echo-reply 消息不能被过滤，除此之外，所有的 ICMP 编码或类型都可以被过

滤。

交换机支持的 IP 协议如下所示：

- 认证头部协议（**ahp**）
- 封装安全负载（**esp**）
- 增强型内部网关路由协议（**eigrp**）
- 通用路由封装（**gre**）
- Internet 控制消息协议（**icmp**）
- Internet 组管理协议（**igmp**）
- 所有的内部协议（**ip**）
- IP-in-IP 隧道协议（**ipinip**）
- 兼容 KA9Q NOS 的 IP-over-IP 隧道协议（**nos**）
- 开放式最短路径优先路由协议（**ospf**）
- 负载压缩协议（**pcp**）
- 协议无关多播（**pim**）
- 传输控制协议（**tcp**）
- 用户数据报协议（**udp**）

命名的 IPv4 ACL

用户可以通过字母和数字组成的字符串为 IPv4 ACL 命名，而不像之前介绍的那样为 ACL 进行编号。相比于编号的方式，用户使用命名的 ACL 可以在一台路由器上配置更多的 IPv4 访问列表。在配置的方式和命令的语法方面，命名的 ACL 也与编号的 ACL 不尽相同。此外，并不是所有应用于配置 IP 访问列表的命令，都可以适用于命名的访问列表。

注释： 用户也可以使用数字为标准或扩展 ACL 命名，所使用的数字范围要符合访问列表编号的规则。也就是说，如果用户希望使用数字为一个标准 IP ACL 命名的话，可以使用的数字范围就是 1 至 99。相比于编号的列表，命名的 ACL 的优势在于，用户可以单独地删除列表中的条目。

用户在配置命名的 ACL 之前，应该考虑如下的指导建议：

- 也可以使用编号的 ACL；
- 用户不能使用同一个名字来命名一个标准 ACL 和一个扩展 ACL。

ACL 日志

交换机的软件可以为用户提供，那些被标准 IP 访问列表允许或拒绝的数据包的日志消息。也就是说，任何与 ACL 成功匹配的数据包，交换机都会为这个数据包生成一条日志消息，并将该日志消息发送给Console接口。用户可以使用控制系统日志消息的命令**logging concole**，来控制发往 Console 接口的日志消息的等级。

注释： 由于交换机的路由操作是由硬件实现的，而日志却是由软件产生的，因此如果有大量的数据包成功匹配上了包含 **log** 关键字的 *permit* 或 *deny* 的 ACE，有可能会出现软件处理速率无法跟上硬件处理速率的情况，那么此时就不是所有的数据包都具有日志消息了。

第一个与 ACL 成功匹配的数据包会立即触发 ACL 产生一个日志消息，软件会收集之后 5 分钟之内所有匹配成功的数据包的信息，以产生它们的日志消息。无论 ACL 允许还是拒绝数据包，日志消息中都会包含访问列表编号、数据包的源 IP 地址，以及 5 分钟内从这个源地址允许或拒绝的数据包数量。

注释： 如果有太多要处理的日志消息，或者如果在 1 秒钟之内有一个以上的日志消息要处理，日志记录工具可能就丢弃一些日志消息包。这种行为是为了防止路由器由于需要处理太多的日志记录数据包而崩溃。因此，用户不应该把日志记录工具作为审计工具，或者作为访问列表匹配数量的准确来源。

IP ACL 的硬件和软件处理

ACL 的处理是在硬件中执行的。如果硬件中到达了用来储存 ACL 配置的容量极限，那么指定接口上的所有数据包都会被丢弃。

注释： 如果由于交换机或堆叠成员上的资源不足，而无法在硬件中实施 ACL 配置，则只有交换机上接收到的指定 VLAN 中的流量才会受到影响。

在用户输入了特权 EXEC 命令 **show ip access-lists** 后，命令的输出信息中显示的匹配计数值不考虑在硬件中进行访问控制的数据包。用户可以使用特权 EXEC 命令来获取交换和路由数据包的一些基本硬件 ACL 统计信息。

VLAN map 的配置指导

VLAN map 是控制 VLAN 内部流量过滤的唯一方法。VLAN map 没有方向。用户如果要通过使用 VLAN map 来过滤特定方向的流量，就需要在 ACL 中指定具体的源或目标地址。如果对于

一个类型的数据包（IP 或 MAC），在 VLAN map 中指定了这种类型数据包的匹配命令，则交换机会默认当数据包与 VLAN map 中的任何条目都不匹配时，丢弃该数据包。如果 VLAN map 中没有定义该类型数据包的匹配命令，则默认转发数据包。

用户可以在配置 VLAN map 是使用以下配置指导：

- 如果接口上没有配置 ACL 来拒绝流量，并且没有配置 VLAN map 的话，所有流量都会被放行；
- 每个 VLAN map 都由一系列条目构成。VLAN map 中的条目顺序至关重要。交换机接收到的数据包都会与 VLAN map 中的第 1 个条目进行匹配。如果相匹配，交换机就会对其应用 VLAN map 中这一部分下配置的行为。如果不相匹配，数据包会与 map 中的下一个条目进行匹配；
- 如果 VLAN map 中为某个类型的数据包（IP 或 MAC）配置了至少一条匹配命令，那么与这些匹配条件不相符的数据包默认都会被丢弃。如果 VLAN map 中没有某个类型的数据包匹配条目，那么默认交换机会转发这个类型的数据包；
- VLAN map 不支持日志消息；
- 当用户把一个 IP 访问列表或 MAC 访问列表应用在交换机的二层接口上，并且为这个端口所属的 VLAN 应用了 VLAN map，那么端口 ACL 的优先级会高于 VLAN map；
- 如果用户不能把 VLAN map 的配置应用在硬件中，那么指定 VLAN 中的数据包都会被丢弃。

VLAN map 和路由器 ACL

要想同时对桥接流量和路由流量应用访问控制，用户可以只使用 VLAN map，或者结合使用路由器 ACL 和 VLAN map。用户可以在路由 VLAN 接口上同时指定入向和出向的路由器 ACL，并且可以定义一个 VLAN map 来对桥接流量实施访问控制。

如果数据包流与 ACL 中的 VLAN map 拒绝语句相匹配，那么无论用户是否配置了路由器 ACL 配置，数据包流都会被拒绝。

注释： 当用户结合使用了路由器 ACL 和 VLAN map，当路由器 ACL 中需要生成日志消息的数据包，与 VLAN map 中的拒绝语句相匹配，那么这个数据包并不会被记录。

如果 VLAN map 中为某个类型的数据包（IP 或 MAC）配置了匹配命令，那么与这些匹配条件不相符的数据包默认都会被丢弃。如果 VLAN map 中没有配置匹配条目，并且没有指定任何行为，那么如果数据包不匹配任何 VLAN map 条目的话，交换机就会转发这个数据包。

VLAN map 和路由器 ACL 的配置指导

如果用户想要在相同的 VLAN 中配置路由器 ACL 和 VLAN map，可以使用以下配置指导。这些指导并不适用于用户想要把路由器 ACL 和 VLAN map 应用于不同 VLAN 的情况。

如果用户必须在同一个 VLAN 上同时配置路由器 ACL 和 VLAN map 的话，可以使用以下指导来实施路由器 ACL 和 VLAN map 配置：

- 用户可以在一个 VLAN 接口上，在每个方向上（入向/出向）只配置一个 VLAN map 和一个路由器 ACL；
- 如果可能的话，用户可以尝试为 ACL 中的所有条目都配置单独的行为，除了最后为其他类型应用的默认行为。也就是说，以下面两种格式之一来编写 ACL：

  permit… permit… permit… deny ip any any

  或者

  deny… deny… deny… permit ip any any
- 要想在一个 ACL 中定义多个行为（permit 或 deny），用户可以把每个行为类型结合起来，来减少条目数量；
- 用户要避免在 ACL 中包含第 4 层信息；添加这种信息会增加合并过程的难度。如果 ACL 中定义的是根据 IP 地址（源和目的）进行过滤，而不是根据完整的流（源 IP 地址、目的 IP 地址、协议和协议端口）进行过滤的话，会得到最好的合并结果；

  如果用户需要使用完整的流匹配模式，并且 ACL 中同时包含 IP ACE，以及携带四层信息的 TCP/UDP/ICMP CE 的话，用户要把四层 ACE 放到列表的最后。让列表优先基于 IP 地址进行过滤。

ACL 的时间范围

用户可以使用全局配置命令 **time-range**，基于一天中的时间和星期来有选择地应用扩展 ACL。首先，用户要定义一个时间范围名称，并设置时间和日期，或者设置具体的星期几。然后用户要在把它应用到 ACL 时输入指定的时间范围名称，以此来对访问列表的行为进行限制。用户可以使用时间范围来限定 ACL 中的 permit 或 deny 语句何时生效，举例来说，在指定时间周期中生效，或者在指定的星期几范围内生效。命名的和编号的扩展 ACL 任务表中可以调用关键字 **time-range** 和参数。

使用时间范围有以下好处：

- 用户可以对允许或拒绝一个用户访问资源的行为施加更多的控制，比如指定一个应用

（通过 IP 地址/掩码对，以及端口号进行标识）；

- 用户可以控制日志消息的生成，可以使 ACL 条目只在指定的星期几对流量进行记录。这样一来，用户可以在高峰时段只简单地拒绝流量访问，而无需对生成的大量日志进行分析。

基于时间的访问列表会触发 CPU 的活动，因为新的访问列表配置必须与其他特性进行合并，结合后的配置会加载到硬件内存中。出于这个考虑，用户应该注意不要让多个访问列表连续生效（每个访问列表的生效时间只间隔短短几分钟）。

注释： 时间范围的工作依赖于交换机系统的时钟；因此用户需要设置一个可靠的时钟源。我们建议用户使用网络时间协议（NTP）来同步交换机时钟。

IPv4 ACL 接口的考量因素

当用户在一个三层接口（SVI 接口、三层 EtherChannel 接口或路由端口）上使用接口配置命令 **ip access-group** 时，接口上必须已经配置了 IP 地址。三层访问列表会过滤由 CPU 进行处理的三层路由出去的或接收到的数据包。它不会对 VLAN 内部的桥接数据包带来任何影响。对于出向 ACL 来说，在接收到数据包后，交换机会用 ACL 来检查这个数据包。如果 ACL 中允许这个数据包，交换机就会继续处理这个数据包。如果 ACL 中拒绝这个数据包，交换机就会丢弃这个数据包。

在默认环境中，当数据包被丢弃时，入站接口会发送 ICMP 不可达消息，无论这个丢弃数据包的行为是由于入站接口上的 ACL 导致的，还是由于出站接口上的 ACL 导致的。每个入站接口每半秒钟只能发送一个 ICMP 不可达消息，但用户可以使用全局配置命令 **ip icmp rate-limit unreachable** 来改变这一限制。

当用户把一个还未定义的 ACL 应用在接口后，交换机会当作这个接口上还没有应用任何 ACL，并且会放行所有数据包。如果用户想要使用未定义的 ACL 来提供网络安全的话，要记得交换机的这种行为。

如何配置 ACL

配置 IPv4 ACL

用户可以按照以下步骤在交换机上配置 IP ACL。

总步骤

1. 通过指定访问列表编号或名称，以及指定访问条件，来创建一个 ACL；

2. 把这个 ACL 应用在接口或终端线路上。用户也可以在 VLAN map 中应用标准和扩展 IP ACL。

具体步骤

	命令或操作	**目的**
步骤 1	通过指定访问列表编号或名称，以及指定访问条件，来创建一个 ACL	
步骤 2	把这个 ACL 应用在接口或终端线路上。用户也可以在 VLAN map 中应用标准和扩展 IP ACL	

创建编号的标准 ACL

用户可以按照以下步骤来创建编号的标准 ACL。

总步骤

1. enable

2. configure terminal

3. access-list *access-list-number* {**deny** | **permit**} *source source-wildcard*]

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**access-list** *access-list-number* {**deny** | **permit**} *source source-wildcard*]	使用源地址和通配符掩码来定义一个标准的 IPv4 访问列表。

	示例： `Device(config)# access-list 2 deny your_host`	在 *access-list* 部分指定一个十进制数值，取值范围是 1 至 99，或 1300 至 1999。 输入 **deny** 和 **permit** 来指定当条件匹配时，拒绝或放行数据包。 在 *source* 部分指定源地址，也就是从指定网络或主机发出的这个数据包，用户可以指定以下信息： • 32 比特点分十进制数值； • 关键字 **any** 是缩写表达，表示 *source* 和 *source-wildcard* 分别是 0.0.0.0 255.255.255.255。用户无需输入完整的源和通配符掩码。 （可选）*source-wildcard* 会在源地址上应用通配符掩码比特。 **注释：** 用户只能够在三层接口上关联的 ACL 中应用日志功能
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置编号的扩展 ACL

用户可以按照以下步骤来创建编号的扩展 ACL。

总步骤

1. configure terminal

2. access-list *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** tos] [**fragments**] [**log** [**log-input**] [**time-range** *time-range-name*] [**dscp** *dscp*]

3. access-list *access-list-number* {**deny** | **permit**} **tcp** *source source-wildcard* [*operator port*] *destination destination-wildcard* [*operator port*] [**established**] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**log** [**log-input**] [**time-range** *time-range-name*] [**dscp** *dscp*] [*flag*]

4. access-list *access-list-number* {**deny** | **permit**} **udp** *source source-wildcard* [*operator port*] *destination destination-wildcard* [*operator port*] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**log** [**log-input**] [**time-range** *time-range-name*] [**dscp** *dscp*]

5. access-list *access-list-number* {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* [*icmp-type* | [[*icmp-type icmp-code*] | [*icmp-message*]] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**time-range** *time-range-name*] [**dscp** *dscp*]

6. access-list *access-list-number* {**deny** | **permit**} **igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**log** [**log-input**] [**time-range** *time-range-name*] [**dscp** *dscp*]

7. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** tos] [**fragments**] [**log**	定义一个扩展的 IPv4 访问列表和访问条件。 在 *access-list-number* 部分指定一个十进制数值，取值范围是 100 至 199，或 2000 至 2699。

	[**log-input**] [**time-range** *time-range-name*] [**dscp** *dscp*] **示例：** `Device(config)#` **access-list 101 permit ip host 10.1.1.2 any precedence 0 tos 0 log**	输入 **deny** 和 **permit** 来指定当条件匹配时，拒绝或放行数据包。 在 *protocol* 部分输入一个 IP 协议的名称或编号：**ahp**、**eigrp**、**esp**、**gre**、**icmp**、**igmp**、**igrp**、**ip**、**ipinip**、**nos**、**ospf**、**pcp**、**pim**、**tcp** 或 **udp**，或者使用代表一个 IP 协议编号的 0 至 255 之间的整数值。要想匹配任何 Internet 协议（包括 ICMP、TCP 和 UDP），用户需要使用关键字 **ip**。 **注释：** 这个步骤中包含了大多数 IP 协议选项。有关 TCP、UDP、ECMP 和 IGMP 的特定参数，用户可参考以下步骤。 *source* 部分指定了发送数据包的网络号或主机号。 *source-wildcard* 部分指定了源的通配符掩码比特。 *destination* 部分指定了数据包发往的网络号或主机号。 *destination-wildcard* 部分指定了目的的通配符掩码比特。 在配置 *source*、*source-wildcard*、*destination* 和 *destination-wildcard* 时，用户可以指定以下信息： • 32 比特的点分十进制数值 • 使用关键字 **any** 来表示 0.0.0.0 255.255.255.255（任意主机） • 使用关键字 **host** 来表示单台主机 0.0.0.0 用户还可以在这条命令中配置以下关键字，这些关键字的含义如下所示： • **precedence**——输入用来匹配数据

		包的优先级，用户可以使用编号 0 至 7，也可以使用名称：**routine**（0）、**priority**（1）、**immediate**（2）、**flash**（3）、**flash-override**（4）、**critical**（5）、**internet**（6）、**network**（7）； • **fragments**——输入这个关键字来检查非初始分片； • **tos**——输入用来匹配数据包的服务类型级别，用户可以使用编号 0 至 15，也可以使用名称：**normal**（0）、**max-reliability**（2）、**max-throughput**（4）、**min-delay**（8）； • **log**——输入这个关键字来创建发送到 Console 的有关匹配数据包的消息，或者输入 **log-input** 在日志条目中包含入站接口； • **time-range**——指定时间范围名称； • **dscp**——输入数据包匹配的 DSCP 值，取值范围是 0 至 63，或者使用问号（?）来查看可用值列表。 **注释：** 如果用户输入了一个 **dscp** 值，就不能再输入 **tos** 或 **precedence** 值了。用户可以在不使用 **dscp** 值的情况下，同时使用 **tos** 和 **precedence**
步骤 3	**access-list** *access-list-number* {**deny** | **permit**} **tcp** *source source-wildcard* [*operator port*] *destination destination-wildcard* [*operator port*] [**established**] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**log** [**log-input**]	定义一个扩展 TCP 访问列表，并指定访问条件。 这条命令中的参数与扩展 IPv4 ACL 中描述的参数相同，除了以下内容： （可选）输入 *operator* 和 *port* 来对比源端口（如果在 *source source-wildcard*

	[**time-range** *time-range-name*] [**dscp** *dscp*] [*flag*] **示例：** `Device(config)# access-list 101 permit tcp any any eq 500`	后面输入的话）或目的端口（如果在 *destination destination-wildcard* 后面输入的话）。可选的运算符包括 **eq**（等于）、**gt**（大于）、**lt**（小于）、**neq**（不等于）和 **range**（包含首尾数值的范围）。运算操作需要有一个对应的端口号（使用关键字 **range** 时需要配置两个端口号，中间以空格分隔）。 在 *port* 部分指定 TCP 端口的十进制数值（取值范围是 0 至 65535）或名称。在过滤 TCP 时，用户只能使用 TCP 端口号或名称。 用户可以配置其他可选关键字，其含义如下所示： • **established**——输入这个关键字来匹配已建立的连接。这个关键字的功能与匹配 **ack** 或 **rst** 标记的功能相同 • *flag*——输入以下标记来匹配指定的 TCP 头部比特：**ack**（确认）、**fin**（完成）、**psh**（推送）、**rst**（重置）、**syn**（同步）或 **urg**（紧急）
步骤 4	**access-list** *access-list-number* {**deny** | **permit**} **udp** *source source-wildcard* [*operator port*] *destination destination-wildcard* [*operator port*] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**log** [**log-input**] [**time-range** *time-range-name*] [**dscp** *dscp*] **示例：**	（可选）定义一个扩展的 UDP 访问列表，并指定访问条件。 UDP 参数与上一步骤中描述的 TCP 参数相同，除了[operator [port]]端口号或名称必须是 UDP 端口号或名称，并且关键字 **flag** 和 **established** 不适用于 UDP

	Device(config)# **access-list 101 permit udp any any eq 100**	
步骤 5	**access-list** *access-list-number* {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* [*icmp-type* | [[*icmp-type icmp-code*] | [*icmp-message*]] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**time-range** *time-range-name*] [**dscp** *dscp*] **示例：** Device(config)# **access-list 101 permit icmp any any 200**	定义一个扩展的 ICMP 访问列表，并指定访问条件。 ICMP 参数与扩展 IPv4 ACL 中大多数 IP 协议的参数相同，除了 ICMP 中还可以指定 ICMP 消息类型和代码参数。这些可选关键字的含义如下所示： • *tcmp-type*——输入这个参数来过滤 ICMP 消息类型，取值范围是 0 至 255 之间的数值 • *icmp-code*——输入这个参数来过滤 ICMP 数据包，并根据 ICMP 消息代码类型进行过滤，取值范围是 0 至 255 之间的数值 • *icmp-message*——输入这个参数来过滤 ICMP 数据包，并根据 ICMP 消息类型名称或 ICMP 消息类型和代码名称进行过滤
步骤 6	**access-list** *access-list-number* {**deny** | **permit**} **igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**log** [**log-input**] [**time-range** *time-range-name*] [**dscp** *dscp*] **示例：** Device(config)# **access-list 101 permit igmp any any 14**	（可选）定义一个扩展的 IGMP 访问列表，并指定访问条件。 IGMP 参数与扩展 IPv4 ACL 中大多数 IP 协议的参数相同，同时还包括以下可选参数： *igmp-type*——为了匹配 IGMP 消息类型，用户可以输入 0 至 15 之间的数值，或者输入消息名称：**dvmrp**、**host-query**、**host-report**、**pim** 或 **trace**
步骤 7	**end**	返回特权 EXEC 模式

	示例： `Device(config)# end`	

创建命名的标准 ACL

用户可以按照以下步骤来创建使用名称的标准 ACL：

总步骤

1. enable

2. configure terminal

3. ip access-list standard *name*

4. 使用以下命令之一：

• **deny** {*source* [*source-wildcard*] | **host** *source* | **any**} [**log**]

• **permit** {*source* [*source-wildcard*] | **host** *source* | **any**} [**log**]

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip access-list standard** *name* **示例：** `Device(config)# ip access-list standard 20`	使用名称定义一个标准 IPv4 访问列表，并进入 access-list 配置模式。 名称也可以是 1 至 99 之间的数值
步骤 4	使用以下命令之一：	在 access-list 配置模式中，指定一个或

	• **deny** {*source* [*source-wildcard*] | **host** *source* | **any**} [**log**] • **permit** {*source* [*source-wildcard*] | **host** *source* | **any**} [**log**] **示例：** `Device(config-std-nacl)#` **`deny 192.168.0.0 0.0.255.255`** **`255.255.0.0 0.0.255.255`** 或者 `Device(config-std-nacl)#` **`permit 10.108.0.0 0.0.0.0`** **`255.255.255.0 0.0.0.0`**	多个条件，以及拒绝或允许行为，来决定转发数据包还是丢弃数据包。 • **host** *source*——表示的源和源通配符掩码为 *source* 0.0.0.0 • **any**——表示的源和源通配符掩码为 0.0.0.0 255.255.255.255
步骤 5	**end** **示例：** `Device(config-std-nacl)#` **`end`**	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device#` **`show running-config`**	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device#` **`copy running-config`** **`startup-config`**	（可选）把输入的命令保存到配置文件中

创建命名的扩展 ACL

用户可以按照以下步骤来创建使用名称的扩展 ACL：

总步骤

1. enable

2. configure terminal

3. ip access-list extended *name*

4. {**deny** | **permit**} *protocol* {*source* [*source-wildcard*] | **host** *source* | **any**} {*destination* [*destination-wildcard*] | host *destination* | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**] [**time-range** *time-range-name*]

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**ip access-list extended** *name* **示例：** `Device(config)#` **`ip access-list extended 150`**	使用名称定义一个扩展的 IPv4 访问列表，并进入 access-list 配置模式。 名称可以是 100 至 199 之间的数值
步骤 4	{**deny** | **permit**} *protocol* {*source* [*source-wildcard*] | **host** *source* | **any**} {*destination* [*destination-wildcard*] | host *destination* | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**] [**time-range** *time-range-name*] **示例：** `Device(config-ext-nacl)#` **`permit 0 any any`**	在 access-list 配置模式中，指定允许或拒绝的条件。用户可以使用关键字 **log** 来获得访问列表的日志消息，其中包括违规行为。 • **host** *source*——表示的源和源通配符掩码为 *source* 0.0.0.0 • **host** *destination*——表示的目的和目的通配符掩码为 *destination* 0.0.0.0 • **any**——表示的源和源通配符掩

		码，或者目的和目的通配符掩码为 0.0.0.0 255.255.255.255
步骤 5	**end** **示例：** `Device(config-ext-nacl)# end`	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

当用户在创建扩展 ACL 时，要记住在默认情况下，如果在到达 ACL 末尾之前都没有找到匹配条目，那么它就会与 ACL 末尾匹配所有数据包的隐含 deny 语句相匹配。对于标准 ACL 来说，如果用户在相关联的 IP 主机地址访问列表的定义中省略了掩码，则默认使用 0.0.0.0 为掩码。

在创建 ACL 后，用户添加的所有条目都会被放置在列表的末尾。用户不能有选择地在 ACL 中的指定位置添加 ACL 条目。但用户可以使用 access-list 配置模式命令 **no permit** 和 **no deny** 从命名 ACL 中删除 ACL 条目。

能够有选择地从命名 ACL 中删除 ACL 条目，是用户使用命名 ACL 而不是编号 ACL 的一个理由。

接下来做什么？

在创建命名的 ACL 后，用户可以把它应用在接口或 VLAN 上。

为 ACL 配置时间范围

用户可以按照以下步骤来为 ACL 配置时间范围参数：

总步骤

1. enable

2. configure terminal

3. time-range *time-range-name*

4. 使用以下命令之一：

• **absolute** [**start** *time date*] [**end** *time date*]

• **periodic** *day-of-the-week hh:mm to* [*day-of-the-week*] *hh:mm*

• **periodic** {**weekdays** | **weekend** | **daily**} *hh:mm to hh:mm*

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**time-range** *time-range-name* **示例：** `Device(config)# time-range workhours`	为创建的时间范围指定一个有意义的名称（比如 *workhours*[工作时间]），并进入 time-range 配置模式。名称中不能包含空格或问号，并且必须以字母开头
步骤 4	使用以下命令之一： **absolute** [**start** *time date*] [**end** *time date*] **periodic** *day-of-the-week hh:mm to* [*day-of-the-week*] *hh:mm* **periodic** {**weekdays**	**weekend**	**daily**} *hh:mm to hh:mm* **示例：**	指定何时使用配置的操作。 • 用户可以在时间范围中只使用一个 **absolute** 语句。如果用户配置了多个 absolute 语句，只有最后配置的会生效 • 用户可以输入多个 **periodic** 语句。举例来说，用户可以在不同的工作日和周末中配置不同的小时 用户可以参考示例配置

	Device(config-time-range)# **absolute start 00:00 1 Jan 2006 end 23:59 1 Jan 2006** 或者 Device(config-time-range)# **periodic weekdays 8:00 to 12:00**	
步骤 5	**end** **示例：** Device(config-time-range)# **end**	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** Device# **show running-config**	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

接下来做什么？

如果用户需要指定多个分别在不同时间运行的项目的话，就需要多次重复这个配置步骤。

在终端线路上应用 IPv4 ACL

用户可以使用编号 ACL 来控制一个或多个终端线路的访问行为。用户不能在线路上应用命名 ACL。用户必须对所有的虚拟终端线路设置相同的限制，因为用户可以尝试连接到其中任何一个线路。

用户可以按照以下步骤来限制虚拟终端线路和 ACL 中指定地址之间入站和出站的连接：

总步骤

1. enable

2. configure terminal

3. line [**console** | **vty**] *line-number*

4. access-class *access-list-number* {**in** | **out**}

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**line** [**console**	**vty**] *line-number* **示例：** `Device(config)# line console 0`	指明要配置的线路，并进入线路配置模式。 • **console**——指定 Console 终端线路。Console 端口是 DCE； • **vty**——指定用于远程 Console 访问的虚拟终端。 在指定了线路类型后，在 *line-number* 部分配置用户希望配置的第 1 个线路编号。取值范围是 0 至 16
步骤 4	**access-class** *access-list-number* {**in**	**out**} **示例：** `Device(config-line)# access-class 10 in`	限制指定的虚拟终端线路（进入一台设备）和访问列表中指定地址之间的入站和出站连接
步骤 5	**end**	返回特权 EXEC 模式

	示例： `Device(config-line)# end`	
步骤 6	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

在接口上应用 IPv4 ACL

这部分描述了如何在网络接口上应用 IPv4 ACL。

从特权 EXEC 模式开始，用户可以按照以下步骤来对接口实施访问控制：

总步骤

1. configure terminal

2. interface *interface-id*

3. ip access-group {*access-list-number* | *name*} {**in** | **out**}

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**interface** *interface-id* **示例：**	指定用户想要配置的接口，并进入接口配置模式。 接口可以是二层接口（端口 ACL），或

	Device(config)# **interface gigabitethernet1/0/1**	者三层接口（路由器 ACL）
步骤 3	**ip access-group** {*access-list-number* | *name*} {**in** | **out**} **示例：** Device(config-if)# **ip access-group 2 in**	对指定接口实施访问控制
步骤 4	**end** **示例：** Device(config-if)# **end**	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** Device# **show running-config**	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

创建命名的 MAC 扩展 ACL

用户可以使用 MAC 地址和命名的 MAC 扩展 ACL，在 VLAN 或二层接口上实施非 IPv4 流量的过滤。这个配置过程与配置其他命名的扩展 ACL 类似。

用户可以按照以下步骤来创建命名的 MAC 扩展 ACL：

总步骤

1. enable

2. configure terminal

3. mac access-list extended *name*

4. {**deny** | **permit**} {**any** | **host** *source MAC address* | *source MAC address mask*} {**any** | **host** *destination MAC address* | *destination MAC address mask*} [*type mask* | **lsap** *lsap mask* | **aarp** |

amber | **dec-spanning** | **decnet-iv** | **diagnostic** | **dsm** | **etype-6000** | **etype-8042** | **lat** | **lavc-sca** | **mop-console** | **mop-dump** | **msdos** | **mumps** | **netbINOS** | **vines-echo** | **vines-ip** | **xns-idp** | 0-*65535*] [**cos** *cos*]

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**mac access-list extended** *name* **示例：** `Device(config)# mac access-list extended mac1`	使用名称定义一个扩展 MAC 访问列表
步骤 4	{**deny** | **permit**} {**any** | **host** *source MAC address* | *source MAC address mask*} {**any** | **host** *destination MAC address* | *destination MAC address mask*} [*type mask* | **lsap** *lsap mask* | **aarp** | **amber** | **dec-spanning** | **decnet-iv** | **diagnostic** | **dsm** | **etype-6000** | **etype-8042** | **lat** | **lavc-sca** | **mop-console** | **mop-dump** | **msdos** | **mumps** | **netbINOS** | **vines-echo** | **vines-ip** | **xns-idp** | 0-*65535*] [**cos** *cos*]	在扩展 MAC access-list 配置模式中，指定 **permit** 或 **deny** 任意源 MAC 地址、源 MAC 地址/掩码，或者指定 **host**（主机）源 MAC 地址和 **any**（任意）目的 MAC 地址、目的 MAC 地址/掩码，或者指定目的 MAC 地址。 （可选）用户也可以输入以下选项： • *type mask*——任意 Ethernet II 或 SNAP 封装类型的数据包 EtherType 编号，格式为十进制、十六进制，或八进制，也可以在进行匹配前在

	示例： **decnet-iv** 或者 Device(config-ext-macl)# **permit any any**	EtherType 上应用可选的*无所谓*比特掩码 • **lsap** *lsap mask*——IEEE 802.2 封装类型的数据包 LSAP 编号，格式为十进制、十六进制，或八进制，还可选的使用*无所谓*比特掩码 • **aarp** | **amber** | **dec-spanning** | **decnet-iv** | **diagnostic** | **dsm** | **etype-6000** | **etype-8042** | **lat** | **lavc-sca** | **mop-console** | **mop-dump** | **msdos** | **mumps** | **netbINOS** | **vines-echo** | **vines-ip** | **xns-idp**——非 IP 协议 • **cos** *cos*——IEEE 802.1Q 服务类别编号，取值范围是 0 至 7，用来设置优先级
步骤 5	**end** 示例： Device(config-ext-macl)# **end**	返回特权 EXEC 模式
步骤 6	**show running-config** 示例： Device# **show running-config**	检查用户输入的信息
步骤 7	**copy running-config startup-config** 示例： Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

在二层接口上应用 MAC ACL

用户可以按照以下命令在二层接口上应用 MAC 访问列表来实施访问控制：

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. mac access-group {*name*} {**in** | **out** }

5. end

6. show mac access-group [**interface** *interface-id*]

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id* **示例：** `Device(config)#` **`interface gigabitethernet1/0/2`**	指定一个接口，并进入接口配置模式。这个接口必须是物理的二层接口（端口 ACL）
步骤 4	**mac access-group** {*name*} {**in**	**out** } **示例：** `Device(config-if)#` **`mac access-group mac1 in`**	通过使用 MAC 访问列表来对指定接口实施访问控制。 用户可以在出方向上和入方向上应用端口 ACL
步骤 5	**end** **示例：**	返回特权 EXEC 模式

	Device(config-if)# **end**	
步骤 6	**show mac access-group** [**interface** *interface-id*] **示例：** Device# **show mac access-group interface gigabitethernet1/0/2**	显示应用在指定接口或所有二层接口上的 MAC 访问列表
步骤 7	**show running-config** **示例：** Device# **show running-config**	检查用户输入的信息
步骤 8	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

在接收到一个数据包后，交换机会使用入方向的 ACL 来检查这个数据包。如果 ACL 中允许数据包通过，交换机就会继续处理这个数据包。如果 ACL 中拒绝数据包，交换机就会丢弃这个数据包。当用户在接口上应用了一个未定义的 ACL 后，交换机会当作这个接口上还没有应用任何 ACL，并且会放行所有数据包。如果用户想要使用未定义的 ACL 来提供网络安全的话，要记得交换机的这种行为。

配置 VLAN map

用户可以按照以下命令来创建 VLAN map 并把它应用在一个或多个 VLAN 上：

在开始前

用户需要先创建想要应用在 VLAN 上的标准或扩展 IPv4 ACL，或者命名的 MAC 扩展 ACL。

总步骤

1. vlan access-map *name* [**number**]

2. match {**ip** | **mac**} **address** {*name* | *number*} [*name* | *number*]

3. 输入以下命令之一，来指定 IP 数据包或非 IP 数据包（只能以已知的 MAC 地址进行指定），并且使用一个或多个（标准或扩展）ACL 来匹配数据包：

• **action** { **forward**}

Device(config-access-map)# **action forward**

• **action** { **drop**}

Device(config-access-map)# **action drop**

4. vlan filter *mapname* **vlan-list** *list*

具体步骤

	命令或操作	目的
步骤 1	**vlan access-map** *name* [**number**] **示例：** Device(config)# **vlan access-map map_1 20**	创建一个 VLAN map，并指定一个名称和（可选）一个编号。编号是这个 map 中条目的序列号。 在用户使用相同的名称创建 VLAN map 时，条目的编号是以 10 递增的。在更改或删除 VLAN map 时，用户可以输入想要更改或删除的 map 条目编号。 VLAN map 中并不能指定 permit 或 deny 关键字。要想使用 VLAN map 拒绝一个数据包，用户需要创建一个 ACL 来进行数据包匹配，并设置丢弃行为。ACL 中的 permit 语句表示相匹配。ACL 中的 deny 语句表示不匹配。输入这条命令会进入 access-map 配置模式
步骤 2	**match** {**ip** | **mac**} **address** {*name* | *number*} [*name* | *number*] **示例：** Device(config-access-map)# **match ip address ip2**	（使用 IP 或 MAC 地址）通过一个或多个标准或扩展访问列表来匹配数据包。需要注意的是，数据包只会以正确的协议类型来匹配访问列表。用户需要使用标准或扩展 IP 访问列表来匹配 IP 数据包。用户需要使用命名的 MAC 扩展访问列表来匹配非 IP 数据包。 **注释：** 如果用户配置 VLAN map 来

		匹配一类数据包（IP 或 MAC），并且 VLAN map 中的行为是丢弃，那么所有匹配这个类型的数据包都会被丢弃。如果 VLAN map 中没有配置匹配条件，并且配置了丢弃行为，那么所有 IP 和二层数据包都会被丢弃

步骤 3	输入以下命令之一，来指定 IP 数据包或非 IP 数据包（只能以已知的 MAC 地址进行指定），并且使用一个或多个（标准或扩展）ACL 来匹配数据包：
	• **action** {**forward**}
	`Device(config-access-map)# action forward`
	• **action** {**drop**}
	`Device(config-access-map)# action drop`
		为 map 条目设置行为

步骤 4	**vlan filter** *mapname* **vlan-list** *list* **示例：** `Device(config)# vlan filter map 1 vlan-list 20-22`	把 VLAN map 应用到一个或多个 VLAN ID。 *list* 可以是一个 VLAN ID（22）、一个连续的列表（10 - 22），或者多个 VLAN ID（12, 22, 30）。逗号和连字符前后的空格是可选的

创建一个 VLAN map

每个 VLAN map 都由一系列有序的条目组成。从特权 EXEC 模式开始，用户可以按照以下步骤来创建、添加或删除 VLAN map 条目：

总步骤

1. configure terminal

2. vlan access-map *name* [**number**]

3. match {**ip** | **mac**} **address** {*name* | *number*} [*name* | *number*]

4. action {**drop** | **forward**}

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 2	**vlan access-map** *name* [**number**] **示例：** `Device(config)#` **`vlan access-map map_1 20`**	创建一个 VLAN map，并指定一个名称和（可选）一个编号。编号是这个 map 中条目的序列号。 在用户使用相同的名称创建 VLAN map 时，条目的编号是以 10 递增的。在更改或删除 VLAN map 时，用户可以输入想要更改或删除的 map 条目编号。 VLAN map 中并不能指定 permit 或 deny 关键字。要想使用 VLAN map 拒绝一个数据包，用户需要创建一个 ACL 来进行数据包匹配，并设置丢弃行为。ACL 中的 permit 语句表示相匹配。ACL 中的 deny 语句表示不匹配。 输入这条命令会进入 access-map 配置模式
步骤 3	**match** {**ip** | **mac**} **address** {*name* | *number*} [*name* | *number*] **示例：** `Device(config-access-map)#` **`match ip address ip2`**	（使用 IP 或 MAC 地址）通过一个或多个标准或扩展访问列表来匹配数据包。需要注意的是，数据包只会以正确的协议类型来匹配访问列表。用户需要使用标准或扩展 IP 访问列表来匹配 IP 数据包。用户需要使用命名的 MAC 扩展访问列表来匹配非 IP 数据包。 **注释：** 如果用户配置 VLAN map 来匹配一类数据包（IP 或 MAC），并且 VLAN

		map 中的行为是丢弃，那么所有匹配这个类型的数据包都会被丢弃。如果 VLAN map 中没有配置匹配条件，并且配置了丢弃行为，那么所有 IP 和二层数据包都会被丢弃
步骤 4	**action {drop | forward}** **示例：** `Device(config-access-map)#` **`action forward`**	（可选）为 map 条目设置行为，默认行为是转发
步骤 5	**end** **示例：** `Device(config-access-map)#` **`end`**	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device#` **`show running-config`**	检查用户输入的信息
步骤 7	**copy running-config startup-config** **示例：** `Device#` **`copy running-config startup-config`**	（可选）把输入的命令保存到配置文件中

在 VLAN 上应用 VLAN map

从特权 EXEC 模式开始，用户可以按照以下步骤在一个或多个 VLAN 上应用 VLAN map：

总步骤：

1. enable

2. configure terminal

3. vlan filter *mapname* **vlan-list** *list*

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**vlan filter** *mapname* **vlan-list** *list* **示例：** `Device(config)# vlan filter map 1 vlan-list 20-22`	在一个或多个 VLAN ID 上应用 VLAN map。 *list* 可以是一个 VLAN ID（22）、一个连续的列表（10 - 22），或者多个 VLAN ID（12, 22, 30）。逗号和连字符前后的空格是可选的
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

监控 IPv4 ACL

用户可以通过查看交换机上配置的 ACL，以及查看应用在接口和 VLAN 上的 ACL，来监控 IPv4 ACL。

在用户使用接口配置命令 **ip access-group**，在二层或三层接口上应用 ACL 时，用户可以查看接口上的 access-group。用户也可以查看应用在二层接口上的 MAC ACL。用户可以使用下面这个表格中展示的特权 EXEC 命令。

表 139：显示 access-list 和 access-group 的命令

命令	目的
show access-lists [*number* | *name*]	显示一个或当前所有 IP 和 MAC 地址访问列表或一个指定访问列表（编号或命名）中的内容
show ip access-lists [*number* | *name*]	显示当前所有 IP 访问列表或指定 IP 访问列表（编号或命名）中的内容
show ip interface *interface-id*	显示一个接口的配置和状态。如果接口上启用了 IP 功能，并且用户使用接口配置命令 **ip access-group** 应用了 ACL，命令的输出内容中就会包含 access-group
show running-config [**interface** *interface-id*]	显示交换机或指定接口的配置文件内容，其中包括所有配置的 MAC 和 IP 访问列表，以及接口上应用的 access-group
show mac access-group [**interface** *interface-id*]	显示应用在所有二层接口或指定接口上的 MAC 访问列表。 二层接口

ACL 的配置示例

示例：在 ACL 中使用时间范围

这个示例展示了在用户配置了时间范围 *workhours*，并把 200 年 1 月 1 日设置为公司假期后，

如何验证相关的配置内容。

```
Device# show time-range
time-range entry: new_year_day_2003 (inactive)
absolute start 00:00 01 January 2006 end 23:59 01 January 2006
time-range entry: workhours (inactive)
periodic weekdays 8:00 to 12:00
periodic weekdays 13:00 to 17:00
```

要想应用一个时间范围，用户需要在能够实施时间范围的扩展 ACL 中，输入时间范围名称。这个示例展示了如何创建和验证扩展访问列表 188 的信息，这个访问列表中拒绝了从任意源去往任意目的地的 TCP 流量，应用的时间范围是假期时间，但在工作时间允许所有 TCP 流量。

```
Device(config)# access-list 188 deny tcp any any time-range new_year_day_2006
Device(config)# access-list 188 permit tcp any any time-range workhours
Device(config)# end
Device# show access-lists
Extended IP access list 188
10 deny tcp any any time-range new_year_day_2006 (inactive)
20 permit tcp any any time-range workhours (inactive)
```

下面这个示例展示了使用命名的 ACL 来放行和拒绝相同的流量。

```
Device(config)# ip access-list extended deny_access
Device(config-ext-nacl)# deny tcp any any time-range new_year_day_2006
Device(config-ext-nacl)# exit
Device(config)# ip access-list extended may_access
Device(config-ext-nacl)# permit tcp any any time-range workhours
Device(config-ext-nacl)# end
Device# show ip access-lists
Extended IP access list lpip_default
10 permit ip any any
Extended IP access list deny_access
10 deny tcp any any time-range new_year_day_2006 (inactive)
```

```
Extended IP access list may_access
10 permit tcp any any time-range workhours (inactive)
```

示例：ACL 中包含的命令

用户可以使用 **remark** 关键字，在任意 IP 标准或扩展 ACL 中包含一些注释（备注）信息。备注信息能够让用户更容易理解和搜索 ACL。每个备注信息限制为 100 个字符。

用户可以在 permit 或 deny 语句的前后设置备注信息。用户应该总是在同样的位置设置备注信息，这样就能看得出来哪条备注是在描述哪条 permit 或 deny 语句了。举例来说，如果有些备注标记在 permit 或 deny 语句前，有些标记在语句后，就会让用户感到混乱。

要想在编号的 IP 标准或扩展 ACL 中包含备注信息，用户需要使用全局配置命令 **access-list** *access-list number* **remark** *remark*。要想移除备注，需要使用这条命令的 **no** 格式。

在这个示例中，用户放行了属于 Jones 的工作站，并拒绝了属于 Smith 的工作站：

```
Device(config)# access-list 1 remark Permit only Jones workstation through
Device(config)# access-list 1 permit 171.69.2.88
Device(config)# access-list 1 remark Do not allow Smith through
Device(config)# access-list 1 deny 171.69.3.13
```

对于命名 IP ACL 中的条目，用户需要使用 access-list 配置命令 **remark**。要想移除备注，需要使用这条命令的 **no** 格式。

在这个示例中，Jones 子网不允许使用出向 Telnet：

```
Device(config)# ip access-list extended telnetting
Device(config-ext-nacl)# remark Do not allow Jones subnet to telnet out
Device(config-ext-nacl)# deny tcp host 171.69.2.88 any eq telnet
```

IPv4 ACL 配置示例

这部分提供了配置和应用 IPv4 ACL 的示例。配置 ACL 的具体信息，用户可以参考 *Inspur INOS Security Configuration Guide, Release 12.4*，以及 *Inspur INOS IP Configuration Guide, Release 12.4* 中“IP Addressing and Services”一章中的“Configuring IP Services”部分。

小型网络办公室中的 ACL

这一部分展示了一个小型网络办公室环境，其中路由端口 2 连接着服务器 A，服务器 A 上包含有效益信息和其他信息，所有雇员都可以访问。路由端口 1 连接着服务器 B，服务器 B 上包含保密的工资数据。所有用户都可以访问服务器 A，但服务器 B 是被限制访问的。

图 105：使用路由器 ACL 来控制流量

Server A Benefits	服务器 A 效益
Server B Payroll	服务器 B 工资
Port 2	端口 2
Port1	端口 1
Human Resources	人力资源
Accounting	审计

用户可以使用以下两种方式之一来应用路由器 ACL：

- 创建一个标准 ACL，过滤从端口 1 去往服务器的流量；
- 创建一个扩展 ACL，过滤从端口 1 进入的服务器流量。

示例：小型网络办公室中的 ACL

这个示例中使用了一个标准 ACL 来过滤从端口进入的服务器 B 流量，只允许审计部门的源地址 172.20.128.64 至 172.20.128.95。这个 ACL 用来匹配从指定源地址去往路由端口 1 的流量。

```
Device(config)# access-list 6 permit 172.20.128.64 0.0.0.31
Device(config)# end
Device# show access-lists
Standard IP access list 6
10 permit 172.20.128.64, wildcard bits 0.0.0.31
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip access-group 6 out
```

下面这个示例中使用了一个扩展 ACL，来过滤从服务器 B 进入端口的流量，允许任意源地址（本例中是服务器 B）流量只能去往审计部门的目的地址 172.20.128.64 至 172.20.128.95。

这个 ACL 用来匹配进入路由端口 1 的流量，并且只允许这些流量去往指定目的地。注意在配置扩展 ACL 时，用户必须在源和目的信息前输入协议（IP）信息。

```
Device(config)# access-list 106 permit ip any 172.20.128.64 0.0.0.31
Device(config)# end
Device# show access-lists
Extended IP access list 106
10 permit ip any 172.20.128.64 0.0.0.31
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip access-group 106 in
```

示例：编号的 ACL

在这个示例中，网络 36.0.0.0 上一个 A 类网络，它的第 2 个八位组用来表示子网；也就是说它的子网掩码是 255.255.0.0。网络地址 36.0.0.0 中的第 3 和第 4 八位组用来指定具体的主机。用户使用了访问列表 2，让交换机接受子网 48 上的 1 个地址，拒绝这个子网上的所有其他地址。列表中的最后一行显示出交换机要接受所有其他网络 36.0.0.0 的子网。用户把这个 ACL 应用在进入端口的数据包上。

```
Device(config)# access-list 2 permit 36.48.0.3
Device(config)# access-list 2 deny 36.48.0.0 0.0.255.255
Device(config)# access-list 2 permit 36.0.0.0 0.255.255.255
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# ip access-group 2 in
```

示例：扩展 ACL

在这个示例中，第一行允许任意入站 TCP 连接，并且目的端口要大于 1023。第二行允许入站 TCP 连接，并且要去往主机 128.88.1.2 的简单邮件传输协议（SMTP）端口。第三行允许用于错误反馈的入站 ICMP 消息。

```
Device(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255
gt 1023
Device(config)# access-list 102 permit tcp any host 128.88.1.2 eq 25
Device(config)# access-list 102 permit icmp any any
Device(config)# interface gigabitethernet2/0/1
```

```
Device(config-if)# ip access-group 102 in
```

在这个示例中，假设用户的网络连接到 Internet，并且用户希望网络中的任意主机能够与 Internet 上的任意主机建立 TCP 连接。但是，用户不希望 IP 主机能够与自己网络中的主机建立 TCP 连接，除了指定邮件主机的邮件（SMTP）端口。

SMTP 在一端使用 TCP 端口 25，在另一端使用随机端口号。在连接建立后到断开前都会使用相同的端口号。从 Internet 进入的邮件数据包的目的端口号是 25。出向数据包的端口号正相反。因为网络的安全系统总是会接受端口 25 上的邮件连接，因此入站和出站服务是分别进行控制的。用户必须在出向接口上配置一个入站 ACL，并且在入向接口上配置一个出站 ACL。

```
Device(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255
eq 23
Device(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255
eq 25
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip access-group 102 in
```

在这个示例中的网络是一个 B 类网络，地址为 128.88.0.0，邮件主机地址为 128.88.1.2。用户只为 TCP 连接使用了 **established** 关键字，以此显示已建立的连接。当 TCP 数据包中设置了 ACK 或 RST 位，就认为数据包匹配，这表示数据包是属于一个已存在的连接。硬件号码 1 上的 GigabitEthernet 接口 1 就是连接去往 Internet 的路由器的接口。

```
Device(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255
established
Device(config)# access-list 102 permit tcp any host 128.88.1.2 eq 25
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip access-group 102 in
```

示例：命名的 ACL

创建命名的标准和扩展 ACL

这个示例中创建了一个标准 ACL，名称为 *Internet_filter*，并创建了一个扩展 ACL，名称为 *marketing_group*。*Internet_filter* ACL 放行了源地址为 1.2.3.4 的所有流量。

```
Device(config)# ip access-list standard Internet_filter
Device(config-ext-nacl)# permit 1.2.3.4
Device(config-ext-nacl)# exit
```

marketing_group ACL 允许去往目的地址和通配符掩码 171.69.0.0 0.0.255.255 的任意 TCP Telnet 流量，并拒绝所有其他 TCP 流量。ACL 允许 ICMP 流量，拒绝从任意源地址去往目的地址范围 171.69.0.0 至 172.69.255.255，且目的端口号小于 1024 的 UDP 流量，拒绝所有其他 IP 流量，并为匹配结果提供日志消息。

```
Device(config)# ip access-list extended marketing_group
Device(config-ext-nacl)# permit tcp any 171.69.0.0 0.0.255.255 eq telnet
Device(config-ext-nacl)# deny tcp any any
Device(config-ext-nacl)# permit icmp any any
Device(config-ext-nacl)# deny udp any 171.69.0.0 0.0.255.255 lt 1024
Device(config-ext-nacl)# deny ip any any log
Device(config-ext-nacl)# exit
```

用户为三层端口的出站流量应用了 *Internet_filter* ACL，为三层端口的入站流量应用了 *marketing_group* ACL。

```
Device(config)# interface gigabitethernet3/0/2
Device(config-if)# no switchport
Device(config-if)# ip address 2.0.5.1 255.255.255.0
Device(config-if)# ip access-group Internet_filter out
Device(config-if)# ip access-group marketing_group in
```

从命名的 ACL 中删除指定的 ACE

这个示例展示了人如何从命名访问列表 *border-list* 中删除指定的 ACE：

```
Device(config)# ip access-list extended border-list
Device(config-ext-nacl)# no permit ip host 10.1.1.3 any
```

示例：为 IP ACL 应用时间范围

用户在这个示例中拒绝了 HTTP 流量，执行时间为周一至周五，8:00 至 18:00。这个示例只有在周六和周日的 12:00 至 20:00 之间，才放行 UDP 流量。

```
Device(config)# time-range no-http
Device(config)# periodic weekdays 8:00 to 18:00
!
Device(config)# time-range udp-yes
Device(config)# periodic weekend 12:00 to 20:00
```

```
!
Device(config)# ip access-list extended strict
Device(config-ext-nacl)# deny tcp any any eq www time-range no-http
Device(config-ext-nacl)# permit udp any any time-range udp-yes
!
Device(config-ext-nacl)# exit
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# ip access-group strict in
```

示例：配置备注 IP ACL 条目

在这个示例中用户配置了一个编号的 ACL，允许属于 Jones 的工作站的访问行为，并拒绝属于 Smith 的工作站的访问行为：

```
Device(config)# access-list 1 remark Permit only Jones workstation through
Device(config)# access-list 1 permit 171.69.2.88
Device(config)# access-list 1 remark Do not allow Smith workstation through
Device(config)# access-list 1 deny 171.69.3.13
```

这个示例中用户配置了一个编号的 ACL，其中拒绝 Winter 和 Smith 工作站使用浏览 Web：

```
Device(config)# access-list 100 remark Do not allow Winter to browse the web
Device(config)# access-list 100 deny host 171.69.3.85 any eq www
Device(config)# access-list 100 remark Do not allow Smith to browse the web
Device(config)# access-list 100 deny host 171.69.3.13 any eq www
```

在这个示例中用户配置了一个命名的 ACL，拒绝了 Jones 子网的访问行为：

```
Device(config)# ip access-list standard prevention
Device(config-std-nacl)# remark Do not allow Jones subnet through
Device(config-std-nacl)# deny 171.69.0.0 0.0.255.255
```

在这个示例中用户配置了一个命名的 ACL，拒绝了 Jones 子网使用出向 Telnet：

```
Device(config)# ip access-list extended telnetting
Device(config-ext-nacl)# remark Do not allow Jones subnet to telnet
```

out

```
Device(config-ext-nacl)# deny tcp 171.69.0.0 0.0.255.255 any eq telnet
```

示例：ACL 日志

路由器 ACL 支持两种日志记录。**log** 关键字能够向 Console 发送与该条目匹配的数据包的信息性日志消息，；**log-input** 关键字会在日志条目中包含输入接口信息。

在这个示例中用户配置了一个命名的标准访问列表 *stan1*，拒绝了来自 10.1.1.0 0.0.0.255 的流量，允许来自所有其他源地址的流量，并在命令中包含了 **log** 关键字。

```
Device(config)# ip access-list standard stan1
Device(config-std-nacl)# deny 10.1.1.0 0.0.0.255 log
Device(config-std-nacl)# permit any log
Device(config-std-nacl)# exit
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip access-group stan1 in
Device(config-if)# end
Device# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
Console logging: level debugging, 37 messages logged
Monitor logging: level debugging, 0 messages logged
Buffer logging: level debugging, 37 messages logged
File logging: disabled
Trap logging: level debugging, 39 message lines logged
Log Buffer (4096 bytes):
00:00:48: NTP: authentication delay calculation problems
<output truncated>
00:09:34:%SEC-6-IPACCESSLOGS:list stan1 permitted 0.0.0.0 1 packet
00:09:59:%SEC-6-IPACCESSLOGS:list stan1 denied 10.1.1.15 1 packet
00:10:11:%SEC-6-IPACCESSLOGS:list stan1 permitted 0.0.0.0 1 packet
```

在这个示例中用户配置了一个命名的扩展访问列表 *ext1*，允许从任意源去往 10.1.1.0 0.0.0.255 的 ICMP 数据包，并拒绝所有 UDP 数据包。

```
Device(config)# ip access-list extended ext1
```

```
Device(config-ext-nacl)# permit icmp any 10.1.1.0 0.0.0.255 log
Device(config-ext-nacl)# deny udp any any log
Device(config-std-nacl)# exit
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# ip access-group ext1 in
```

这个示例展示的是一个扩展 ACL 的日志消息：

```
01:24:23:%SEC-6-IPACCESSLOGDP:list ext1 permitted icmp 10.1.1.15 ->
10.1.1.61 (0/0), 1
packet
01:25:14:%SEC-6-IPACCESSLOGDP:list ext1 permitted icmp 10.1.1.15 ->
10.1.1.61 (0/0), 7
packets
01:26:12:%SEC-6-IPACCESSLOGP:list ext1 denied udp 0.0.0.0(0) ->
255.255.255.255(0), 1 packet
01:31:33:%SEC-6-IPACCESSLOGP:list ext1 denied udp 0.0.0.0(0) ->
255.255.255.255(0), 8 packets
```

注意所有 IP ACL 条目的日志消息都是以%SEC-6-IPACCESSLOG 开头的，并且根据 ACL 的类别和匹配的访问条目，这些信息会有些许变化。

这个示例展示的是一个启用了 **log-input** 关键字的输出消息：

```
00:04:21:%SEC-6-IPACCESSLOGDP:list inputlog permitted icmp 10.1.1.10
(Vlan1 0001.42ef.a400)
->
10.1.1.61 (0/0), 1 packet
```

使用 **log** 关键字记录的相同数据包日志消息中不包含入站接口信息：

```
00:05:47:%SEC-6-IPACCESSLOGDP:list inputlog permitted icmp 10.1.1.10
-> 10.1.1.61 (0/0), 1
packet
```

ACL 和 VLAN map 的配置示例

示例：创建 ACL 和 VLAN map 来拒绝数据包

这个示例展示了如何创建一个 ACL 和一个 VLAN map，来拒绝数据包。在第一个 map 中，所有匹配 *ip1* ACL（TCP 数据包）的数据包都会被丢弃。用户首先创建 *ip1* ACL，在其中允许所有 TCP 数据包，并拒绝其他数据包。由于 VLAN map 中有一个匹配 IP 数据包的条目，默认行为是丢弃所有不匹配条件的 IP 数据包。

```
Device(config)# ip access-list extended ip1
Device(config-ext-nacl)# permit tcp any any
Device(config-ext-nacl)# exit
Device(config)# vlan access-map map_1 10
Device(config-access-map)# match ip address ip1
Device(config-access-map)# action drop
```

示例：创建 ACL 和 VLAN map 来允许数据包

这个示例展示了如何创建一个 VLAN map 来放行数据包。这个示例中使用了 ACL *ip2*，允许 UDP 数据包，并且所有匹配 *ip2* ACL 的数据包都会被转发。在这个 map 中，所有不匹配之前 ACL 的 IP 数据包（也就是那些既不是 TCP 数据包，也不是 UDP 数据包的数据包）都会被丢弃。

```
Device(config)# ip access-list extended ip2
Device(config-ext-nacl)# permit udp any any
Device(config-ext-nacl)# exit
Device(config)# vlan access-map map_1 20
Device(config-access-map)# match ip address ip2
Device(config-access-map)# action forward
```

示例：丢弃 IP 数据包和转发 MAC 数据包的默认行为

在这个示例中，VLAN map 中指定了丢弃 IP 数据包的默认行为，以及转发 MAC 数据包的默认行为。这个 VLAN map 与标准 ACL 101 和命名的扩展访问列表 **igmp-match** 和 **tcp-match** 结

合使用，map 会执行以下行为：

- 转发所有 UDP 数据包
- 丢弃所有 IGMP 数据包
- 转发所有 TCP 数据包
- 丢弃所有其他 IP 数据包
- 转发所有非 IP 数据包

```
Device(config)# access-list 101 permit udp any any
Device(config)# ip access-list extended igmp-match
Device(config-ext-nacl)# permit igmp any any
Device(config-ext-nacl)# permit tcp any any
Device(config-ext-nacl)# exit
Device(config)# vlan access-map drop-ip-default 10
Device(config-access-map)# match ip address 101
Device(config-access-map)# action forward
Device(config-access-map)# exit
Device(config)# vlan access-map drop-ip-default 20
Device(config-access-map)# match ip address igmp-match
Device(config-access-map)# action drop
Device(config-access-map)# exit
Device(config)# vlan access-map drop-ip-default 30
Device(config-access-map)# match ip address tcp-match
Device(config-access-map)# action forward
```

示例：丢弃 MAC 数据包和转发 IP 数据包的默认行为

在这个示例中，VLAN map 中配置了丢弃 MAC 数据包的默认行为，以及转发 IP 数据包的默认行为。这个 VLAN map 与 MAC 扩展访问列表 **good-hosts** 和 **good-protocols** 结合使用，map 会执行以下行为：

- 转发来自主机 0000.0c00.0111 和 0000.0c00.0211 的 MAC 数据包
- 转发携带 decnet-iv 或 vines-ip 协议的 MAC 数据包
- 丢弃所有其他非 IP 数据包
- 转发所有 IP 数据包

示例：丢弃所有数据包的默认行为

在这个示例中，VLAN map 中配置了丢弃所有数据包（IP 和非 IP）的默认行为。这个 VLAN map 与示例 2 和 3 中的访问列表 **tcp-match** 和 **good-hosts** 结合使用，map 会执行以下行为：

- 转发所有 TCP 数据包
- 转发来自主机 0000.0c00.0111 和 0000.0c00.0211 的 MAC 数据包
- 丢弃所有其他 IP 数据包
- 丢弃所有其他 MAC 数据包

```
Device(config)# vlan access-map drop-all-default 10
Device(config-access-map)# match ip address tcp-match
Device(config-access-map)# action forward
Device(config-access-map)# exit
Device(config)# vlan access-map drop-all-default 20
Device(config-access-map)# match mac address good-hosts
Device(config-access-map)# action forward
```

在用户网络中使用 VLAN map 的配置示例

示例：配线柜的配置

在配线柜的配置中，交换机上可能没有启用路由功能。在这种配置中，交换机仍可以支持 VLAN map 和 QoS 分类 ACL。假设主机 X 和主机 Y 分别位于不同的 VLAN，并且分别连接到配线柜交换机 A 和 C。从主机 X 去往主机 Y 的流量最终会由交换机 B 进行路由，交换机 B 是启用了路由功能的三层交换机。从主机 X 去往主机 Y 的流量可以在流量进入交换机 A 时收到访问控制。

图 106：配线柜的配置

Switch B	交换机 B
Switch A	交换机 A
Switch C	交换机 C
VLAN map: Deny HTTP from X to Y	VLAN map：拒绝 HTTP 从 X 去往 Y
HTTP is dropped	HTTP 是在进入

at entry point.	位置被丢弃的
Host X	主机 X
Host Y	主机 Y
Packet	数据包

如果用户不希望交换机转发从主机 X 去往主机 Y 的 HTTP 流量，用户可以在交换机 A 上配置一个 VLAN map，丢弃从主机 X（IP 地址 10.1.1.32）去往主机 Y（IP 地址 10.1.1.34）的所有 HTTP 流量，并且不会把这些流量转发到交换机 B。

首先，用户需要定义 IP 访问列表 *http*，允许（匹配）HTTP 端口上的所有 TCP 流量。

```
Device(config)# ip access-list extended http
Device(config-ext-nacl)# permit tcp host 10.1.1.32 host 10.1.1.34 eq
www
Device(config-ext-nacl)# exit
```

接下来，用户要创建 VLAN map *map2*，使它丢弃与访问列表 *http* 匹配的流量，并转发所有其他 IP 流量。

```
Device(config)# vlan access-map map2 10
Device(config-access-map)# match ip address http
Device(config-access-map)# action drop
Device(config-access-map)# exit
Device(config)# ip access-list extended match_all
Device(config-ext-nacl)# permit ip any any
Device(config-ext-nacl)# exit
Device(config)# vlan access-map map2 20
Device(config-access-map)# match ip address match_all
Device(config-access-map)# action forward
```

然后，把 VLAN map *map2* 应用到 VLAN 1。

```
Device(config)# vlan filter map2 vlan 1
```

示例：限制访问另一个 VLAN 上的服务器

用户可以丢弃访问另一个 VLAN 上服务器的流量。举例来说，VLAN 10 中的服务器 10.1.1.100 需要拒绝下列主机的访问：

- 应该拒绝 VLAN 20 中子网 10.1.2.0/8 中主机的访问；
- 应该拒绝 VLAN 10 中主机 10.1.1.4 和 10.1.1.8 的访问。

图 107：限制访问另一个 VLAN 上的服务器

Server (VLAN 10)	服务器（VLAN 10）
Host (VLAN 10)（共 2 处）	主机（VLAN 10）
Layer 3 switch	三层交换机
Subnet	子网
Host (VLAN 20)	主机（VLAN 20）

示例：拒绝访问另一个 VLAN 上的服务器

这个示例展示了用户如何通过创建 VLAN map SERVER1_ACL，来拒绝访问另一个 VLAN 上服务器的流量，这个 VLAN map 拒绝了去往子网 10.1.2.0/8、主机 10.1.1.4 和主机 10.1.1.8 的流量，并允许其他 IP 流量。最后一步用户把 map SERVER1_ACL 应用到 VLAN 10。

用户先定义 IP ACL 来匹配正确的数据包。

```
Device(config)# ip access-list extended SERVER1_ACL
Device(config-ext-nacl))# permit ip 10.1.2.0 0.0.0.255 host
10.1.1.100
Device(config-ext-nacl))# permit ip host 10.1.1.4 host 10.1.1.100
Device(config-ext-nacl))# permit ip host 10.1.1.8 host 10.1.1.100
Device(config-ext-nacl))# exit
```

用户定义一个 VLAN map，在其中调用这个 ACL，丢弃匹配 SERVER1_ACL 的 IP 数据包，转发不匹配这个 ACL 的 IP 数据包。

```
Device(config)# vlan access-map SERVER1_MAP
Device(config-access-map)# match ip address SERVER1_ACL
Device(config-access-map)# action drop
Device(config)# vlan access-map SERVER1_MAP 20
Device(config-access-map)# action forward
Device(config-access-map)# exit
```

用户把 VLAN map 应用到 VLAN 10。

```
Device(config)# vlan filter SERVER1_MAP vlan-list 10
```

在 VLAN 上应用路由器 ACL 和 VLAN map 的配置示例

在这部分展示的示例中，用户在 VLAN 上应用了路由器 ACL 和 VLAN map，应用于交换、桥

接、路由和组播数据包。虽然在下面的展示中，数据包都被转发到了它们的目的地，但每次数据包路径上应用了 VLAN map 或 ACL 时，数据包也可能被丢弃，而不是被转发。

示例：ACL 和被交换的数据包

在这个示例中，展示了如何为在 VLAN 内部进行交换的数据包应用 ACL。在 VLAN 内部进行交换的数据包不会进行路由，或者由回退-桥接进行转发，因此它会受到入站 VLAN 的 VLAN map 影响。

图 108：在被交换的数据包上应用 ACL

Input router ACL	入站 路由器 ACL
Output router ACL	出站 路由器 ACL
Frame	数据帧
Host A	主机 A
Host C	主机 C
Packet	数据包

示例：ACL 和被桥接的数据包

这个示例中展示了如何在回退-桥接的数据包上应用 ACL。对于桥接的数据包，只能在入站 VLAN 上应用二层 ACL。只有非 IP、非 ARP 数据包可以进行回退-桥接。

图 109：在桥接的数据包上应用 ACL

Host A	主机 A
Frame	数据帧
Host B	主机 B
Fallback bridge	回退桥接
Packet	数据包

示例：ACL 和被路由的数据包

这个示例展示了如何在被路由的数据包上应用 ACL。ACL 是按照以下顺序应用的：

1. 入站 VLAN 的 VLAN map
2. 入站路由器 ACL
3. 出站路由器 ACL
4. 出站 VLAN 的 VLAN map

图 110：在被路由的数据包上应用 ACL

Input router ACL	入站 路由器 ACL
Output router ACL	出站 路由器 ACL
Frame	数据帧
Host A	主机 A
Host B	主机 B
Routing function	路由功能
Packet	数据包

示例：ACL 和组播数据包

这个示例中展示了如何在通过 IP 组播进行复制的数据包上应用 ACL。被路由的组播数据包有两种应用过滤的方式：一个用来匹配入站 VLAN 中其他端口的目的地，另一个用来匹配其他 VLAN（数据包被路由的 VLAN）中的每个目的地。数据包可能会被路由到多个出现出向 VLAN，在这种情况下，用户可以为每个目的 VLAN 应用不同的路由器出向 ACL 和 VLAN map。最终结果是有些出向 VLAN 可能会放行数据包，而在其他出向 VLAN 中则拒绝。数据包的副本会被转发到被放行的那些目的地。但是，如果入向 VLAN map 丢弃了数据包，则没有目的地能够接收到这个数据包的副本。

图 111：在组播数据包上应用 ACL

Input router	入站 路由器

ACL	ACL
Output router ACL	出站 路由器 ACL
Frame	数据帧
Host A	主机 A
Host B	主机 B
Routing function	路由功能
Host C	主机 C
Packet	数据包

其他参考资料

相关主题

相关主题	文档名称
IPv4 访问控制列表主题	Securing the Data Plane Configuration Guide Library, Inspur INOS XE Release 3SE (Inspur 6850 Switches) http://www.icntnetworks.com

错误消息解码器

描述	链接
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术	http://www.icntnetworks.com

信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	

配置 IPv6 ACL

查询特性信息

用户的软件版本可能无法支持这部分文档所提到的全部特性。想要查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

IPv6 ACL 概述

用户可以通过创建 IPv6 访问控制列表（ACL），并把它们应用到接口，来过滤 IP 版本 6（IPv6）流量，就像创建并应用 IP 版本 4（IPv4）命名 ACL。在运行 IP Base 和 LAN Base 特性集的交

换机上，用户也可以创建并应用入向路由器 ACL，来过滤三层管理流量。

交换机支持以下三种类型的 IPv6 ACL：

- 用户可以为三层接口上的出向或入向流量应用 IPv6 路由器 ACL，这个接口可以是路由端口、交换机虚拟接口（SVI），或三层 EtherChannel。IPv6 路由器 ACL 只应用在被路由的 IPv6 数据包上；
- 入向二层接口上支持 IPv6 端口 ACL。IPv6 端口 ACL 能够应用于进入接口的所有 IPv6 数据包上；
- VLAN ACL 或 VLAN map 能够对一个 VLAN 内部的所有数据包执行访问控制。用户可以使用 VLAN map 来过滤同一个 VLAN 不同设备之间的流量。ACL VLAN map 可以应用在二层 VLAN 上。VLAN map 中可以基于三层地址，对 IPv6 执行访问控制。用户要理解通过 MAC 地址，使用以太网 ACE 执行访问控制的协议。在把 VLAN map 应用到一个 VLAN 后，所有进入这个 VLAN 的数据包都会由 VLAN map 进行检查；

用户可以在一个接口上同时应用 IPv4 和 IPv6 ACL。与 IPv4 ACL 一样，IPv6 端口 ACL 的优先级高于路由器 ACL。

交换机堆栈和 IPv6 ACL

主用交换机能够在硬件中支持 IPv6 ACL，并把这个 IPv6 ACL 分发到堆栈成员上。

如果备用交换机接管并成为了主用交换机，它会把 ACL 的配置分发到所有堆栈成员。成员交换机会与新的主用交换机分发的配置进行同步，并把不需要的条目移除。

在用户修改 ACL、在接口上关联或解除关联 ACL 时，主用交换机会把变更分发给所有堆栈成员。

ACL 优先级

当用户在同一台交换机上配置 VLAN map、端口 ACL 和路由器 ACL 时，对于入向流量来说，这些访问限制按照过滤优先级的从高到底排列为：端口 ACL、VLAN map，然后是路由器 ACL。

对于出向流量来说，过滤优先级排列为：路由器 ACL、VLAN map，然后是端口 ACL。

以下示例描述了简单的使用情况：

- 但用户同时应用了入向端口 ACL 和 VLAN map 时，应用了端口 ACL 的端口在接收到入站数据包时，这些入站数据包会由端口 ACL 进行过滤。其他数据包由 VLAN map 进行过滤；
- 当一个交换机虚拟接口（SVI）上同时应用了入向路由器 ACL 和入向端口 ACL 时，应用了端口 ACL 的端口在接收到入站数据包时，这些入站数据包是由端口 ACL 进行过滤的。

其他端口上接收到的入站路由 IP 数据包是由路由器 ACL 进行过滤的。其他数据包不执行过滤；

- 当一个 SVI 接口上同时应用了出向路由器 ACL 和入向端口 ACL 时，应用了端口 ACL 的端口在接收到入站数据包时，这些入站数据包是由端口 ACL 进行过滤的。出向路由 IP 数据包是由路由器 ACL 进行过滤的。其他数据包不执行过滤；
- 当一个 SVI 接口上同时应用了 VLAN map、入向路由器 ACL 和入向端口 ACL 时，应用了端口 ACL 的端口在接收到入站数据包时，这些入站数据包只会由端口 ACL 进行过滤。其他端口上收到的入站路由 IP 数据包会同时由 VLAN map 和路由器 ACL 进行过滤。其他数据包只会由 VLAN map 进行过滤；
- 当一个 SVI 接口上同时应用了 VLAN map、出向路由器 ACL 和出向端口 ACL 时，应用了端口 ACL 的端口在接收到入站数据包时，这些入站数据包只会由端口 ACL 进行过滤。其他端口上收到的入站路由 IP 数据包会同时由 VLAN map 和路由器 ACL 进行过滤。其他数据包只会由 VLAN map 进行过滤。

VLAN map

用户可以使用 VLAN ACL 或 VLAN map 来对一个 VLAN 内部的网络流量实施控制。用户可以为一台交换机或一个交换机堆栈上，一个 VLAN 内部桥接的所有数据包应用 VLAN map。VACL 专门用来执行安全数据包过滤行为，并且用于把流量重定向到指定的物理接口。VACL 在定义时不涉及方向性（入向或出向）。

所有非 IP 协议都是使用 MAC VLAN map 进行访问控制的，需要匹配 MAC 地址和以太类型（IP 流量不会由 MAC VLAN map 来提供访问控制）。用户可以在穿越交换机的数据包上实施 VLAN map；用户不能在通过集线器或另一台交换机，连接到本地交换机的主机与本地交换机之间的流量上应用 VLAN map。

在使用 VLAN map 时，交换机会根据 map 中指定的行为，来允许或拒绝数据包的转发行为。下图中展示了如何应用 VLAN map 来执行过滤的情景，用户要拒绝来自 VLAN 10 中主机 A 的指定类型流量。用户只可以在一个 VLAN 上应用一个 VLAN map。

图 112：使用 VLAN map 来实施流量控制

Host A	主机 A
Host B	主机 B
VLAN map denying specific type of traffic from Host A	VLAN map 拒绝了来自主机 A 的特性类型流量

Packet	数据包

与其他特性和交换机的互操作

- 如果用户配置了一个 IPv6 路由器 ACL 来拒绝数据包，则数据包不能被路由。这个数据包的一个副本会被发送到 Internet 控制消息协议（ICMP）队列，以便为这个数据帧生成 ICMP 不可达消息；
- 如果由于端口 ACL 而丢弃了一个桥接的数据帧，则这个数据包无法被桥接；
- 用户可以在一台交换机或交换机堆栈上同时创建 IPv4 和 IPv6 ACL，用户也可以在同一个接口上同时应用 IPv4 和 IPv6 ACL。每个 ACL 必须有唯一的名称；如果用户尝试使用已经配置过的名称，就会看到一条错误消息。

  用户需要使用不同的命令来创建 IPv4 和 IPv6 ACL，以及在相同的二层或三层接口上关联 IPv4 或 IPv6 ACL。如果用户在关联一个 ACL 时使用了错误的命令（比如使用 IPv4 命令来关联 IPv6 ACL），用户就会看到一条错误消息；
- 用户不能使用 MAC ACL 来过滤 IPv6 数据帧。MAC ACL 只能用来过滤非 IP 数据帧；
- 如果设备的硬件内存满了，那么数据包会在接口上就被丢弃，并且设备会记录一条 Unload 错误消息。

配置 IPv6 ACL 的限制条件

在 IPv4 中，用户可以配置编号的标准和扩展 IP ACL、命名 IP ACL 和 MAC ACL。IPv6 只支持命名 ACL。

交换机上能够支持大多数 Inspur INOS 所支持的 IPv6 ACL，除了以下注意事项：

- 交换机不支持使用这些关键字进行匹配：**routing header** 和 **undetermined-transport**；
- 交换机不支持自反 ACL（使用关键字 **reflect**）；
- 交换机不能在 IPv6 数据帧上应用基于 MAC 的 ACL；
- 用户不能在二层 EtherChannel 上应用 IPv6 端口 ACL；
- 在配置 ACL 时，对于用户在 ACL 中输入的关键字并没有限制，除非设备平台不支持。当用户在需要执行硬件转发的接口（物理端口或 SVI 接口）上应用 ACL 时，交换机会检查并确认这个接口是否能够支持 ACL。如果不支持，关联 ACL 的行为被拒绝；
- 如果用户在接口上应用了一个 ACL，并且尝试在一个访问控制条目（ACE）中使用接口不支持的关键字，那么交换机不会允许用户在当前关联到接口上的这个 ACL 中添加这

条 ACE。

交换机上的 IPv6 ACL 具有以下特征：

- 支持分片的数据帧（与 IPv4 中 **fragments** 关键字相同）；
- IPv4 中支持的状态统计信息，在 IPv6 ACL 中也同样支持；
- 如果交换机的硬件空间不足，与 ACL 相关联的数据包会在接口上被丢弃；
- 路由器 ACL 能够使用日志功能，端口 ACL 不能使用日志功能；
- 交换机支持使用全范围的前缀长度进行 IPv6 地址匹配。

默认的 IPv6 ACL 配置

以下为默认的 IPv6 ACL 配置：

```
Switch# show access-lists preauth_ipv6_acl
IPv6 access list preauth_ipv6_acl (per-user)
permit udp any any eq domain sequence 10
permit tcp any any eq domain sequence 20
permit icmp any any nd-ns sequence 30
permit icmp any any nd-na sequence 40
permit icmp any any router-solicitation sequence 50
permit icmp any any router-advertisement sequence 60
permit icmp any any redirect sequence 70
permit udp any eq 547 any eq 546 sequence 80
permit udp any eq 546 any eq 547 sequence 90
deny ipv6 any any sequence 100
```

配置 IPv6 ACL

用户可以按照以下步骤来过滤 IPv6 流量：

总步骤

1. enable

2. configure terminal

3. [**no**]{**ipv6 access-list** *list-name*| **client permit-control-packets**| **log-update threshold**| **role-based**

list-name}

4. [**no**]{**deny** | **permit**} protocol {*source-ipv6-prefix/*|*prefix-length*|**any threshold**| **host** *source-ipv6-address*} [operator [*port-number*]] { *destination-ipv6-prefix/ prefix-length* | **any** | **host** *destination-ipv6-address*} [operator [*port-number*]][**dscp** *value*] [**fragments**] [**log**] [**log-input**] [**routing**] [**sequence** *value*] [**time-range** *name*]

5. {**deny** | **permit**} **tcp** {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [**operator** [**port-number**]] {*destination-ipv6- prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [operator [*port-number*]] [**ack**] [**dscp** *value*] [**established**] [**fin**] [**log**] [**log-input**] [**neq** {**port** | protocol}] [**psh**] [**range** {**port** | protocol}] [**rst**] [**routing**] [**sequence** *value*] [**syn**] [**time-range** *name*] [**urg**]

6. {**deny** | **permit**} **udp** {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [operator [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [operator [*port-number*]] [**dscp** *value*] [**log**] [**log-input**] [**neq** {*port* | *protocol*}] [**range** {*port* | *protocol*}] [**routing**] [**sequence** *value*] [**time-range** *name*]]

7. {**deny** | **permit**} **icmp** {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [operator [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [operator [*port-number*]] [*icmp-type* [*icmp-code*] | icmp-message] [**dscp** *value*] [**log**] [**log-input**] [**routing**] [**sequence** *value*] [**time-range** *name*]

8. end

9. show ipv6 access-list

10. show running-config

11. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	[**no**]{**ipv6 access-list** *list-name*| **client**	定义一个 IPv6 ACL 名称，并进入 IPv6

	permit-control-packets| **log-update threshold**| **role-based** *list-name*} **示例：** `Device(config)#` **`ipv6 access-list example_acl_list`**	访问列表配置模式
步骤 4	[**no**]{**deny** | **permit**} protocol {*source-ipv6-prefix/|prefix-length*|**any threshold**| **host** *source-ipv6-address*} [operator [*port-number*]] { *destination-ipv6-prefix/ prefix-length* | **any** | **host** *destination-ipv6-address*} [operator [*port-number*]][**dscp** *value*] [**fragments**] [**log**] [**log-input**] [**routing**] [**sequence** *value*] [**time-range** *name*]	输入 **deny** 和 **permit** 来指定当条件匹配时，拒绝或放行数据包。 • 在 *protocol* 部分输入一个 IP 协议的名称或编号：**ahp**、**esp**、**icmp**、**ipv6**、**pcp**、**stcp**、**tcp** 或 **udp**，或者使用代表一个 IPv6 协议编号的 0 至 255 之间的整数值； • 在 *source-ipv6-prefix/prefix-length* 或 *destination-ipv6-prefix/prefix-length* 部分指定要为其设置拒绝和允许条件的源和目的 IPv6 网络或网络类，使用十六进制的 16 比特数值，以冒号分隔（详见 RFC 2373 文档）； • 使用 **any** 这个缩写来表示 IPv6 前缀::/0； • 在 **host** *source-ipv6-address* 或 *destination-ipv6-address* 部分输入要为其设置拒绝和允许条件的源或目的 IPv6 主机地址，使用十六进制的 16 比特数值，以冒号分隔； • （可选）在 **operator** 部分指定要进行对比运算的指定协议源或目的端口。可选的运算符包括 **lt**（小于）、**gt**（大于）、**eq**（等于）、**neq**（不等于）和 **range**。

		如果 **operator** 后面跟着 *source-ipv6-prefix/prefix-length* 参数，则它必须匹配源端口。如果 **operator** 后面跟着 *destination-ipv6-prefix/prefix-length* 参数，则它必须匹配目的端口； • （可选）在 **port-number** 部分指定 TCP 或 UDP 端口的十进制数值，取值范围是 0 至 65535。在过滤 TCP 时，用户只可以使用 TCP 端口名称。在过滤 UDP 时，用户只可以使用 UDP 端口名称； • （可选）输入 **dscp** 值来匹配差分服务代码点值，来匹配 IPv6 数据包头部中的流量类别字段。取值范围是 0 至 63； • （可选）输入 **fragments** 来检查非初始分片。只有当 protocol 为 ipv6 时，用户才会看到这个关键字； • （可选）输入 **log** 关键字来创建发送到 Console 的有关匹配数据包的消息。输入 **log-input** 在日志条目中包含入站接口。只有路由器 ACL 可以支持日志功能； • （可选）输入 **routing** 来指定被路由的 IPv6 数据包； • （可选）输入 **sequence** *value* 来为访问列表条目指定序列号。可选范围是 1 至 4294967295； • （可选）输入 **time-range** 名称来指定要用来拒绝或允许数据包的时间范围

步骤 5	{**deny** | **permit**} **tcp** {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [**operator** [**port-number**]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [operator [*port-number*]] [**ack**] [**dscp** *value*] [**established**] [**fin**] [**log**] [**log-input**] [**neq** {**port** | protocol}] [**psh**] [**range** {**port** | protocol}] [**rst**] [**routing**] [**sequence** *value*] [**syn**] [**time-range** *name*] [**urg**]	（可选）定义一个 TCP 访问列表和访问条件。 输入 **tcp** 来指定传输控制协议。这条命令中的参数与步骤 3a 中指定的参数相同，此外用户还可以配置以下可选参数： • **ack**——设置确认比特； • **established**——已建立的连接。如果 TCP 数据段中设置了 ACK 或 RST 比特，则认为匹配； • **fin**——设置完成比特；不会再发送任何数据； • **neq** {*port* | protocol}——只匹配携带非指定端口号的数据包； • **psh**——设置推送功能比特； • **range** {*port* | protocol}——只匹配携带端口号范围的数据包； • **rst**——设置重置比特； • **syn**——设置同步比特； • **urg**——设置紧急指针比特
步骤 6	{**deny** | **permit**} **udp** {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [operator [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [operator [*port-number*]] [**dscp** *value*] [**log**] [**log-input**] [**neq** {*port* | *protocol*}] [**range** {*port* | *protocol*}] [**routing**] [**sequence** *value*] [**time-range** *name*]]	（可选）定义一个 UDP 访问列表和访问条件。 输入 **udp** 来指定用户数据报协议。UDP 参数与 TCP 部分描述的参数相同，除了 [operator [*port*]]部分的端口号或名称必须是 UDP 端口好或名称，已建立参数不适用于 UDP
步骤 7	{**deny** | **permit**} **icmp** {*source-ipv6-prefix/prefix-length* | **any** |	（可选）定义一个 ICMP 访问列表和访问条件。

	host *source-ipv6-address*} [operator [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [operator [*port-number*]] [*icmp-type* [*icmp-code*] | icmp-message] [**dscp** *value*] [**log**] [**log-input**] [**routing**] [**sequence** *value*] [**time-range** *name*]	输入 **icmp** 来指定 Internet 控制消息协议。ICMP 参数与步骤 1 中描述的大多数 IP 协议相同，除了用户还可以设置 ICMP 消息类型和代码参数。用户可以使用的可选参数有如下含义： • *icmp-type*——输入这个参数来按照 ICMP 消息类型进行过滤，取值范围为 0 至 255 之间的数值； • *icmp-code*——输入这个参数来按照 ICMP 消息代码类型进行 ICMP 数据包过滤，取值范围为 0 至 255 之间的数值； • *icmp-message*——输入这个参数来按照ICMP消息类型名称或ICMP消息类型和代码名称进行 ICMP 数据包过滤。要想查看完整的 ICMP 消息类型名称和代码名称，用户可以使用?或查看这个版本的命令参考
步骤 8	**end**	返回特权 EXEC 模式
步骤 9	**show ipv6 access-list**	检查访问列表的配置
步骤 10	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 11	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

接下来做什么？

在接口上关联 IPv6 ACL

在接口上关联 IPv6 ACL

用户可以在三层接口的出方向上或入方向上应用 ACL，或者在二层接口的入方向上应用 ACL。

用户也可以只在三层接口的入向管理流量上应用 ACL。

用户可以按照以下步骤，控制接口上的流量访问行为：

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. no switchport

5. ipv6 address *ipv6-address*

6. ipv6 traffic-filter *access-list-name* {**in** | **out**}

7. end

8. show running-config

9. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface** *interface-id*	指定用户想要应用访问列表的二层接口（端口 ACL）或三层接口（路由器 ACL），并进入接口配置模式
步骤 4	**no switchport**	如果要应用路由器 ACL，用户需要使用这条命令把接口从二层模式（默认）更改为三层模式

步骤 5	**ipv6 address** *ipv6-address*	在三层接口（路由器 ACL）上配置 IPv6 地址
步骤 6	**ipv6 traffic-filter** *access-list-name* {**in** | **out**}	为接口上的入站或出站流量应用访问列表。 **注释：**
步骤 7	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 8	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 9	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把输入的命令保存到配置文件中

配置 VLAN map

用户可以按照以下步骤，来创建一个 VLAN map，并将其应用在一个或多个 VLAN 上。

在开始前

用户需要创建想要应用在 VLAN 上的 IPv6 ACL。

总步骤

1. enable

2. configure terminal

3. vlan access-map *name* [**number**]

4. match {**ip** | **ipv6** | **mac**} **address** {*name* | *number*} [*name* | *number*]

5. 输入以下命令之一，来指定 IP 数据包或非 IP 数据包（只能以已知的 MAC 地址进行指定），并且使用一个或多个（标准或扩展）ACL 来匹配数据包：

• **action** { **forward**}

Device(config-access-map)# action forward

• **action** { **drop**}

Device(config-access-map)# action drop

6. vlan filter *mapname* **vlan-list** *list*

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**vlan access-map** *name* [**number**] **示例：** `Device(config)# vlan access-map map_1 20`	创建一个 VLAN map，并指定一个名称和（可选）一个编号。编号是这个 map 中条目的序列号。 在用户使用相同的名称创建 VLAN map 时，条目的编号是以 10 递增的。在更改或删除 VLAN map 时，用户可以输入想要更改或删除的 map 条目编号。 VLAN map 中并不能指定 permit 或 deny 关键字。要想使用 VLAN map 拒绝一个数据包，用户需要创建一个 ACL 来进行数据包匹配，并设置丢弃行为。ACL 中的 permit 语句表示相匹配。ACL 中的 deny 语句表示不匹配。输入这条命令会进入 access-map 配置模式
步骤 4	**match** {**ip** | **ipv6** | **mac**} **address** {*name* | *number*} [*name* | *number*]	通过一个或多个访问列表来匹配数据包。需要注意的是，数据包只会以

	示例： `Device(config-access-map)#` **`match ipv6 address ip_net`**	正确的协议类型来匹配访问列表。用户需要使用 IP 访问列表来匹配 IP 数据包。用户需要使用命名的 MAC 访问列表来匹配非 IP 数据包。 **注释：** 如果用户配置 VLAN map 来匹配一类数据包（IP 或 MAC），并且 VLAN map 中的行为是丢弃，那么所有匹配这个类型的数据包都会被丢弃。如果 VLAN map 中没有配置匹配条件，并且配置了丢弃行为，那么所有 IP 和二层数据包都会被丢弃

步骤 5	输入以下命令之一，来指定 IP 数据包或非 IP 数据包（只能以已知的 MAC 地址进行指定），并且使用一个或多个（标准或扩展）ACL 来匹配数据包：
	• **action** {**forward**}
	`Device(config-access-map)# action forward`
	• **action** {**drop**}
	`Device(config-access-map)# action drop`
		为 map 条目设置行为

步骤 6	**vlan filter** *mapname* **vlan-list** *list* **示例：** `Device(config)#` **`vlan filter map 1 vlan-list 20-22`**	把 VLAN map 应用到一个或多个 VLAN ID。 *list* 可以是一个 VLAN ID（22）、一个连续的列表（10 - 22），或者多个 VLAN ID（12, 22, 30）。逗号和连字符前后的空格是可选的

在 VLAN 上应用 VLAN map

从特权 EXEC 模式开始，用户可以按照以下步骤在一个或多个 VLAN 上应用 VLAN map：

总步骤：

1. enable

2. configure terminal

3. vlan filter *mapname* **vlan-list** *list*

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**vlan filter** *mapname* **vlan-list** *list* **示例：** `Device(config)# vlan filter map 1 vlan-list 20-22`	在一个或多个 VLAN ID 上应用 VLAN map。 *list* 可以是一个 VLAN ID（22）、一个连续的列表（10 - 22），或者多个 VLAN ID（12, 22, 30）。逗号和连字符前后的空格是可选的
步骤 4	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device# show running-config`	检查用户输入的信息
步骤 6	**copy running-config startup-config**	（可选）把输入的命令保存到配置文件中

	示例： Device# **copy running-config startup-config**	

监控 IPv6 ACL

用户可以使用下面表格中展示的一条或多条特权 EXEC 命令，来查看所有配置的访问列表、IPv6 访问列表，或指定的访问列表。

命令	目的
show access-lists	显示交换机上配置的所有访问列表
show ipv6 access-lists [*access-list-name*]	显示所有配置的 IPv6 访问列表，或使用名称指定访问列表
show vlan access-map[*map-name*]	显示 VLAN 访问 map 的配置
showvlan filter[**access-map***access-map*|**vlan***vlan-id*]	显示 VACL 和 VLAN 之间的映射关系

以下示例展示了特权 EXEC 命令 **show access-list** 的输入信息。输出信息中会包含交换机或交换机堆栈上配置的所有访问列表。

```
Switch # show access-lists
Extended IP access list hello
10 permit ip any any
IPv6 access list ipv6
permit ipv6 any any sequence 10
```

以下示例展示了特权 EXEC 命令 **show ipv6 access-list** 的输出信息。输出信息中只包含交换机或交换机堆栈上配置的 IPv6 访问列表。

```
Switch# show ipv6 access-list
IPv6 access list inbound
permit tcp any any eq bgp (8 matches) sequence 10
permit tcp any any eq telnet (15 matches) sequence 20
permit udp any any sequence 30
IPv6 access list outbound
deny udp any any sequence 10
deny tcp any any eq telnet sequence 20
```

以下示例展示了特权 EXEC 命令 **show vlan access-map** 的输出信息。输出信息中展示了 VLAN 访问 map 的信息。

```
Switch# show vlan access-map
Vlan access-map "m1" 10
Match clauses:
ipv6 address: ip2
Action: drop
```

其他参考资料

相关主题

相关主题	文档名称
IPv6 安全配置主题	IPv6 Configuration Guide, Inspur INOS XE Release 3SE (Inspur 6850 Switches) http://www.icntnetworks.com
IPv6 命令参考	IPv6 Command Reference, Inspur INOS XE Release 3SE (Inspur 6850 Switches) http://www.icntnetworks.com

错误消息解码器

描述	链接
为了帮助用户查找并解决于这个版本相关的系统错误消息，用户可以使用错误消息解码器（Error Message Decoder）工具	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持网页中提供了大量在线资源，其中的文档和工具能够用来帮助用户排查和解决与 Inspur 产品和技术相关技术问题。 要想收到与用户自己产品相关的安全和技术	http://www.icntnetworks.com

信息，用户可以订阅多种服务，比如产品告警工具（Product Alert Tool；从 Field Notices 中进行访问）、Inspur 技术服务时事（Technical Services Newsletter）和简易信息聚合（RSS）消息。 在 Inspur 支持网页上访问大多数工具都需要用户在 icntnetworks.com 上注册用户 ID 和密码。	

配置 DHCP

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于 DHCP 的信息

DHCP 服务器

DHCP 服务器从交换机或路由器上指定的地址池中分配 IP 地址给 DHCP 客户端，并管理这些地址。如果 DHCP 服务器自己的数据库不能给 DHCP 客户端提供请求的配置参数，该服务器会把请求转发给网络管理员定义的一个或多个次级 DHCP 服务器。交换机可以被配置为 DHCP 服务器。

DHCP 中继代理

DHCP 中继代理是一个在 DHCP 客户端和服务器之间转发 DHCP 包的三层设备。当客户端和服务器不在相同的物理子网上时，中继代理需要在其间转发请求和响应。在正常的二层转发过程中，IP 数据包会透明地在网络之间进行交换，而中继代理的转发行为与正常的二层转发不同。中继代理接收到 DHCP 信息后，会生成新的 DHCP 消息并在出接口上发送。

DHCP 侦听

DHCP 侦听是一项提供网络安全性的 DHCP 安全特性，其过滤不可信的 DHCP 消息，构建并维护一个 DHCP 侦听绑定数据库，也称为 DHCP 侦听绑定表。

DHCP 侦听是不可信主机和 DHCP 服务器之间的防火墙。管理员可以使用 DHCP 侦听特性来区分连接到终端用户的不可信接口，以及连接到 DHCP 服务器或其他交换机的可信接口。

注释： 为使 DHCP 侦听特性正常工作，所有 DHCP 服务器必须通过可信接口连接到交换机。不可信的 DHCP 消息是通过不可信接口收到的消息。默认情况下，交换机认为所有的接口都是不可信的。所以为了使用 DHCP 侦听，用户必须配置交换机的一些接口为可信接口。在服务提供商环境中使用 DHCP 侦听特性时，不可信消息是从服务提供商网络外的设备发来的，比如客户的交换机。来自未知设备的消息是不可信的，因为这些设备可能是流量攻击的源点。DHCP 侦听绑定数据库中包含 MAC 地址、IP 地址、租用时间、绑定类型、VLAN 编号以及对应于一个交换机本地不可信接口的接口信息。数据库中没有与可信接口互连的主机的信息。在服务提供商网络中，可能配置为可信接口的一个例子是与相同网络中主机端口相连的接口。不可信接口的例子是与网络中不可信接口相连的接口，或是与网络之外的设备相连的接口。

当交换机在一个不可信接口上收到一个包，且接口属于启用了 DHCP 侦听的 VLAN 时，交换机会对比源 MAC 地址以及 DHCP 客户端的硬件地址。如果两地址相同（默认情况），交换机会转发此包。如果两地址不同，交换机会丢弃此包。

交换机在以下情况之一发生时，会丢弃 DHCP 包：

- 从网络或防火墙外部收到了来自 DHCP 服务器的包，类型如 DHCPOFFER、DHCPACK、DHCPNAK 或 DHCPLEASEQUERY；
- 在不可信接口上接收到数据包，且源 MAC 地址与 DHCP 客户端的硬件地址不相同；
- 交换机接收到了 MAC 地址在 DHCP 侦听绑定数据库中的 DHCPRELEASE 或 DHCPDECLINE 广播消息，但绑定数据库中的接口信息与接收消息的接口不相同；
- DHCP 中继代理转发了中继代理 IP 地址不为 0.0.0.0 的 DHCP 包，或者中继代理将一个包含可选 82 信息的包转发给了不可信端口。

如果交换机是一台支持 DHCP 侦听的汇聚层交换机，且连接到了一台插入可选 82 信息的边界交换机，该交换机将丢弃从不可信接口接收到的带有可选 82 信息的包。如果启用了 DHCP 侦听且在可信端口上收到了包，该汇聚层交换机不会学习连接设备的 DHCP 侦听绑定信息，且不能构建完整的 DHCP 侦听绑定数据库。

当一台汇聚层交换机可以通过一个不可信接口连接到一台边界交换机，且用户输入了 **ip dhcp snooping information option allow-untrusted** 全局配置命令时，该汇聚层交换机会接受来自边界交换机的带有可选 82 信息的包。该汇聚层交换机将会学习通过不可信交换机接口连接的主机的绑定信息。当交换机通过主机连接的不可信入接口收到带有可选 82 信息的包时，在该汇聚层交换机上仍然可以启用 DHCP 安全特性，如动态 ARP 监测或 IP 源防护。连接到汇聚层交换机的边界交换机的端口必须被配置为可信接口。

插入可选 82 数据

在住宅及城域以太网接入环境中，DHCP 可以中心化地管理大量租户的 IP 地址分配。在交换机上启用 DHCP 可选 82 特性时，租户的设备（除 MAC 地址外）由将其连接至网络的交换机端口标识。租户 LAN 中的多台主机可以连接到接入交换机的相同端口上，且被唯一标识。

注释： 只有在使用可选 82 的租户设备分配的 VLAN 上全局启用 DHCP 侦听特性时，才支持使用 DHCP 可选 82 特性。

下面展示了一个城域以太网网络，其中有一个中心化的 DHCP 服务器给连接到接入层交换机的租户分配 IP 地址。因为 DHCP 客户端及相关联的 DHCP 服务器不在相同的 IP 网络或子网中，所以给一个 DHCP 中继代理（Inspur 交换机）配置了 helper 地址，使其可以转发广播包并在客户端和服务器之间传输 DHCP 消息。

图 113：城域以太网络中的 DHCP 中继代理

DHCP server	DHCP 服务器
Catalystswitch（DHCP relay agent）	Catalyst 交换机（DHCP 中继代理）
Access layer	接入层
Host A (DHCP client)	主机 A（DHCP 客户端）
Subscribers	租户
Host B (DHCP client)	主机 B（DHCP 客户端）
VLAN 10	VLAN 10

在交换机上启用 DHCP 侦听信息可选 82 时，将会发生以下一系列事件：

- 主机（DHCP 客户端）生成 DHCP 请求并广播到网络中；
- 当交换机收到 DHCP 请求时，它向包中添加可选 82 信息。默认情况下，远程 ID 子选项是交换机的 MAC 地址，电路 ID 子选项是端口标识符 **vlan-mod-port**，即接收包的端口。可以配置远程 ID 及电路 ID；
- 如果配置了中继代理的 IP 地址，交换机把此 IP 地址添加到 DHCP 包中；
- 交换机将包含可选 82 字段的 DHCP 请求转发给 DHCP 服务器；
- DHCP 服务器收到包。如果服务器可以处理可选 82 信息，它会使用远程 ID 或电路 ID 来分配 IP 地址并实施策略，比如限制可以分配给一个远程 ID 或电路 ID 的 IP 地址数量。随后 DHCP 服务器在 DHCP 应答中回复该可选 82 字段；
- 如果请求是通过交换机中继给服务器的，DHCP 服务器会把应答单播发给交换机。交换机通过检查远程 ID 或电路 ID 字段，证实该字段是自己插入的。交换机会移除可选 82 字段，并把包转发给连接发送 DHCP 请求的 DHCP 客户端的交换机端口。

在默认的子选项配置中，当上述的一系列事件发生时，这些字段中的值不会改变（见图示 *子选项包格式*）：

- 电路 ID 子选项字段
  - 子选项类型
  - 子选项类型长度
  - 电路 ID 类型
  - 电路 ID 类型长度
- 远程 ID 子选项自选
  - 子选项类型
  - 子选项类型长度

- 远程 ID 类型
- 远程 ID 类型长度

在电路 ID 子选项的端口字段中，端口编号从 3 开始。比如，在一台有 24 个 10/100/1000 端口和四个小型可插拔（small form-factor pluggable，SFP）模块插槽的交换机上，端口 3 是吉比特以太网 1/0/1 端口，端口 4 是吉比特以太网 1/0/2 端口，以此类推。端口 27 是 SFP 模块插槽吉比特以太网 1/0/25 端口，以此类推。

图示*子选项包格式*展示了使用默认子选项配置时的远程 ID 子选项和电路 ID 子选项。对于电路 ID 子选项，模块编号对应于堆栈中的交换机编号。在全局启用 DHCP 侦听并输入 ip dhcp snooping information option 全局配置命令时，交换机使用此包格式。

图 114：子选项包格式

Circuit ID Suboption Frame Format	电路 ID 子选项帧格式
Suboption type	子选项类型
Length	长度
Circuit ID type	电路 ID 类型
Module	模块
Port	端口
1 byte	1 字节
Remote ID Suboption Frame Format	远程 ID 子选项帧格式
Remote ID type	远程 ID 类型
MAC address	MAC 地址

图示用户配置的子选项包格式展示了用户配置的远程 ID 和电路 ID 子选项的包格式。用户在全局启用 DHCP 侦听，且输入全局配置命令 **ip dhcp snooping information option format remote-id** 以及接口配置命令 **ip dhcp snooping vlan information option format-type circuit-id string** 时，交换机使用这些包格式。

用户在配置远程 ID 和电路 ID 子选项时，包中这些字段的值会从默认值变为配置值：

- 电路 ID 子选项字段
  - 电路 ID 类型为 1；
  - 长度值可变，取决于配置的字符串长度。
- 远程 ID 子选项字段
  - 远程 ID 类型为 1；
  - 长度值可变，取决于配置的字符串长度。

图 115：用户配置的子选项包格式

Circuit ID Suboption Frame Format(for user-configured string)	电路 ID 子选项帧格式（用户定义字符串）
Suboption type	子选项类型
Length	长度
Circuit ID type	电路 ID 类型
ASCII Circuit ID string	ASCII 电路 ID 字符串
1 byte	1 字节
Remote ID Suboption Frame Format(for user-configured string)	远程 ID 子选项帧格式（用户定义字符串）
Remote ID type	远程 ID 类型
MAC address	MAC 地址
ASCII Remote ID string or hostname	ASCII 远程 ID 字符换或 hostname

Inspur INOS DHCP 服务器数据库

在基于 DHCP 的自动配置过程中，指定的 DHCP 服务器会使用 Inspur INOS DHCP 服务器数据库。其中有 IP 地址、地址绑定以及配置的参数，如启动文件。

地址绑定是 Inspur INOS DHCP 服务器数据库中 IP 地址和 MAC 地址的一个映射。管理员可以手动指定客户端 IP 地址，DHCP 服务器也可以从 DHCP 地址池中分配 IP 地址。更多有关手动及自动进行地址绑定的信息，参见 *Inspur INOS IP 配置指南 12.4 版*的“配置 DHCP”章节。

有关启用并配置 Inspur INOS DHCP 服务器数据库过程的信息，参见 *Inspur INOS IP 配置指南 12.4 版*的“配置 DHCP”章节中的“DHCP 配置任务列表”一节。

DHCP 侦听绑定数据库

启用 DHCP 侦听时，交换机使用 DHCP 侦听绑定数据库存储不可信接口的相关信息。数据库可存储至多 64000 个绑定条目。

每个数据库条目（绑定）都包含一个 IP 地址，一个关联的 MAC 地址，租用时间（十六进制格式），绑定适用的接口以及接口所属的 VLAN 信息。数据库代理将绑定信息以文件的形式存储在配置的位置。每个条目结尾都有一个校验和，负责对从文件开始的所有与条目相关的字节进行校验。每个条目为 72 字节，后接一个空格，然后是校验和值。

为了在交换机重启时保留绑定信息，管理员必须使用 DHCP 侦听数据库代理。如果代理被禁用，动态 ARP 监测或 IP 源防护会被启用，且 DHCP 侦听绑定数据库有动态绑定条目，则交换机会失去连通性。如果代理被禁用且只启用了 DHCP 侦听，交换机不会使用连通性，但 DHCP 侦听可能无法阻住 DHCP 伪造攻击。

重启时，交换机会读取绑定文件以构建 DHCP 侦听绑定数据库。数据库变化时交换机会更新此文件。

当交换机学习到新的绑定信息或者丢失绑定时，它会立即更新数据库中的条目。交换机也会更新绑定文件中的条目。更新文件的频率是基于可配置的时延的，而且更新批量进行。如果文在在特定的时间内没有被更新（由写时延以及终止超时时延设置），更新停止。

以下是绑定文件的格式：

```
<initial-checksum>
TYPE DHCP-SNOOPING
VERSION 1
BEGIN
<entry-1><checksum-1>
<entry-2><checksum-1-2>
...
...
<entry-n><checksum-1-2-..-n>
END
```

文件中的每个条目都标记有一个校验和值，交换机在读取文件时用此值验证条目。第一行的初始校验和条目区分了与最新的文件更新相关的条目和与上一个文件更新相关的条目。

绑定文件的示例如下：

```
2bb4c2a1
TYPE DHCP-SNOOPING
VERSION 1
BEGIN
192.1.168.1 3 0003.47d8.c91f 2BB6488E Gi1/0/4 21ae5fbb
192.1.168.3 3 0003.44d6.c52f 2BB648EB Gi1/0/4 1bdb223f
192.1.168.2 3 0003.47d9.c8f1 2BB648AB Gi1/0/4 584a38f0
END
```

当交换机启动且计算出的校验和值与存储的校验和值相同时，交换机从绑定文件中读取条目并将其添加到自己的 DHCP 侦听绑定数据库中。当以下情况之一发生时，交换机忽略条目：

- 交换机读取了条目，且计算出的校验和值与存储的校验和值不同。该条目及其后的条目会被忽略。
- 一个条目有过期的租用时间（交换机可能不会在租用时间过期时移除绑定）。
- 系统中不再有条目中的接口。
- 接口是被路由接口或 DHCP 侦听可信接口。

DHCP 侦听及交换机堆栈

DHCP 侦听在堆栈主用设备上进行管理。新的交换机加入堆栈时，该交换机从堆栈主用设备接收 DHCP 侦听配置。当成员离开堆栈时，所有与该交换机关联的 DHCP 侦听地址绑定都会超时。

所有的侦听统计信息都在堆栈主用设备上产生。如果选出了新的堆栈主用设备，统计计数器会被重置。

当堆栈合并发生时，如果堆栈主用设备不再是新堆栈的主用设备，其上的所有 DHCP 侦听绑定都会被丢弃。对于使用堆栈分区的情况，现有堆栈主用设备不变，属于分区交换机的绑定都会超时。分区堆栈的新主用设备开始处理新进的 DHCP 包。

如何配置 DHCP 特性

默认的 DHCP 侦听配置

表 140：默认的 DHCP 配置

特性	**默认设置**
DHCP 服务器	在 Inspur INOS 软件中启用，需要配置 [13]
DHCP 中继代理	启用 [14]
DHCP 包转发地址	无配置
检查中继代理信息	启用（非法信息被丢弃）
DHCP 中继代理转发策略	替代现有的中继代理信息
全局启用 DHCP 侦听	禁用
DHCP 侦听信息选项	启用
接受不可信入端口包的 DHCP 侦听选项 [15]	禁用

DHCP 侦听限速	无配置
DHCP 侦听可信	不可信
DHCP 侦听 VLAN	禁用
DHCP 侦听 MAC 地址认证	启用
Inspur INOS DHCP 服务器绑定数据库	在 Inspur INOS 软件中启用，需要配置。 **注释：**交换只通过配置为 DHCP 服务器的设备获取网络地址及配置参数
DHCP 侦听绑定数据库代理	在 Inspur INOS 软件中启用，需要配置。此特性仅在配置了目的时可用

[13] 交换机仅被配置为 DHCP 服务器时才会响应 DHCP 请求。

[14] 交换机仅在 DHCP 客户端的 SVI 中配置了 DHCP 服务器的 IP 地址时才会中继 DHCP 包。

[15] 当交换机是汇聚层交换机且从边界交换机接收带有可选 82 信息的包时，使用此特性。

DHCP 侦听配置指南

如果一个交换机端口连接到 DHCP 服务器，输入配置命令 **ip dhcpsnooping trust interface** 配置该端口为可信。

如果一个交换机端口连接到 DHCP 客户端，用户需要输入接口配置命令 **no ip dhcpsnooping trust** 把该端口配置为不可信。

管理员可以输入用户 EXEC 命令 **show ip dhcp snooping statistics** 显示 DHCP 侦听统计信息，也可以输入 **clear ip dhcp snoopingstatistics** 特权 EXEC 命令清除侦听统计计数器。

配置 DHCP 服务器

交换机可以作为 DHCP 服务器使用。

关于配置交换机作为 DHCP 服务器使用的过程，参见 *Inspur INOS IP 配置指南 12.4 版*的“IP 编址及服务”章节中的“配置 DHCP”一节。

DHCP 服务器及交换机堆栈

DHCP 绑定数据库由堆栈主用设备管理。指定新的堆栈主用设备时，新的主用设备通过 TFTP 服务器下载存储的绑定数据库。如果堆栈主用设备故障，所有未保存的绑定都会丢失。与丢失的绑定相关的 IP 地址会被释放。管理员应用使用全局配置命令 **ip dhcp database** *url* [**timeout**

seconds | **write-delay** *seconds*]配置自动备份。

堆栈合并发生时，变为堆栈成员设备的堆栈主用设备丢失所有的 DHCP 租用绑定。当堆栈分区发生时，分区中的新主用设备会成为新的 DHCP 服务器，不含有任何现有的 DHCP 租用绑定。

配置 DHCP 终极代理

按照以下步骤在交换机上启用 DHCP 中继代理：

总步骤

1. enable

2. configure terminal

3. service dhcp

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式
步骤 3	**service dhcp** **示例：** `Device(config)#` **service dhcp**	在交换机上启用 DHCP 服务器器及中继代理。此特性默认被启用
步骤 4	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式
步骤 5	**show running-config** **示例：** `Device#` **show running-config**	验证配置的条目

步骤 6	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中

接下来做什么？

查看 *Inspur INOS IP 配置指南 12.4 版*的“IP 编址及服务”章节“配置 DHCP”部分的以下步骤：

- 检查（验证）中继代理信息
- 配置中继代理转发策略

指定包转发地址

如果 DHCP 服务器和 DHCP 客户端不再相同的网络或子网上，管理员必须使用 **ip helper-address** *address* 接口配置命令配置交换机。一般规则是在接近客户端的三层接口上配置该命令。命令 **ip helper-address** 中使用的地址可以是特定的 DHCP 服务器 IP 地址，也可以是其他 DHCP 服务器所在目的网段的网络地址。使用网络地址允许任意 DHCP 服务器响应请求。

从特权 EXEC 模式开始，按照以下步骤指定包转发地址：

总步骤

1. enable

2. configure terminal

3. interface vlan *vlan-id*

4. ip address *ip-address subnet-mask*

5. ip helper-address *address*

6. end

7. 使用以下命令之一:

• interface range *port-range*

• interface *interface-id*

8. switchport mode access

9. switchport access vlan *vlan-id*

10. end

11. show running-config

12. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>enable`	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式
步骤 3	**interface vlan** *vlan-id* **示例：** `Device(config)# interface vlan 1`	输入 VLAN ID 创建交换机虚接口，并进入接口配置模式

步骤 4	**ip address** *ip-address subnet-mask*
	示例：
	`Device(config-if)# ip address`
	`192.108.1.27`
	`255.255.255.0`
		给接口配置 IP 地址和 IP 子网

步骤 5	**ip helper-address** *address*
	示例：
	`Device(config-if)# ip helper-address`
	`172.16.1.2`
		指定 DHCP 包的转发地址。
		helper 地址可以是一个特定的 DHCP 服务器地址，也可以是其他 DHCP 服务器所在目的网段的网络地址。使用网络地址允许其他服务器应答 DHCP 请求。
		如果有多个服务器，可以为每个服务器配置一个 helper 地址

步骤 6	**end** **示例：** `Device(config-if)# end`	返回全局配置模式

步骤 7	使用以下命令之一:
	• **interface range** *port-range*
	• **interface** *interface-id*
	示例：
	`Device(config)# interface`
	`gigabitethernet1/0/2`
		配置多个连接到 DHCP 客户端的物理端口，并进入端口范围配置模式。
		或
		配置一个连接到 DHCP 客户端的物理端口，并进入接口配置模式

步骤 8	**switchport mode access** **示例：** `Device(config-if)# switchport mode access`	为端口定义 VLAN 成员模式
步骤 9	**switchport access vlan** *vlan-id* **示例：** `Device(config-if)# switchport access vlan 1`	指定端口到第 3 步配置的 VLAN 中
步骤 10	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式
步骤 11	**show running-config** **示例：** `Device# show running-config`	验证配置的条目
步骤 12	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把配置保存在配置文件中

配置 DHCP 侦听及可选 82 的前提

配置 DHCP 侦听及可选 82 特性的前提如下：

- 管理员必须在交换机上全局启用 DHCP 侦听；
- 全局启用 DHCP 侦听之前，确保配置了作为 DHCP 服务器工作的设备，且 DHCP 中继代理配置并启用；
- 如果希望交换机响应 DHCP 请求，必须配置交换机作为 DHCP 服务器；
- 在交换机上配置 DHCP 侦听信息选项之前，确保配置了作为 DHCP 服务器工作的设备。管理员必须指定该 DHCP 服务器可以分配或排除的 IP 地址，或者必须为这些设备配置 DHCP 选项；
- 为了让 DHCP 侦听能正常工作，所有的 DHCP 服务器必须通过可信接口连接到交换机。在服务提供商网络中，可信接口是连接到位于相同网络的设备端口上的接口；
- 为使用 DHCP 侦听特性，必须配置交换机使用 Inspur INOS DHCP 服务器绑定数据库；
- 为使用 DHCP 侦听选项接受在不可信入端口上收到的包，交换机必须是汇聚层交换机，且从边界交换机上接收带有可选 82 信息的包；

- DHCP 侦听绑定数据库配置的前提如下：
  - 为使用 DHCP 侦听特性，必须在 DHCP 侦听绑定数据库中配置目的；
  - 因为 NVRAM 和闪存的存储空间都很有限，建议将绑定文件存储在 TFTP 服务器上；
  - 对于使用基于网络的 URL 来说（如 TFTP 以及 FTP），管理员必须为配置的 URL 创建一个空文件，以使交换机可以写入绑定信息到该 URL 的绑定文件中。请查看使用的 TFTP 服务器的文档，确定是否必须现在服务器上创建空文件；一些 TFTP 服务器无法这样配置；
  - 为了确保数据库中的租用时间是准确的，建议管理员启用并配置网络时间协议（Network Time Protocol，NTP）；
  - 如果配置了 NTP，交换机只会在交换机的系统时间与 NTP 同步后才会将绑定更新写入到绑定文件中。
- 在交换机上配置 DHCP 中继代理之前，确保配置了工作为 DHCP 服务器的设备。管理员必须指定该 DHCP 服务器可以分配或排除的 IP 地址，配置该设备的 DHCP 选项，或设置 DHCP 数据库代理；
- 如果希望交换机中继 DHCP 包，必须在 DHCP 客户端的交换机虚接口（SVI）上必须配置 DHCP 服务器的 IP 地址；
- 如果交换机端口连接到DHCP服务器，需输入配置命令ip dhcpsnooping trust interface将端口配置为可信；
- 如果端口连接到DHCP客户端，需使用接口配置命令no ip dhcpsnooping trust将端口配置为不可信。

启用 Inspur INOS DHCP 服务器数据库

有关启用及配置 Inspur INOS DHCP 服务器数据库的过程，参见 *Inspur INOS IP 配置指南 12.4 版*的“配置 DHCP”章节的“DHCP 配置任务列表”部分。

监控 DHCP 侦听信息

表 141：显示 DHCP 信息的命令

命令	**描述**
show ip dhcp snooping	显示交换机的 DHCP 侦听配置。
show ip dhcp snooping binding	只显示 DHCP 侦听绑定数据库（绑定表）中

	动态配置的绑定信息。
show ip dhcp snooping database	显示 DHCP 侦听绑定数据库的状态及统计信息。
show ip dhcp snooping statistics	显示 DHCP 侦听统计信息的汇总详情。
show ip source binding	显示动态及静态配置的绑定信息。

注释： 如果启用了 DHCP 侦听，且接口变为 down 状态，交换机不会删除静态配置的绑定条目。

配置 DHCP 服务器进行基于端口的地址分配

配置 DHCP 服务器进行基于端口地址分配的相关信息

DHCP 服务器的基于端口地址分配特性允许 DHCP 在一个以太网交换机端口上保持使用相同的 IP 地址，无论连接的设备客户端标识符或客户端硬件地址如何变化。

在网络中部署使用以太网交换机时，它们为直连设备提供连通性。在比如工厂车间这样的环境中，如果设备发生故障，替换的设备必须能立刻在现有网络中工作。当前的 DHCP 部署方式无法保证 DHCP 能给替换的设备提供相同的 IP 地址。控制、监控及其他软件希望每台设备能有关联的稳定的 IP 地址。如果设备被替换，即便 DHCP 客户端改变了，地址分配过程也应保持稳定。

配置了 DHCP 服务器的基于端口的地址分配特性后，就能确保给相同的连接端口分配相同的 IP 地址，即使从该端口收到的 DHCP 消息中客户端标识符或客户端硬件地址发生了改变。DHCP 协议通过 DHCP 包中的客户端标识符选项识别 DHCP 客户端。不包含客户端标识符选项的客户端通过其硬件地址进行标识。配置了此特性后，接口的名称覆盖客户端标识符或硬件地址，而交换机端口这个实际的连接点成为了客户端标识符。

所有情况中，通过以太网线缆连接到相同端口的设备都能通过 DHCP 获取相同的 IP 地址。

DHCP 服务器的基于端口的地址分配特性只在 Inspur INOS DHCP 服务器上支持，第三方服务器不支持此特性。

默认的基于端口地址分配配置

默认情况下，DHCP 服务器的基于端口地址分配特性被禁用。

基于端口的地址分配配置指南

- 默认情况下，DHCP 服务器的基于端口地址分配特性被禁用；
- 为把地址的分配从 DHCP 地址池限制到预配置的预留地址中（非预留地址不会提供给客户端，其他客户端不由该地址池服务），管理员可以输入 **reserved-only** DHCP 地址池配置命令。

启用 DHCP 侦听绑定数据库代理

总步骤

1. enable

2. configure terminal

3. ip dhcp snooping database {**flash**[*number*]**:**/*filename* | **ftp://***user***:***password***@***host***/***filename* | **http://**[[*username***:***password*]**@**]{*hostname* | *host-ip*}[/*directory*] **/***image-name***.tar** | **rcp://***user***@***host***/***filename*}| **tftp://***host***/***filename*

4. ip dhcp snooping database timeout *seconds*

5. ip dhcp snooping database write-delay *seconds*

6. end

7. ip dhcp snooping binding *mac-address* **vlan** *vlan-id ip-address* **interface** *interface-id* **expiry** *seconds*

8. show ip dhcp snooping database [**detail**]

9. show running-config

10. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal**	进入全局配置模式

	示例： `Device#` **configure terminal**	
步骤 3	**ip dhcp snooping database** {**flash**[*number*]**:**/*filename* | **ftp://***user*:*password*@*host*/*filename* | **http://**[[*username*:*password*]**@**]{*hostname* | *host-ip*}[/*directory*] **/***image-name***.tar** | **rcp://***user*@*host*/*filename*}| **tftp://***host*/*filename* **示例：** `Device(config)#` **ip dhcp snooping database tftp://10.90.90.90/snooping-rp2**	使用以下命令之一指定数据库代理或绑定文件的 URL： • **flash**[*number*]**:/***filename* （可选）使用 *number* 参数指定堆栈主用设备的堆栈成员编号。*number* 的范围从 1 到 9。 • **ftp://***user*:*password*@*host*/*filename* • **http://**[[*username*:*password*]**@**]{*hostname* | *host-ip*}[/*directory*] **/***image-name***.tar** • **rcp://***user*@*host*/*filename* • **tftp://***host*/*filename*
步骤 4	**ip dhcp snooping database timeout** *seconds* **示例：** `Device(config)#` **ip dhcp snooping database timeout 300**	指定在停止数据库传输过程前等待多久（单位秒）。 默认值是 300 秒，范围从 0 到 86400。使用 0 指定无限期间，表示无限期的尝试传输
步骤 5	**ip dhcp snooping database write-delay** *seconds* **示例：** `Device(config)#` **ip dhcp snooping database write-delay 15**	指定绑定数据库变化后应该延迟传输的时间长度。范围从 15 到 86400 秒，默认值是 300 秒（5 分钟）
步骤 6	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式
步骤 7	**ip dhcp snooping binding** *mac-address* **vlan** *vlan-id* *ip-address* **interface** *interface-id* **expiry** *seconds*	（可选）向 DHCP 侦听绑定数据库添加绑定条目。*vlan-id* 的范围从 1 到 4904。*seconds* 的范围从 1 到 4294967295。 添加每个条目都要输入此命令。

	示例： `Device#` **ip dhcp snooping binding 0001.1234.1234 vlan 1 172.20.50.5 interface gi1/1 expiry 1000**	测试或调试交换机时使用此命令
步骤 8	**show ip dhcp snooping database** [**detail**] **示例：** `Device# show ip dhcp snooping database detail`	显示 DHCP 侦听绑定数据库代理的状态和统计信息
步骤 9	**show running-config** **示例：** `Device#` **show running-config**	验证配置的条目
步 骤 10	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中

启动 DHCP 服务器的基于端口地址分配特性

按照以下步骤全局启用基于端口的地址分配特性，并在接口上自动生成租户标识符。

总步骤

1. enable

2. configure terminal

3. ip dhcp use subscriber-id client-id

4. ip dhcp subscriber-id interface-name

5. interface *interface-id*

6. ip dhcp server use subscriber-id client-id

7. end

8. show running-config

9. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：**	进入特权 EXEC 模式。在提示时输入密码

	Device>**enable**	
步骤 2	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式
步骤 3	**ip dhcp use subscriber-id client-id** **示例：** Device(config)# **ip dhcp use subscriber-id client-id**	配置 DHCP 服务器，将租户标识符作为客户端标识符全局应用在所有进入的 DHCP 消息上
步骤 4	**ip dhcp subscriber-id interface-name** **示例：** Device(config)# **ip dhcp subscriber-id interface-name**	基于接口的短名称自动生成租户标识符。 特定接口上的租户标识符配置优先于此命令
步骤 5	**interface** *interface-id* **示例：** Device(config)# **interface gigabitethernet1/0/1**	指定需要配置的接口，进入接口配置模式
步骤 6	**ip dhcp server use subscriber-id client-id** **示例：** Device(config-if)# **ip dhcp server use subscriber-id client-id**	配置 DHCP 服务器使用租户标识符作从该接口进入的所有 DHCP 消息的客户端标识符
步骤 7	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 8	**show running-config** **示例：** Device# **show running-config**	验证配置的条
步骤 9	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把配置保存在配置文件中

接下来做什么？

在交换机上启用 DHCP 基于端口的地址分配之后，使用 **ip dhcp pool** 全局配置命令预分配 IP 地址并将其与客户端关联。

监控 DHCP 服务器基于端口的地址分配

表 142：显示 DHCP 基于端口的地址分配信息的命令

命令	目的
show interface *interface id*	显示特定接口的状态和配置
show ip dhcp pool	显示 DHCP 地址池
show ip dhcp binding	显示 Inspur INOS DHCP 服务器上的地址绑定信息

其他参考资源

相关文档

相关主题	文档题目
DHCP 配置信息及过程	IP 编址：DHCP 配置指南，Inspur INOS XE 3S 版 http://www.icntnetworks.com

错误信息解释

描述	链接
为帮助管理员搜索并解决该版本中的系统错误信息，管理员可使用错误信息解释工具。	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。	http://www.icntnetworks.com

访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	

配置 IP 源防护

IP 源防护（IP Source Guard，IPSG）是一项在非路由的二层接口上限制 IP 流量的安全特性。该特性基于 DHCP 侦听绑定数据库和手动配置的 IP 源绑定信息进行流量过滤。

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

有关 IP 源防护的信息

IP 源防护

如果一台主机尝试使用邻居的 IP 地址，可以使用 IP 源防护来避免流量攻击。当 DHCP 侦听在不可信接口上启用时也可以启用 IP 源防护。

在接口上启用 IPSG 之后，交换机会阻隔除了 DHCP 侦听允许的 DHCP 包之外的所有在接口上

收到的 IP 流量。

交换机使用硬件的源 IP 查找表来绑定 IP 地址到端口。对于 IP 和 MAC 过滤，交换机会组合进行源 IP 以及源 MAC 的查找。源 IP 地址在绑定表中 IP 流量会被允许，而所有其他流量会被拒绝。

IP 源绑定表中的绑定条目是通过 DHCP 侦听学习到的或者是手工配置的（静态 IP 源绑定）。表中的一个条目包含 IP 地址，关联的 MAC 地址以及关联 VLAN 编号。交换机只在启用 IP 源防护的时候使用 IP 源绑定表。

只有包括接入端口和中继端口这样的二层端口才支持 IPSG。可以配置 IPSG 进行源 IP 地址过滤或源 IP 及 MAC 地址过滤。

静态主机的 IP 源防护

注释： 不要对上行链路端口上的静态主机或中继端口使用 IPSG（IP 源防护）。

静态主机的 IP 源防护将 IPSG 的能力扩展到了非 DHCP 的静态环境中。以前的 IPSG 使用 DHCP 侦听创建的条目来验证连接到交换机的主机。任何从主机收到的没有对应合法 DHCP 绑定条目的流量都会被丢弃。这项安全特性限制了非路由二层接口上的 IP 流量。它基于 DHCP 侦听绑定数据库以及手动配置的 IP 源绑定信息过滤流量。以前的 IPSG 版本要求有 DHCP 环境才能工作。

静态主机的 IPSG 允许在不使用 DHCP 的情况下工作。静态主机的 IPSG 依赖 IP 设备追踪表的条目来安装端口 ACL。交换机根据 ARP 请求或者其他的 IP 包来创建条目，维护特定端口的合法主机列表。管理员也可以指定允许给特定端口发送流量的主机数量。这项操作等同于三层的端口安全特性。

静态主机的 IPSG 也支持动态主机。如果一台动态主机接收了一个 DHCP 分配的 IP 地址，且这个地址同时在 IP DHCP 侦听表中可用，IP 设备追踪表也会学习到相同的条目信息。在堆叠环境中，当主用设备故障切换发生时，连接到成员端口的静态主机的 IP 源防护条目将被保留。当管理员输入 EXEC 命令 **show ip device tracking all** 时，IP 设备追踪表会显示这些条目状态的为 ACTIVE。

注释： 一些有多个网络接口的 IP 主机可能会向网络接口发送非法的数据包。非法的数据包会以该主机其他网络接口的 IP 或 MAC 地址作为源。这些非法的数据包可以导致静态主机的 IPSG 连接到该主机，获知非法的 IP 或 MAC 地址绑定信息，并拒绝合法的绑定。请咨询对应操作系统及网路接口的提供商，避免主机发送非法的数据包。

静态主机的 IPSG 开始时通过基于 ACL 的侦听机制动态地学习 IP 或 MAC 绑定。IP 或 MAC 的绑定是通过 ARP 和 IP 包从静态主机上学习来的。这些信息被存储在设备追踪数据库中。当

特定端口上动态学习或者静态配置的 IP 地址数量达到最大值时，交换机硬件会丢弃任何使用新的 IP 地址的包。为了解决主机因故移动或移除的问题，静态主机的 IPSG 会利用 IP 设备追踪功能来对动态获知的 IP 地址绑定信息进行超时处理。这项特性可以与 DHCP 侦听特性一起使用。对于同时连接了 DHCP 主机和静态主机的端口，将会有多个绑定条目被创建。例如，绑定信息会同时存储在设备追踪数据库和 DHCP 侦听绑定数据库中。

IP 源防护配置指南

- 管理员只能在非路由端口上配置静态IP绑定特性。如果在被路由接口上输入了**ip source binding***mac-address***vlan***vlan-id ip-address***interface** *interface-id*全局配置命令，将出现以下错误信息：
  - `Static IP source binding can only be configured on switch port.`
- 在接口上启用过滤源 IP 的 IP 源防护时，必须在该接口的接入 VLAN 上启用 DHCP 侦听；
- 如果在有多个 VLAN 的中继端口上启用了 IP 源防护，且对所有 VLAN 都启用了 DHCP 侦听，源 IP 地址过滤会被应用到所有 VLAN 上；

注释： 如果启用了 IP 源防护且管理员对中继端口上的 VLAN 启用或禁用了 DHCP 侦听，交换机可能无法正常过滤流量。

- 可以在启用 802.1x 基于端口认证特性的同时启用此特性。

如何配置 IP 源防护

启用 IP 源防护

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. ip verify source [**mac-check**]

5. exit

6. ip source binding *mac-address* **vlan***vlan-id ip-address* **interface** *interface-id*

7. end

8. show running-config

9. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet 1/0/1**	指定要配置的接口，进入接口配置模式
步骤 4	**ip verify source** [**mac-check**] **示例:** `Device(config-if)#` **ip verify source**	启用进行源 IP 地址过滤的 IP 源防护。 （可选）**mac-check**——启用进行源 IP 地址过滤机 MAC 地址过滤的 IP 源防护
步骤 5	**exit** **示例:** `Device(config-if)#` **exit**	返回全局配置模式
步骤 6	**ip source binding** *mac-address* **vlan***vlan-id ip-address* **interface** *interface-id* **示例:** `Device(config)#` **ip source binding 0100.0230.0002 vlan 11 10.0.0.4 interface gigabitethernet1/0/1**	添加静态 IP 源绑定条目。 每个静态绑定条目都需输入此命令
步骤 7	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式
步骤 8	**show running-config** **示例：**	验证配置的条目

	`Device#` **show running-config**	
步骤 9	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中

在二层接入端口上配置静态主机的 IP 源防护

为使静态主机的 IP 源防护特性工作，必须全局配置接口配置命令 **ip device tracking maximum***limit-number*。如果只在端口上配置了此命令而没有全局启用 IP 设备追踪，或者对该端口设置了最大的 IP 设备追踪数量，静态主机的 IPSG 会拒绝所有来自该接口的 IP 流量。

总步骤

1. enable

2. configure terminal

3. ip device tracking

4. interface *interface-id*

5. switchport mode access

6. switchport access vlan*vlan-id*

7. ip verify source[**tracking**] [**mac-check**]

8. ip device tracking maximum *number*

9. end

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip device tracking** **示例:** `Device(config)#` **ip device tracking**	开启 IP 主机表，并全局启用 IP 设备追踪
步骤 4	**interface** *interface-id*	进入接口配置模式

	示例: `Device(config)#` **interface gigabitethernet 1/0/1**	
步骤 5	**switchport mode access** **示例:** `Device(config-if)#` **switchport mode access**	将端口配置为 Access 模式
步骤 6	**switchport access vlan***vlan-id* **示例:** `Device(config-if)#`**switchport access vlan 10**	为此端口配置 VLAN
步骤 7	**ip verify source**[**tracking**] [**mac-check**] **示例:** `Device(config-if)#`**ip verify source tracking mac-check**	启用进行源 IP 地址过滤的 IP 源防护。 （可选）为静态主机启用 IP 源防护。 （可选）启用 MAC 地址过滤。 命令 **ip verify source tracking mac-check**enables 启用进行 MAC 地址过滤的静态主机 IP 源防护
步骤 8	**ip device tracking maximum** *number* **示例:** `Device(config-if)#`**ip device tracking maximum 8**	设置 IP 设备追踪表允许端口拥有的最大静态 IP 数量。范围从 1 到 10，最大值是 10。 **注释**必须配置接口配置命令**ip device tracking maximum***limit-number*
步骤 9	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式

监控 IP 源防护

表 143：特权 EXEC show 命令

命令	目的
show ip verify source [**interface** *interface-id*]	显示交换机或者特定接口的 IP 源防护配置

showip device tracking { **all**| **interface** *interface-id* | **ip***ip-address* | **mac** *imac-address*}	显示 IP 设备追踪表中的条目信息

表 144：接口配置命令

命令	目的
ip verify source tracking	验证数据源

有关显示输出字段的详细信息，参见此版本的命令参考手册。

其他参考资料

错误信息解释

描述	**链接**
为帮助管理员搜索并解决该版本中的系统错误信息，管理员可使用错误信息解释工具。	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	http://www.icntnetworks.com

配置动态 ARP 监测

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

动态 ARP 监测的限制

本节列出了在交换机上配置动态 ARP 监测的限制条件和配置指南。

- 动态 ARP 监测是一种入向的安全特性，它不执行出向检查；
- 动态 ARP 监测对于连接到交换机上的不支持动态 ARP 监测或未启用此特性的主机无效。因为中间人攻击被限制在一个二层的广播域，而这个广播域被执行动态 ARP 监测检查的主机和不进行检查的主机分隔开。该特性的操作保护了域中启用动态 ARP 监测主机的 ARP 缓存；
- 动态 ARP 监测依靠 DHCP 侦听绑定数据库中的条目来验证入向 ARP 请求和响应的 IP-MAC 地址映射。确保启用了 DHCP 侦听特性，放行使用动态分配 IP 地址的 ARP 包。

在禁用 DHCP 侦听或非 DHCP 的环境中，使用 ARP ACL 允许或拒绝数据包。

- 动态 ARP 监测支持接入端口、中继端口以及 EtherChannel 端口。

注释： 不要在 RSPAN VLAN 上启用动态 ARP 监测特性。如果在 RSPAN VLAN 上启用了动态 ARP 监测，动态 ARP 监测包可能无法到达 RSPAN 目的端口。

- 只有当物理端口的可信状态与通道端口的可信状态相同时，物理端口才可以加入 EtherChannel 端口通道。否则，物理端口会在端口通道中保持挂起。端口通道从第一个加入通道的物理端口上继承可信状态。因此，第一个物理端口的可信状态无需与通道的可信状态相同。

  相反的，更改端口通道的可信状态时，交换机会在组成通道的所有物理端口上配置新的

可信状态；

- 限速在交换机堆栈中的每个交换机上独立计算。对于跨堆栈的 EtherChannel，实际的限速值可能比配置的值高。例如，对于一个端口在交换机 1 上而另一个端口在交换机 2 上的 EtherChannel，如果管理员设置了限速为 30pps，每个端口在不造成 EtherChannel 错误禁用的情况下可以接收数据包的速度是 29pps；
- 端口通道的运行速率是通道内所有物理端口速率的累加和。例如，如果配置端口通道对 ARP 的限速为 400pps，通道中所有端口总共接收速率为 400pps。EtherChannel 端口的入向 ARP 包速率是所有通道成员的入向包速率的和。请在检查通道端口成员的入向 ARP 包速率之后再配置 EtherChannel 端口的限速速率。

  物理端口上入向包的速率与端口通道的配置对比，不与物理端口的配置进行对比。端口通道的限速配置独立于物理端口的配置。

  如果 EtherChannel 接收了超过配置速率的 ARP 包，信道（包括所有物理端口）会被置为错误禁用状态；
- 确保对入向中继端口上的 ARP 包进行限速。应给中继端口配置较高的速率以反映其聚合性，以便处理多个启用了动态 ARP 监测 VLAN 的数据包。也可以使用接口配置命令 **ip arp inspection limit none** 设置速率为不限制。当系统将端口置为错误禁用状态时，一个 VLAN 上的高限速设置可能会造成对另一个的 VLAN 的拒绝服务攻击；
- 在交换机上启用动态 ARP 监测时，配置用于管理 ARP 流量的策略器将不再生效。启用的结果是所有 ARP 流量都会被送往 CPU。

理解动态 ARP 监测

ARP 通过进行 IP 地址与 MAC 地址的映射在二层广播域间提供 IP 通信。例如，主机 B 希望给主机 A 发送信息，但是它的 ARP 缓存中没有主机 A 的 MAC 地址。主机 B 生成一个发往广播域中所有主机的消息，以获取与主机 A 的 IP 地址关联的 MAC 地址。广播域中的所有主机都会收到此 ARP 请求，而主机 A 使用自己的 MAC 地址来应答。然而，因为 ARP 允许主机在没有收到 ARP 请求的情况下发出无故应答，ARP 伪造攻击和 ARP 缓存的毒化就可以发生。在攻击发生后，被攻击设备的所有流量都会流经攻击者的计算机，然后再发往路由器、交换机或者主机。

恶意用户可以攻击连接到二层网络的主机、交换机和路由器，毒化连接到子网的系统的 ARP 缓存，并截获发往子网上其他主机的流量。图 26-1 展示了 ARP 缓存毒化的示例。

图 116：APR 缓存毒化

Host A	主机 A
Host B	主机 B
Host C(man-in-the-middle)	主机 C（中间人）

主机 A、B 和 C 连接到交换机的接口 A、B 和 C，这些端口在相同的子网中。主机的 IP 和 MAC 在括号中标出，如主机 A 使用 IP 地址 IA 和 MAC 地址 MA。当主机 A 需要与主机 B 在 IP 层通信时，它会广播一个 ARP 消息请求与 IP 地址 IB 关联的 MAC 地址。当交换机和主机 B 收到此 ARP 请求时，它们会填充自己的 ARP 缓存，产生主机 IP 地址为 IA，MAC 地址为 MA 的 ARP 绑定信息，如 IP 地址 IA 被绑定到 MAC 地址 MA 上。当主机 B 应答时，交换机和主机 A 填充其 ARP 缓存，产生主机 IP 地址为 IB，MAC 地址为 MB 的绑定。

主机 C 可以广播伪造的 ARP 应答，将主机 IP 地址 IA（或 IB）与 MAC 地址 MC 地址绑定，进而毒化交换机、主机 A 和主机 B 的 ARP 缓存。ARP 缓存被毒化的主机会使用 MAC 地址 MC 作为发往 IA 或 IB 的流量的目的 MAC 地址。这意味着主机 C 截获了这些流量。因为主机 C 知道与 IA 和 IB 关联的真正 MAC 地址，它可以使用正确的 MAC 地址作为目的把截获的流量转发给这些主机。主机 C 把自己插入在主机 A 到主机 B 的流量之间，这就是*典型的中间人（man-in-the middle）*攻击。

动态 ARP 监测是一项验证网络中 ARP 包的安全特性，它截获、记录并丢弃 IP-MAC 地址绑定非法的 ARP 包。这项特性能保护网络免于特定的中间人攻击。

动态 ARP 监测确保只有合法的 ARP 请求和应答被转发。交换机执行这些行为：

- 截获不可信端口上的所有 ARP 请求和应答
- 在更新本地 ARP 缓存或者把包转发给正确目的之前验证每个截获的数据包有合法的 IP-MAC 地址绑定
- 丢弃非法 ARP 包

动态 ARP 监测根据存储在可信数据库（DHCP 侦听绑定数据库）中的合法 IP-MAC 地址绑定来确定一个 ARP 包的合法性。如果交换机和 VLAN 上启用了 DHCP 侦听，这个数据库由 DHCP 侦听构建。如果在可信接口上收到 ARP 包，交换机不做检查转发此数据包。在不可信接口上，交换机只在数据包合法时才进行转发。

管理员可以使用全局配置命令 **ip arp inspection vlan** *vlan-range* 基于每个 VLAN 启用动态 ARP 监测。

在非 DHCP 环境中，动态 ARP 监测可以对比用户配置的 ARP 访问控制列表（access control lists，ACLs）验证静态配置 IP 地址的主机。管理员可以使用全局配置命令 **arp access-list** *acl-name* 定义 ARP ACL。

可以配置动态 ARP 监测在 ARP 包中的 IP 地址非法或者 ARP 包中的 MAC 地址与以太网报头中的地址不同时就丢弃数据包。使用全局配置命令 **ip arp inspection validate** {**[src-mac]** [**dst-mac]**

[**ip**]}进行配置。

接口可信状态及网络安全性

动态 ARP 监测会把可信状态与交换机上的每个接口关联。可信接口上到达的数据包会绕过所有的动态 ARP 监测验证检查，不可信接口上到达的数据包会经历动态 ARP 监测验证过程。典型的网络配置中，可以把所有连接到主机端口的交换机端口配置为不可信，把所有连接到交换机的交换机端口配置为可信。在此配置中，从特定交换机进入网络的所有 ARP 包会绕过安全检查，在 VLAN 或网络的任何其他地方都无需进行验证。可以使用接口配置命令 **iparp inspection trust** 设置可信状态。

注意： 请小心使用可信状态配置。在接口应该被信任时把它配置成不可信可能会导致丢失连通性。

下图中，假设交换机 A 和交换机 B 都在包含主机 1 和主机 2 的 VLAN 上运行动态 ARP 监测。如果主机 1 和主机 2 都通过连接到交换机 A 的 DHCP 服务器获取 IP 地址，只有交换机 A 会进行主机 1 的 IP-MAC 地址绑定。因此，如果交换机 A 和交换机 B 之间的接口是不可信的，来自主机 1 的 ARP 包会被交换机 B 丢弃。主机 1 和主机 2 之间的连通性会丢失。

图 117：启用动态 ARP 监测 VLAN 上的 ARP 包验证

DHCP server	DHCP 服务器
Switch A	交换机 A
Switch B	交换机 B
Host 1	主机 1
Port 1	端口 1

当接口实际不可信时配置其为可信会在网络中留下安全漏洞。如果交换机 A 不运行动态 ARP 监测，主机 1 可以容易的毒化交换机 B 的 ARP 缓存（如果交换机间的链路配置为可信，也会毒化主机 2 的缓存）。即使交换机 B 运行动态 ARP 监测也可以发生这种情况。

动态 ARP 监测确保连接到交换不可信接口上的主机不能毒化网络中其他主机的 ARP 缓存。然而，对于连接到运行动态 ARP 监测的交换机上的主机，动态 ARP 监测不能防止在网络其他部分的主机毒化的这些主机的缓存。

在 VLAN 里一些交换机运行动态 ARP 监测而一些交换机不运行的情况中，应配置连接到这些交换机的接口为不可信。然而，为了验证来自非动态 ARP 监测交换机的数据包的绑定，可以配置运行动态 APR 监测的交换机使用 ARP ACL。在不能确定这样的绑定信息时，应在三层隔离运行动态 ARP 监测的交换机和不运行的交换机。

注释： 根据 DHCP 服务器以及网络设置的不同，可能无法在 VLAN 中的所有交换机上验证

特定 ARP 包。

ARP 包的限速

交换机的 CPU 执行动态 ARP 监测检查。因此，为了避免拒绝服务攻击，要对入向 ARP 包的数量进行限速。默认情况下，不可信接口的速率是 15 包每秒（packets per second，pps）。可信接口不被限速。可以使用接口配置命令 **ip arp inspection limit** 更改此设置。

当入向 ARP 包的速率超过配置的限制时，交换机会把端口置为错误禁用状态。端口在管理员干预之前保持此状态。可以使用全局配置命令 **errdisable recovery** 启用错误禁用恢复，这样端口就可以在指定的超时周期之后自动摆脱此状态。

注释： 对于 EtherChannel 的限速会独立地应用到堆栈中的每个交换机上。比如，如果在 EtherChannel 上配置了限速 20pps，EtherChannel 中的每个交换机端口可以承载至多 20pps。如果任意交换机超过了限制，整个 EtherChannel 都会被置为错误禁用状态。

ARP ACL 和 DHCP 侦听条目的相对优先级

动态 ARP 监测使用 DHCP 侦听数据库作为合法 IP-MAC 地址的映射表。

ARP ACL 优先于 DHCP 侦听绑定数据库中的条目。交换机仅在使用全局配置命令 **ip arp inspection filter vlan** 配置时才使用 ACL。交换机首先把 ARP 包和用户配置的 ARP ACL 进行比较。如果 ARP ACL 拒绝此 ARP 包，即使 DHCP 侦听填充的数据库中有对应合法的绑定存在，交换机也会拒绝此数据包。

记录丢弃的数据包

当交换机丢弃一个数据包时，交换机会在日志缓存中放置一个条目，然后在控制速率的基础上生成系统消息。在消息生成后，交换机会从日志缓存中清除条目。每个日志条目都包含流信息，如接收 VLAN、端口号、源目 IP 地址以及源目 MAC 地址。

可以使用全局配置命令 **ip arp inspection log-buffer** 配置缓存的条目数量以及特定间隔内生成系统消息所需的条目数量。可以使用全局配置命令 **ip arp inspection vlan logging** 指定记录的数据包类型。

默认的动态 ARP 监测配置

特性	默认设置
动态 ARP 监测	在所有 VLAN 上禁用
接口可信状态	所有接口都是不可信
入向 ARP 包限速	不可信接口的速率是 15pps，假定网络是被交换网络，其中主机每秒连接多达 15 个新主机。 对所有可信接口不限速。 突发间隔是 1 秒
非 DHCP 环境的 ARP ACL	无 ARP ACL 定义
验证检查	不执行检查
日志缓存	启用动态 ARP 检查时，所有拒绝或丢弃的 ARP 包都被记录。 日志的条目数量是 32。 系统消息数量限制为 5 个每秒。 日志速率间隔是 1 秒
基于 VLAN 的日志	所有拒绝或丢弃的 ARP 包都被记录

ARP ACL 和 DHCP 侦听条目的相对优先级

动态 ARP 监测特性会为有效的 IP 到 MAC 地址绑定表，使用 DHCP 侦听（Snooping）绑定数据库。

ARP ACL 的优先级高于 DHCP Snooping 绑定数据库中的条目。只有用户使用全局配置命令 ip arp inspection filter vlan 进行了配置后，交换机才会使用这个 ACL。交换机会首先用 ARP 数据包与用户配置的 ARP ACL 进行比较。如果 ARP ACL 中拒绝 ARP 数据包，交换机也就会拒绝数据包，即使 DHCP Snooping 生成的数据库中存在有效的绑定关系。

为非 DHCP 环境配置 ARP ACL

以下展示了当图 2 中的交换机 B 不支持动态 ARP 监测或 DHCP 侦听时如何配置动态 ARP 监测特性。

如果配置交换机 A 的端口 1 为可信，就产生了一个安全漏洞，因为交换机 A 和主机 1 都可能被交换机 B 或主机 2 攻击。为了避免这种可能性，用户必须把交换机 A 上的端口 1 配置成不可信。要允许来自主机 2 的 ARP 包通过，必须设置 ARP ACL 并把它应用在 VLAN 1 上。如果主机 2 的 IP 地址不是静态的（不可能在交换机 A 上应用 ACL 配置），必须在三层隔离交换机 A 和交换机 B，并使用路由器在它们之间进行路由。

以下是在交换机 A 上配置 ARP ACL 的步骤。此过程应在非 DHCP 环境中执行。

总步骤

1. **enable**
2. **configure terminal**
3. **arp access-list** *acl-name*
4. **permit ip host** *sender-ip* **mac host** *sender-mac*
5. **exit**
6. **ip arp inspection filter** *arp-acl-name* **vlan** *vlan-range* [**static**]
7. **interface** *interface-id*
8. **no ip arp inspection trust**
9. **end**
10. 使用以下show命令:
    - **show arp access-list** acl-name
    - **show ip arp inspection vlan** *vlan-range*
    - **show ip arp inspection interfaces**
11. **show running-config**
12. **copy running-config startup-config**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码

步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式
步骤 3	**arp access-list** *acl-name*	定义一个 ARP ACL，然后进入 ARP 访问列表配置模式。默认情况下，没有定义 ARP 访问列表。 **注释：**在ARP访问列表的末尾，有一个隐含的**deny ip anymac any**命令
步骤 4	**permit ip host** *sender-ip* **mac host** *sender-mac*	允许特定主机（主机 2）的 ARP 包。 • 对于 *sender-ip,*，输入主机 2 的 IP 地址。 • 对于 *sender-mac*，输入主机 2 的 MAC 地址
步骤 5	**exit**	返回全局配置模式。
步骤 6	**ip arp inspection filter** *arp-acl-name* **vlan** *vlan-range* [**static**]	把 ARP ACL 应用到 VLAN 上。默认情况下，没有应用到 VLAN 上的 ARP ACL。 • 对于 *arp-acl-name*，指定第 3 步中创建的 ACL 名称。 • 对于 *vlan-range*，指定交换机以及主机所属的 VLAN。 可以使用 VLAN ID 编号指定一个 VLAN，可以指定由连字符分隔的 VLAN 范围，也可以指定由逗号分隔的一组 VLAN。VLAN 范围从 1 到 4096。 • （可选）指定 **static** 字段，把 ARP ACL 中隐含的拒绝条目当作显式条目对待，丢弃与 ACL 中之前任意条目都不匹配的数据包。DHCP 绑定不被使用。 如果不指定此关键字，意味着 ACL 中没有显式的拒绝数据包的条目

		存在，DHCP 绑定就会在数据包不匹配 ACL 中任意行的时候决定其被允许还是被拒绝。 只包含 IP-MAC 地址映射的 ARP 包与 ACL 进行比较。只有在访问列表允许时这些数据包才被允许转发
步骤 7	**interface** *interface-id*	指定交换机 A 连接到交换机 B 的接口，并进入接口配置模式
步骤 8	**no ip arp inspection trust**	把交换机 A 连接到交换机 B 的接口配置为不可信。 默认情况下，所有接口都是不可信的。 对于不可信接口，交换机会截获所有的ARP请求和应答。交换机在更新本地缓存并把数据包转发到目的之前会验证截获的包是否有合法的IP-MAC地址绑定。交换机会丢弃非法的数据包，并根据全局配置命令**ip arpinspection vlan logging**指定的记录配置把这些包记录在日志缓存中
步骤 9	**end**	返回特权 EXEC 模式
步骤 10	使用以下show命令: • **show arp access-list** acl-name • **show ip arp inspection vlan** *vlan-range* • **show ip arp inspection interfaces**	验证配置的条目
步骤 11	**show running-config** **示例：** `Device#` **show running-config**	验证配置的条目
步骤 12	**copy running-config startup-config** **示例：** `Device#` **copy running-config**	（可选）把配置保存在配置文件中

	startup-config	

在 DHCP 环境中配置动态 ARP 监测

在开始前

以下展示了两台交换机都支持动态 ARP 监测特性时如何进行配置。主机 1 连接到交换机 A，主机 2 连接到交换机 B。两台交换机都在主机所属的 VLAN 1 上运行动态 ARP 监测。有一台 DHCP 服务器连接到交换机 A。两台主机都通过相同的 DHCP 服务器获取 IP 地址。因此，交换机 A 有主机 1 和主机 2 的绑定信息，交换机 B 有主机 2 的绑定信息。

注释： 动态 ARP 监测根据 DHCP 侦听数据库中的条目验证入向 ARP 请求和 ARP 应答中的 IP-MAC 地址绑定信息。确保让 DHCP 侦听允许动态分配 IP 地址的 ARP 包。

按照以下步骤配置动态 ARP 监测。管理员必须在两台交换机上都进行此配置过程。此过程是必需的。

总步骤

1. enable

2. show cdp neighbors

3. configure terminal

4. ip arp inspection vlan *vlan-range*

5. interface*interface-id*

6. ip arp inspection trust

7. end

8. show ip arp inspection interfaces

9. show ip arp inspection vlan *vlan-range*

10. show ip dhcp snooping binding

11. show ip arp inspection statistics vlan *vlan-range*

~~**12. configure terminal**~~

~~**13. configure terminal**~~

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：**	进入特权 EXEC 模式。在提示时输入密码

	Device>**enable**	
步骤 2	**show cdp neighbors** **示例:** Device(config-if)#**show cdp neighbors**	验证交换机之间的连接
步骤 3	**configure terminal** **示例:** Device# **configure terminal**	进入全局配置模式
步骤 4	**ip arp inspection vlan** *vlan-range* **示例:** Device(config)# **ip arp inspection vlan 1**	基于 VLAN 启用动态 ARP 监测。默认情况下，动态 ARP 监测在所有 VLAN 上禁用。对于 *vlan-range* 字段，可以使用 VLAN ID 编号指定一个 VLAN，可以指定由连字符分隔的 VLAN 范围，也可以指定由逗号分隔的一组 VLAN。VLAN 范围从 1 到 4096。请对两台交换机指定相同的 VLAN ID
步骤 5	**interface** *interface-id* **示例:** Device(config)# **interface gigabitethernet1/0/1**	指定连接到其他交换机的接口，并进入接口配置模式
步骤 6	**ip arp inspection trust** **示例:** Device(config-if)#**ip arp inspection trust**	把交换机之间的连接配置为可信。默认情况下，所有的接口都是不可信的。 交换机不会检查来自可信接口上其他交换机的 ARP 包。它会直接转发这些数据包。 对于不可信接口，交换机会截获所有的 ARP 请求和应答。交换机在更新本地缓存并把数据包转发到目的之前会验证截获的包是否有合法的 IP-MAC 地址绑定。交换机会丢弃非法的数据包，并根据全局配置命令 **ip arpinspection vlan logging** 指定的记录配置把这些包记录在日志缓存中
步骤 7	**end** **示例:**	返回特权 EXEC 模式

	Device(config-if)#**end**	
步骤 8	**show ip arp inspection interfaces**	验证接口上的动态 ARP 监测配置
步骤 9	**show ip arp inspection vlan** *vlan-range* **示例:** Device(config-if)#**show ip arp** **inspection** **vlan 1**	验证 VLAN 上的动态 ARP 监测配置
步骤 10	**show ip dhcp snooping binding** **示例:** Device(config-if)#**show ip dhcp** **snooping** **binding**	验证 DHCP 绑定信息
步骤 11	**show ip arp inspection statistics** **vlan** *vlan-range* **示例:** Device(config-if)#**show ip arp** **inspection** **statistics vlan 1**	检查 VLAN 上的动态 ARP 监测统计信息

对入向 ARP 包限速

动态 ARP 监测验证检查由交换机的 CPU 执行；因此，为避免拒绝服务攻击，入向 ARP 包的数量应被限速。

当入向 ARP 包的速率超过了配置的限制时，交换机会把接口置为错误禁用状态。端口会已知保持在此状态中，直到管理员启用了错误禁用恢复，允许端口在指定的超时间隔后自动脱离此状态。

注释： 除非在接口上配置了限速，否则改变接口的可信状态也会把该可信状态的限速设置变为默认值。配置限速之后，即使接口的可信状态变化，接口也保持限速设置。如果输入接口配置命令 **no ip arp inspection limit**，接口会恢复到默认的限速设置。

按照以下步骤设置入向 ARP 包的限速值。此步骤是可选的。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. ip arp inspection limit {rate pps [burst interval seconds] | none}

5. exit

6. 使用以下show命令:

- **errdisable detect cause arp-inspection**
- **errdisable recovery cause arp-inspection**
- **errdisable recovery interval** *interval*

7. exit

8. 使用以下show命令:

- **show ip arp inspection interfaces**
- **show errdisable recovery**

9. show running-config

10. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式
步骤 3	**interface** *interface-id*	指定要被限速的接口，进入接口配置模式
步骤 4	**ip arp inspection limit {rate pps [burst interval seconds] | none}**	对接口的入向 ARP 请求和应答进行限速。不可信接口上的默认速率是 15pps，可信接口无限制。突发间隔是 1 秒。 关键字的含义如下： • 对于 **rate** *pps*，指定每秒处理的入向数据包数量的上限。

		范围从 0 到 2048pps。 • （可选）对于 **burst interval** *seconds*，指定连续的间隔秒数，在此期间接口因高的 ARP 包速率被监控。范围从 1 到 15。 • 对于 **rate none**，指定对处理的入向 ARP 数据包数量不设上限
步骤 5	**exit**	返回全局配置模式
步骤 6	使用以下 show 命令: **• errdisable detect cause arp-inspection** **• errdisable recovery cause arp-inspection** **• errdisable recovery interval** *interval*	（可选）启用动态 ARP 监测错误禁用状态的错误恢复，配置动态 ARP 监测恢复机制的参数。 默认情况下，恢复被禁用，恢复间隔是 300 秒。 对于 **interval** *interval*，以秒的形式指定从错误禁用状态恢复的时间。范围从 30 到 86400
步骤 7	**exit**	返回特权 EXEC 模式
步骤 8	使用以下 show 命令: **• show ip arp inspection interfaces** **• show errdisable recovery**	验证设置
步骤 9	**show running-config** **示例：** `Device# show running-config`	验证配置的条目
步骤 10	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把配置保存在配置文件中

执行动态 ARP 监测验证检查

动态 ARP 监测截获、记录并丢弃含有非法 IP-MAC 地址映射信息的 ARP 包。管理员可以配置交换机对目的 MAC 地址、发送方和目标的 IP 地址以及源 MAC 地址进行额外的检查。

按照以下步骤配置对入向 ARP 包的特定检查。此过程是可选的。

总步骤

1. enable

2. configure terminal

3. ip arp inspection validate {[src-mac] [dst-mac] [ip]}

4. exit

5. show ip arp inspection vlan *vlan-range*

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式
步骤 3	**ip arp inspection validate** **{[src-mac] [dst-mac] [ip]}**	对入向 ARP 包执行特定的检查。默认情况下，这些检查不执行。 关键字的含义如下： • 对于 **src-mac**，把以太网报头中的源 MAC 地址与 ARP 报文中的发送方 MAC 地址进行比较。此检查对 ARP 请求和 ARP 应答都会进行。启用时，MAC 地址不同的数据包会被分类为非法数据包并被丢弃。

		• 对于 **dst-mac**，把以太网报头中的目的 MAC 地址与 ARP 报文中的目标 MAC 地址进行比较。此检查对 ARP 应答进行。启用时，MAC 地址不同的数据包会被分类为非法数据包并被丢弃。 • 对于 **ip**，检查 ARP 报文体中的非法 IP 地址。这些地址包括 0.0.0.0,255.255.255.255 以及所有的 IP 组播地址。对于所有的 ARP 请求和应答发送方 IP 地址都会被检查，对于 ARP 应答检查目标 IP 地址。 管理员必须至少指定一个关键字。每条命令都会覆盖之前命令的配置，即如果一条命令启用了 src 和 dstmac 验证，而第二条命令仅启用了 IP 验证，src 和 dstmac 的验证都会因第二条命令而被禁用
步骤 4	**exit**	返回特权 EXEC 模式
步骤 5	**show ip arp inspection vlan** *vlan-range*	验证设置
步骤 6	**show running-config** **示例：** `Device#` **show running-config**	验证配置的条目
步骤 7	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中

监控 DAI

使用以下命令监控 DAI：

命令	**描述**
clear ip arp inspection statistics	清除动态 ARP 监测统计数据。
show ip arp inspection statistics [**vlan** *vlan-range*]	显示特定 VLAN 的统计数据，包括转发、丢弃、MAC 验证失败、IP 验证失败、ACL 允许及拒绝以及 DHCP 允许及拒绝的数据包信息。如果不指定 VLAN 或者指定了 VLAN 范围，则只显示启用了动态 ARP 监测（active）的 VLAN 的信息
clear ip arp inspection log	清除动态 ARP 监测日志缓存
show ip arp inspection log	显示动态 ARP 监测日志缓存的配置及内容

对于 **show ip arp inspection statistics** 命令，交换机会递增每个可信动态 ARP 监测端口上转发的 ARP 请求和应答包的数量。交换机会递增被 ACL 或 DHCP 允许的数据包数量。对于每个被源 MAC、目的 MAC 或 IP 验证检查拒绝的数据包，交换机会增加对应的计数。

验证 DAI 配置

使用以下命令显示及验证 DAI 配置：

命令	**描述**
show arp access-list [*acl-name*]	显示 ARP ACL 的详细信息
show ip arp inspection interfaces [interface-id]	显示特定接口或者所有接口的可信状态以及 ARP 包限速设置
show ip arp inspection vlan *vlan-range*	显示特定 VLAN 的动态 ARP 监测配置及运行状态。如果不指定 VLAN 或者指定了 VLAN 范围，则只显示启用了动态 ARP 监测（active）的 VLAN 的信息

其他参考资料

错误信息解释

描述	**链接**
为帮助管理员搜索并解决该版本中的系统错误信息，管理员可使用错误信息解释工具。	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	http://www.icntnetworks.com

配置 IEEE 802.1x 基于端口的认证

本章描述如何配置 IEEE 802.1x 基于端口的认证。IEEE 802.1x 认证阻止未认证的设备（客户端）获取网络访问权限。除非另有说明，交换机一词表示独立交换机或交换机堆栈。

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于 802.1x 基于端口认证的信息

802.1x 标准定义了一个基于客户端服务器的访问控制及认证协议，可以阻止未被授权的客户端通过可公开访问的端口连接 LAN。在提供交换机或者 LAN 的服务之前，认证服务器会对每个连接到交换机端口的客户端进行认证。

注释： TACACS 不支持 802.1x 认证。

在客户端被认证之前，802.1x 访问控制只允许基于 LAN 的可扩展认证协议（Extensible Authentication Protocol overLAN，EAPOL）、Inspur 发现协议（Inspur Discovery Protocol，IDP）以及生成树协议（Spanning Tree Protocol，STP）的流量通过客户端连接的端口。认证成功后，正常流量才可以通过端口。

注释： 使用 **show platform software trace message smd** 命令查看 RADIUS 以及 AAA 的调试日志信息。更多信息参见 *Cisco IOS XE Denali 16.1.1 命令参考指南*的 Trace 命令章节。

基于端口的认证过程

要配置 IEEE 802.1x 基于端口的认证功能，必须启用认证、授权和审计（authentication, authorization, accounting，AAA）并指定认证方式列表。方式列表描述了认证用户时查询认证方式的顺序。

AAA 过程从认证开始。当启用了 802.1x 基于端口的认证且客户端支持兼容 802.1x 的客户端软件时，会发生以下事件：

- 如果客户端身份合法且 802.1x 认证成功，交换机允许客户端访问网络；
- 如果 802.1x 认证等待 EAPOL 消息交换超时且启用了 MAC 旁路认证，交换机可以使用客户端的 MAC 地址进行认证。如果客户端的 MAC 地址合法且认证成功，交换机允许客户端网络。如果因客户端的 MAC 地址非法而认证失败，在配置了访客 VLAN 的情况下交换机会给客户端分配一个访客 VLAN，提供有限的访问服务；
- 如果交换机在 802.1x 兼容的客户端上得到了非法的身份，在配置了首先 VLAN 的情况下交换机会给客户端分配一个受限 VLAN，提供有限的访问服务；
- 如果 RADIUS 认证服务器不可用且启用了不可访问旁路认证，交换机会允许客户端访问网络，并把置为临界认证状态的端口放在 RADIUS 配置或用户指定的接入 VLAN 中。

注释： 不可访问旁路认证也被称为临界认证或 AAA 失败策略。

如果在端口上启用了多域认证（Multi Domain Authentication，MDA），也可以使用以上流程加上适用于语音认证的例外情况进行认证。

下图显示了认证的过程。

图 120：认证流程图

Start	开始
Done	结束
Yes	是
No	否
Is the client IEEE 802.1x capable?	客户端是否兼容 IEEE 802.1x？
IEEE 802.1x authentication process times out.	IEEE 802.1x 认证过程超时。
Is MAC authentication bypass enabled?	是否启用了 MAC 旁路认证？
Use MAC authentication bypass.[1]	使用 MAC 旁路认证。[1]
Client MAC address identity is valid.	客户端 MAC 地址身份合法。
Client MAC address identity is invalid.	客户端 MAC 地址身份非法。
Assign the port to a VLAN.	分配端口给 VLAN。
Assign the port to a guest VLAN.[1]	分配端口给访客 VLAN。[1]
The switch gets an EAPOL message, and the EAPOL message exchange begins.	交换机收到了 EAPOL 消息，EAPOL 消息交换开始。
Start IEEE 802.1x port-based authentication.	开始 IEEE 802.1x 基于端口的认证。
User does not have a certificate but the system previously logged on to the network using a computer certificate.	用户没有证书，但是系统之前使用计算机证书登录过网络。

Assign the port to a restricted VLAN.	分配端口给受限 VLAN。
Client identity is invalid.	客户端身份非法。
Client identity is valid.	客户端身份合法。
All authentication servers are down.	所有认证服务器均故障。
Use inaccessible authentication bypass(critical authentication) to assign the critical port to a VLAN.	使用不可访问旁路认证（临界认证）分配临界端口给 VLAN。
1 = This occurs if the switch does not detect EAPOL packets from the client.	1 =在交换机没有从客户端检测到 EAPOL 包时发生。

以下情况之一发生时交换机重新认证客户端：

- 启用了周期性重新认证，且重新认证计时器超时。

  用户可以配置重新认证计时器使用交换机特定的值或者基于 RADIUS 服务器的值计时。配置使用 RADIUS 服务器进行 802.1x 认证后，交换机基于会话超时 RADIUS 属性（属性[27]）以及终止操作 RADIUS 属性（属性[29]）设置计时器。

  会话超时 RADIUS 属性（属性[27]）指定了进行重新认证的经过时间。

  终止操作 RADIUS 属性（属性[29]）指定了在重新认证过程中需要采取的操作。这些操作是*初始化*以及*重新认证*。设置*初始化*操作时（该属性值为*默认*），802.1x 会话结束，且重新认证过程中连接性丢失。设置*重新认证*操作时（该属性值为 RADIUS 请求），会话在重新认证过程中不受影响；
- 可以输入特权 EXEC 命令 **dot1x re-authenticate interface** *interface-id* 手动重新认证客户端。

基于端口的认证初始化及消息交换

在 802.1x 认证期间，交换机或客户端可以发起认证。如果使用接口配置命令 **authentication port-control auto** 启用了端口认证，交换机会在链路状态从 down 变为 up 时发起认证，或在端口保持 up 且未认证状态时周期性地发起认证。交换机给客户端发送 EAP 请求/身份数据帧请求其身份。接收到数据帧之后，客户端会使用 EAP 应答/身份帧回应。

然而，如果在启动期间客户端没有从交换机上收到 EAP 请求/身份帧，客户端可通过发送 EAP 开始帧发起认证，这会促使交换机请求客户端的身份。

注释： 如果 802.1x 认证未在网络接入设备上启用或设备不支持，来自客户端的 EAPOL 帧会被丢弃。如果客户端三次尝试接收 EAP 请求/身份失败，客户端会当作端口在已认证的状态来发送数据帧。处于已认证状态的端口相当于客户端已经被成功认证。

客户端提供自己的身份信息后，交换机开始作为中介的角色，在客户端和认证服务器之间传

递 EAP 帧直到认证成功或者失败。如果认证成功，交换机端口变为已认证状态。如果认证失败，可以重新尝试认证，端口可能被分配给提供有限的接入服务的 VLAN，也可能被授予网络访问权限。

具体的 EAP 交换过程取决于使用的认证方式。

下图显示了客户端和 RADIUS 服务器使用一次性密码（One-Time-Password，OTP）认证方式时由客户端发起的消息交换过程。

图 121：消息交换

Client	客户端
Authentication server(RADIUS)	认证服务器（RADIUS）
EAPOL-Start	EAPOL 开始
EAPOL-Request/Identity	EAPOL 请求/身份
EAPOL-Response/Identity	EAPOL 应答/身份
EAPOL-Request/OTP	EAPOL 请求/OTP
EAPOL-Response/OTP	EAPOL 应答/OTP
EAPOL-Success	EAPOL 成功
EAPOL-Logoff	EAPOL 登出
RADIUS Access-Request	RADIUS 访问请求
RADIUS Access-Challenge	RADIUS 访问挑战
RADIUS Access-Request	RADIUS 访问请求
RADIUS Access-Accept	RADIUS 访问接受
Port Authorized	端口已认证
Port Unauthorized	端口未认证

如果等待 EAPOL 消息交换时 802.1x 认证超时且启用了 MAC 旁路认证，交换机可以在从客户端检测到以太网数据包时授权客户端。交换机使用客户端的 MAC 地址作为其身份，并把此信息包含在发往 RADIUS 服务器的 RADIUS 访问请求帧中。在服务器给交换机发送了 RADIUS 访问接受帧之后（授权成功），端口变为已认证。如果授权失败且指定了访客 VLAN，交换机会把端口分配给访客 VLAN。如果交换机等待以太网数据包时检测到了 EAPOL 包，交换机会停止 MAC 旁路认证过程并开始 802.1x 认证。

下图显示了 MAC 旁路认证过程中的消息交换过程。

图 122：MAC 旁路认证过程中的消息交换

Client	客户端
Authentication server(RADIUS)	认证服务器（RADIUS）

Switch	交换机
EAPOL-Request/Identity	EAPOL 请求/身份
RADIUS Access-Request	RADIUS 访问请求
RADIUS Access-Accept	RADIUS 访问接受
Ethernetpacket	以太网数据包

基于端口认证的认证管理器

基于端口的认证方式

表 145:802.1x 特性

认证方式	模式			
	单主机	多主机	MDA	多认证
802.1x	VLAN 分配 基于用户的 ACL 过滤 ID 属性 可下载的 ACL[16] 重定向 URL	VLAN 分配	VLAN 分配 基于用户的 ACL 过滤 ID 属性 可下载的 ACL 重定向 URL	VLAN 分配 基于用户的 ACL 过滤 ID 属性 可下载的 ACL 重定向 URL
MAC 旁路认证	VLAN 分配 基于用户的 ACL 过滤 ID 属性 可下载的 ACL 重定向 URL	VLAN 分配	VLAN 分配 基于用户的 ACL 过滤 ID 属性 可下载的 ACL 重定向 URL	VLAN 分配 基于用户的 ACL 过滤 ID 属性 可下载的 ACL 重定向 URL
独立网页认证	代理 ACL，过滤 ID 属性，可下载的 ACL			
NAC 二层 IP 验证	过滤 ID 属性 可下载的 ACL 重定向 URL	过滤 ID 属性 可下载的 ACL 重定向 URL	过滤 ID 属性 可下载的 ACL 重定向 URL	过滤 ID 属性 可下载的 ACL 重定向 URL
备用的网页认证[17]	代理 ACL，过滤 ID 属性，可下载	代理 ACL，过滤 ID 属性，可下载	代理 ACL，过滤 ID 属性，可下载	代理 ACL，过滤 ID 属性，可下载

	的 ACL	的 ACL	的 ACL	的 ACL

[16] InspurINOS 12.2(50)SE 及之后版本支持。

[17] 对于不支持 802.1x 认证的客户端使用。

基于用户的 ACL 和过滤 ID 属性

注释： 在 ACL 中源只能设置为 **any**。

注释： 对于为多主机模式配置的 ACL，声明的源部分必须是 *any*（比如 **permit icmp** *any* **host 10.10.1.1**）。

对于定义的任意 ACL，源部分必须指定为 **any**。否则，ACL 不能被应用且认证会失败。单主机模式是唯一的例外，可以向后兼容。

在启用 MDA 或多认证的端口上可以认证多台主机。应用于一台主机的 ACL 策略不会影响其他主机的流量。如果一台主机在一个多主机端口上进行了认证，而其他主机没有认证就获得了网络访问权限，通过设置源地址为 any 可以把用于第一台主机的 ACL 策略应用到其他连接的主机上。

基于端口认证管理器的 CLI 命令

认证管理器的接口配置命令管理所有的认证方式，比如 802.1x、MAC 旁路认证以及网页认证。认证管理器的命令决定应用到连网主机上的认证方式的优先级以及顺序。

认证管理器的命令控制通用的特征特性，比如主机模式、违反模式以及认证计时器。通用的认证命令包括接口配置命令 **authentication host-mode**，**authentication violation**，以及 **authentication timer**。

802.1x特性的命令以**dot1x**关键字开始。例如，接口配置命令**authentication port-controlauto**在接口上启用认证。然而，全局配置命令**dot1xsystem-authentication control**只能全局地启用或禁用802.1x认证。

注释： 如果全局禁用了802.1x认证，其他认证方式仍会在端口上启用，如网页认证。

authentication manager 命令与以前的 802.1x 命令功能相同。

当过滤掉由认证管理器生成的详细系统消息时，被过滤的内容通常与认证成功有关。也可以过滤 802.1x 认证以及 MAB 认证的详细消息。每种认证方式都有独立的配置命令：

- 全局配置命令 **no authentication logging verbose** 过滤来自认证管理器的详细消息；
- 全局配置命令 **no dot1x logging verbose** 过滤 802.1x 认证的详细消息；
- 全局配置命令 **no mab logging verbose** 过滤 MAC 旁路认证（MAB）的详细消息。

表 146：认证管理器命令及以前的 802.1x 命令

Inspur INOSRelease 12.2(50)SE 及之后版本的认证管理器命令	**Inspur INOSRelease 12.2(46)SE 及之前版本的等同 802.1x 命令**	**描述**
authentication control-direction {both | in}	**dot1x control-direction {both | in}**	启用 802.1x 认证以及局域网唤醒（wake-on-LAN，WoL）特性，并配置单向或者双向端口控制
authentication event	**dot1x auth-fail vlan** **dot1x critical (interface configuration)** **dot1x guest-vlan6**	在端口上启用受限 VLAN。 启用不可访问旁路认证特性。 指定一个活跃的 VLAN 作为 802.1x 的访客 VLAN
authentication fallback *fallback-profile*	**dot1x fallback** *fallback-profile*	配置端口对于不支持 802.1x 认证的客户端使用网页认证作为备用方式
authentication host-mode [multi-auth | multi-domain | multi-host | single-host]	**dot1x host-mode {single-host | multi-host | multi-domain}**	允许 802.1x 授权的端口上有一个或多个主机（客户端）
authentication order	**mab**	提供灵活定义使用的认证方式的功能
authentication periodic	**dot1x reauthentication**	启用客户端的周期性重新认证
authentication port-control {auto | force-authorized | force-un authorized}	**dot1x port-control {auto | force-authorized | force-unauthorized}**	允许人工控制端口的认证状态
authentication timer	**dot1x timeout**	设置 802.1x 的计时器
authentication violation {protect | restrict | shutdown}	**dot1x violation-mode {shutdown | restrict | protect}**	配置新设备连接到端口或者在端口已连接了最大数量的设备之后有新设备连接到端口时的违反模式

已认证以及未认证状态的端口

在802.1x认证过程中，交换机可以根据交换机端口的状态授予一个客户端访问网络的权限。端口开始的状态是*未认证*。在此状态中的未被配置为语音VLAN的端口会禁止除802.1x认证、CDP以及STP包之外的所有入向及出向流量。当客户端被成功授权时，端口的状态变为*已认证*状态，允许客户端的所有流量正常流通。如果端口被配置为语音VLAN端口，在客户端被成功认证之前，端口允许VoIP流量以及802.1x协议数据包通过。

注释： 不支持CDP旁路功能，这可能导致端口进入错误禁用状态。

如果不支持802.1x认证的客户端连接到了一个未认证的802.1x端口，交换机会请求客户端的身份。在此情况下，客户端不会响应请求，端口会保持未认证状态，而客户端不被授予网络访问权限。

相反，当启用802.1x的客户端连接到了不运行802.1x标准的端口时，客户端会发送EAPOL开始帧以发起认证过程。在未收到应答的情况下，客户端会发送固定次数的请求。因为仍未接收到应答，客户端会把端口当作是已认证的状态并开始发送数据帧。

管理员可以使用接口配置命令 **authentication port-control** 及以下关键字控制端口的认证状态：

- **force-authorized**——禁用802.1x认证，使端口无需经过认证交换过程而直接变为已认证状态。端口不对客户端进行基于802.1x的认证并正常收发流量。这是默认的设置；
- **force-unauthorized**——使端口保持未认证状态，忽略客户端的所有认证尝试。交换机不能通过该端口为客户端提供认证服务；
- **auto**——启用802.1x认证，让端口的开始状态是未认证，只允许EAPOL数据帧通过端口收发。认证过程在端口链路状态从down变为up时或者在收到EAPOL开始帧的时候开始。交换机请求客户端的身份并在客户端和认证服务器之间传递认证消息。每个尝试访问网络的客户端都会被交换机使用客户端的MAC地址唯一标识。

如果客户端认证成功（从认证服务器收到了接受帧），端口状态会变为已认证，且所有来自已认证客户端的数据帧都被允许通过端口。如果认证失败，端口会保持在未认证的状态，但可以重新尝试认证。如果认证服务器不可达，交换机可以重新发送请求。如果在特定的尝试次数之后还是没有接收到来自服务器的应答，认证失败，且不授予网络访问权限。

客户端登出时，会发送一个EAPOL登出消息，使交换机的端口变为未认证状态。

如果端口的链路状态从up变为down，或者收到了EAPOL登出帧，端口都会返回未认证的状态。

基于端口的认证以及交换机堆栈

如果一台交换机被添加进交换机堆栈或被从交换机堆栈移除，只要堆栈与 RADIUS 服务器保持 IP 连通性，802.1x 认证就不受影响。以上说明也适用于堆栈主用设备被从交换机堆栈移除的情况。如果堆栈主用设备故障，堆栈成员会通过选举过程成为新的堆栈主用设备，且 802.1x 认证过程照常进行。

如果因为连接到服务器的交换机被移除或故障，导致了与 RADIUS 服务器的 IP 连通性中断，会发生以下事件：

- 已经被认证且没有启用周期性重新认证的端口会保持已认证的状态。无需与 RADIUS 进行通信；
- 已经被认证且启用了周期性重新认证（使用全局配置命令**dot1xre-authentication**）的端口会在重新认证发生时认证失败。在重新认证过程中端口会返回未认证状态。需要与RADIUS服务器进行通信。

  对于正在进行的认证，认证会因为与服务器无连通性而立即失败。

如果发生故障的交换机启动并重新加入了交换机堆栈，认证可能失败也可能成功，这取决于交换机的启动时间以及尝试认证的时候与RADIUS服务器的连通性是否已经重建。

为了避免失去与RADIUS服务器的连通性，应确保存在冗余连接。例如，管理员可以让RADIUS服务器有到堆栈主用设备和堆栈成员的冗余连接，这样如果堆栈主用设备故障，交换机堆栈仍然有到RADIUS服务器的连通性。

802.1x 主机模式

管理员可以把端口配置成单主机模式或多主机模式。在单主机模式中，只有一台客户端可以连接到启用了 802.1x 的交换机端口。交换机可以通过发送 EAPOL 帧或者在端口链路状态变为 up 状态时发现客户端。如果客户端离开或者被另一个客户端代替，交换机会把端口的链路状态变为 down，且端口返回未认证状态。

在多主机模式中，可以把多台主机连接到一个启用了 802.1x 的端口。在此模式中，只需有一台连接的客户端被认证，所有客户端都可以被授予网络访问权限。如果端口变为未认证状态（重新认证失败或收到了 EAPOL 登出消息），交换机会拒绝所有连接主机的网络访问。

图 123：多主机模式示例

Wireless clients	无线客户端
Access point	接入点

Authentication server(RADIUS)	认证服务器（RADIUS）

注释： 对于所有主机模式，配置基于端口的认证时，认证之前链路协议保持为 up。

交换机支持多域认证（multidomain authentication，MDA），允许数据设备和语音设备（如 Inspur 或非 Inspur 的 IP 电话）同时连接到相同的交换机端口。

802.1x 多认证模式

多认证（multiple-authentication，multiauth）模式允许数据 VLAN 中有多个被认证的客户端。每台主机被独立认证。如果配置了语音 VLAN，此模式也允许 VLAN 上有一个客户端（如果端口检测到了其他的语音客户端，它们会被端口丢弃，但此时不会发生违反错误）。

如果启用了 802.1x 的端口连接了集线器或者无线接入点，每个连接的设备都必须进行认证。

对于非 802.1x 设备，可以使用 MAC 旁路认证或者网页认证作为备用的主机认证方式，对单个端口上的不同主机使用不同的认证方式。

多认证端口可以认证的数据主机数量没有限制。然而，如果配置了语音 VLAN，只允许有一台语音设备。因为没有定义主机数量限制，也就不会触发违反操作，如果发现了第二台语音设备，其流量会被静默地丢弃。对于语音 VLAN 上的 MDA 功能，多认证模式会根据从认证服务器收到的 VSA 把已认证的设备分配到数据 VLAN 或者语音 VLAN。

注释： 端口在多认证模式时，访客 VLAN 以及认证失败的 VLAN 特性不被激活。

在以下情况中，多认证模式可以分配 RADIUS 服务器提供的 VLAN：

- 主机是端口上认证的第一台主机，且 RADIUS 服务器提供了 VLAN 信息；
- 后续认证主机使用的 VLAN 与运行的 VLAN 相同；
- 端口上被认证的主机没有 VLAN 分配信息，且后续主机也没有 VLAN 分配，或者其 VLAN 信息与运行的 VLAN 相同；
- 端口上第一台被认证的主机有一组 VLAN 分配信息，且后续主机没有 VLAN 分配，或者其 VLAN 组信息与端口的 VLAN 组信息相同。后续主机必须与第一台主机使用 VLAN 组中相同的 VLAN。如果使用了 VLAN 列表，所有主机都要服从 VLAN 列表中定义的条件；
- 多认证端口上只支持一个语音 VLAN 的分配；
- 在给端口上的主机分配了 VLAN 之后，后续主机必须有相同的 VLAN 信息，否则就会被拒绝访问端口；
- 在多认证模式中不能配置访客 VLAN 或者认证失败 VLAN；
- 多认证模式下临界认证 VLAN 的行为不变。当主机尝试认证而服务器不可达时，所有已认证的主机都会被重新初始化到配置的 VLAN 中。

基于用户 VLAN 分配的多认证

注释： 此特性只在运行 LAN Base 镜像的 Inspur 2960X 交换机上支持。

基于用户 VLAN 分配的多认证特性允许在拥有一个配置的接入 VLAN 的端口上，根据分配给端口上客户端的 VLAN 创建多个运行的接入 VLAN。配置为接入端口的交换机端口不进行 dot1q 标记，其上的所有 VLAN 的流量都与数据域关联，且这些 VLAN 被当做本征 VLAN。

每个多认证端口的主机数量是 8 个，然而也可以有更多的主机。

注释： 基于用户 VLAN 分配的多认证特性不支持语音 VLAN。端口上所有语音域中的客户端都是用一个 VLAN。

以下是基于用户 VLAN 分配的多认证情景：

情景一

集线器连接到接入端口，且端口配置了接入 VLAN（V0）。

主机（H1）通过集线器分配了 VLAN（V1）。端口的运行 VLAN 被改为 V1。此行为与单主机或多域认证端口相似。

当第二台主机（H2）连接集线器且被分配了 VLAN（V2），端口将有两个运行的 VLAN（V1 和 V2）。如果 H1 和 H2 发送了未打标记的入向流量，H1 的流量会被映射到 VLAN（V1），而 H2 的流量会被映射到 VLAN（V2），所有该端口的出向 VLAN（V1）和 VLAN（V2）的流量都不会被打标记。

如果两台主机 H1 和 H2 都登出，或者会话因故被移除，VLAN（V1）和 VLAN（V2）会被从端口上移除，且端口恢复为配置的 VLAN（V0）。

情景二

集线器连接到接入端口，且端口配置了接入 VLAN（V0）。

主机（H1）通过集线器分配了 VLAN（V1）。端口的运行 VLAN 被改为 V1。

当第二台主机（H2）连接到集线器，被授权且没有显式的 VLAN 策略，H2 希望使用恢复到端口上的配置的 VLAN（V0）。所有从 VLAN（V0）和 VLAN（V1）发出的出向流量都不被打标记。

如果主机（H2）登出或因故会话被移除，配置的 VLAN（V0）会被从端口上移除，而 VLAN（V1）会成为端口上的唯一运行 VLAN。

情景三

集线器连接到开放模式的接入端口，且端口配置了接入 VLAN（V0）。

主机（H1）通过集线器分配了 VLAN（V1）。端口的运行 VLAN 被改为 V1。当第二台主机（H2）连接上且保持未认证时，因为使用开放模式，其仍然可以访问运行 VLAN（V1）。

如果主机（H1）登出或因故会话被移除，VLAN（V1）被从端口上移除，且主机（H2）被分配到 VLAN（V0）。

注释： 开放模式以及 VLAN 分配的组合对主机（H2）有负面影响，因为其 IP 地址子网对应于 VLAN（V1）。

基于用户 VLAN 分配的多认证的限制

在基于用户 VLAN 分配的多认证特性中，一个端口上来自多个 VLAN 的出向流量不会被打标记，主机会收到发给其他主机的流量。这可能对广播以组播流量造成问题。

- **IPv4 ARP**：主机会接收到来自其他子网的 ARP 包。如果端口上有两个活跃的在不同虚拟路由转发（Virtual Routing and Forwarding，VRF）表的子网，且子网使用了重叠的 IP 地址范围，就会发生问题。主机的 ARP 缓存可能有非法的条目；
- **IPv6 控制包**：在 IPv6 中，路由器通告（Router Advertisements，RA）会被不应接收的主机处理。当 VLAN 中的一台主机接收到了来自不同 VLAN 的 RA，主机会给自己分配不正确的 IPv6 地址。这样的主机无法访问网络。

  解决方法是启用 IPv6 首跳安全功能，让广播的 ICMPv6 包转化为单播包并从启用了多认证的端口发出。此时的数据包会复制给多认证端口上属于 VLAN 的每个客户端，且目的 MAC 地址会被设置为每个客户端的地址。如果端口有一个 VLAN，则 ICMPv6 包正常广播；
- **IP 组播**：如果 VLAN 中的主机加入了组播组，发往组播组的流量会被复制给不同的 VLAN。如果一个多认证端口上的两个不同 VLAN 的主机加入了一个组播组，每个组播包会从这个端口上发出两份。

MAC 移动

如果一个 MAC 地址在一个交换机端口上被认证，这个地址就不被允许出现该交换机上另一个启用了认证管理器的端口上。如果交换机在另一个启用了认证管理器的端口上检测到了相同的 MAC 地址，该地址不被允许。

有一些情况下 MAC 地址可能需要从一个端口移动到相同交换机的另一个端口。比如，当认证的主机和交换机端口之间有另一台设备时（如集线器或者 IP 电话），管理员可能希望断开主机与另一个台设备的连接并直接连接到相同交换机的另一个端口上。

可以全局启用 MAC 移动特性，设备会在新端口上重新被认证。当主机移动到第二个端口上时，第一个端口上的会话会被删除，而主机会在新端口上重新被认证。MAC 移动在所有主机模式中都支持（被认证的主机可以移动到交换机的任意端口上，无论该端口启用了何种主机模式）。当 MAC 地址从一个端口移动到另一个，交换机会结束原始端口上的认证会话，并

在新端口上发起新的认证过程。MAC 移动特性对于语音和数据主机都适用。

注释： 在开发认证模式中，MAC 地址可以立即从原始端口移动到新端口上，而无需在新端口上进行认证。

MAC 替换

MAC 替换特性可以用来解决主机尝试连接到之前认证了另一台主机的端口的违规情况。

注释： 此特性不适用于多认证模式的端口，因为违规情况在该模式中不会被触发。此特性不适用于多主机模式的端口，因为在该模式中，只要求认证第一台主机。

如果配置了接口配置命令 **authentication violation** 以及 **replace** 关键字，多域模式端口的认证过程如下：

- 在有已认证 MAC 地址的端口上收到了新的 MAC 地址；
- 认证管理器会用新的 MAC 地址替换端口上当前数据主机的 MAC 地址；
- 认证管理器会发起新 MAC 地址的认证过程；
- 如果认证管理器确定新主机是语音主机，原始的语音主机会被移除；

如果端口为开放认证模式，新的 MAC 地址会立即被加入 MAC 地址表中。

802.1x 审计

802.1x 标准定义了如何对用户的网络访问进行认证和授权，但不记录网络的使用情况。

802.1x 审计默认被禁用。可以启用 802.1x 审计功能监控启用了 802.1x 的端口活动：

- 用户成功认证
- 用户登出
- 链路 down
- 重新认证成功
- 重新认证失败

交换机不会记录 802.1x 的审计信息。它会把这些信息发给 RADIUS 服务器，必须配置服务器记录审计消息。

802.1x 审计属性-值对

发送给 RADIUS 服务器的信息以属性-值（Attribute-Value，AV）对的形式展示。这些 AV 对给不同的应用提供数据（比如，审计程序可能需要 RADIUS 包中 Acct-Input-Octets 或

Acct-Output-Octets 属性的信息）

AV 对由配置了 802.1x 审计的交换机自动发送。交换机会发送三种类型的 RADIUS 审计包：

- 开始——在新用户会话开始时发送
- 中间——在现有会话更新时发送
- 停止——在会话终止时发送

注释： 使用命令 **show platform software trace message smd** 查看 RADIUS 和 AAA 的调试信息。更多信息参见 *Cisco IOS XE Denali 16.1.1 命令参考指南*的 Trace 命令章节。

下表列出了 AV 对及何时由交换机发出。

表 147：审计 AV 对

属性名	**AV 对名**	**开始**	**中间**	**停止**
属性[1]	User-Name	总是	总是	总是
属性[4]	NAS-IP-Address	总是	总是	总是
属性[5]	NAS-Port	总是	总是	总是
属性[8]	Framed-IP-Address	从不	有时 [18]	有时
属性[30]	Called-Station-ID	总是	总是	总是
属性[31]	Calling-Station-ID	总是	总是	总是
属性[40]	Acct-Status-Type	总是	总是	总是
属性[41]	Acct-Delay-Time	总是	总是	总是
属性[42]	Acct-Input-Octets	从不	总是	总是
属性[43]	Acct-Output-Octets	从不	总是	总是
属性[47]	Acct-Input-Packets	从不	总是	总是
属性[48]	Acct-Output-Packets	从不	总是	总是
属性[44]	Acct-Session-ID	总是	总是	总是
属性[45]	Acct-Authentic	总是	总是	总是
属性[46]	Acct-Session-Time	从不	总是	总是
属性[49]	Acct-Terminate-Cause	从不	从不	总是
属性[61]	NAS-Port-Type	总是	总是	总是

[18] Framed-IP-Address AV 对在配置了合法的静态 IP 地址或当 DHCP 侦听绑定表中存在主机的 DHCP 绑定时发送。

802.1x 就绪状态检查

802.1x 就绪状态检查监控交换机所有端口上的 802.1x 活动，且显示连接到端口的支持 802.1x 的设备的信息。可以使用此特性确定连接到交换机端口的设备是否兼容 802.1x。对于不支持 802.1x 的设备可以使用如 MAC 旁路认证以及网页认证等其他认证方式。

此特性只在客户端支持使用 NOTIFY EAP 通知包查询时有效。客户端必须在 802.1x 超时时间之内应答。

交换机到 RADIUS 服务器的通信

RADIUS 安全服务器的标识方式有主机名或 IP 地址、主机名及特定的 UDP 端口号以及 IP 地址以及特定的 UDP 端口号。IP 地址和 UDP 端口号的组合是唯一的标识符，允许把 RADIUS 请求发给相同 IP 地址服务器上的多个 UDP 端口。可以为相同 RADIUS 服务器的相同服务（如认证）配置两个不同的主机条目，第二个条目作为第一个条目的备用项。RADIUS 主机条目会按照配置的顺序被尝试访问。

进行 VLAN 分配的 802.1x 认证

交换机支持进行 VLAN 分配的 802.1x 认证。在端口 802.1x 认证成功后，RADIUS 服务器会发送 VLAN 分配信息来配置交换机端口。RADIUS 服务器数据库维护着用户名到 VLAN 的映射，基于连接到交换机端口的客户端用户名分配 VLAN。可以使用此特性限制特定用户的网络访问。

Inspur INOS 12.2(37)SE 版本支持多域主机模式中的语音设备认证。在 Inspur INOS 12.2(40)SE 及之后版本中，当语音设备被授权且 RADIUS 服务器返回了授权的 VLAN 时，会配置端口上的语音 VLAN 为指定的 VLAN 并收发数据包。在启用多域认证（MDA）的端口上，语音 VLAN 的分配过程与数据 VLAN 相同。

在交换机和 RADIUS 服务器上配置时，进行 VLAN 分配的 802.1x 认证特征如下：

- 如果 RADIUS 服务器没有提供 VLAN 或者禁用了 802.1x 认证，认证成功后端口被配置在其所属的接入 VLAN 中。接入 VLAN 是分配给接入端口的 VLAN。在这个端口上收发的所有数据包都属于此 VLAN；
- 如果启用了 802.1x 认证但是来自 RADIUS 服务器的 VLAN 信息不合法，认证失败且配置的 VLAN 保持使用。这避免了端口因为配置错误意外地出现在不合适的 VLAN 中。

配置错误的情况可能包括为被路由端口指定了 VLAN、异常的 VLAN ID、不存在或内部（被路由端口）VLAN ID、RSPAN VLAN 以及关闭或停用的 VLAN。在多域主机端口上，配置错误也可能包括尝试分配与配置或指定的语音 VLAN ID 相同的数据 VLAN（反之亦然）；

- 如果启用了 802.1x 认证，且所有来自 RADIUS 服务器的信息都是合法的，被授权的设备会在认证后被置于指定的 VLAN 中；
- 如果在 802.1x 端口上启用了多主机模式，所有主机都会被置入与第一台认证主机相同的 VLAN（有 RADIUS 服务器指定）中；
- 启用端口安全特性不会影响 RADIUS 服务器分配的 VLAN 行为；
- 如果端口上禁用了 802.1x 认证，端口会恢复配置的接入 VLAN 以及配置的语音 VLAN。
- 如果 802.1x 端口被认证且被置入了 RADIUS 服务器分配的 VLAN 中，任何对端口接入 VLAN 配置的更改都不会生效。在多域主机的场景中，以上规则适用于完全授权语音设备，但包含例外情况。
  - 如果一台设备的配置变化导致其 VLAN 与其他设备配置或分配的 VLAN 相同，那么该端口上所有设备的授权都会被终止，且多域主机模式被禁用，直到恢复了数据和语音设备配置的 VLAN 不相同的合法配置；
  - 如果一台语音设备被授权且使用下载的语音 VLAN，移除语音 VLAN 配置或者将配置值修改为 dot1p 或未标记都会导致语音设备变为未授权且多域主机模式被禁用。

当端口在强制授权、强制未授权、未授权或关闭状态时，端口会被置入配置的接入 VLAN。

进行 VLAN 分配的 802.1x 认证特性不支持中继端口、动态端口以及通过 VLAN 成员策略服务器（VLAN Membership Policy Server，VMPS）进行动态接入分配的端口。

要配置 VLAN 分配，用户需要执行以下操作：

- 使用 **network** 关键字启用 AAA 认证，允许 RADIUS 服务器配置接口；
- 启用 802.1x 认证（在接入端口上配置 802.1x 认证时 VLAN 分配特性会被自动启用）；
- 指定 RADIUS 服务器厂商特定的隧道属性。RADIUS 服务器必须给交换机返回以下属性：
  - [64] 隧道类型（Tunnel-Type） = VLAN
  - [65] 隧道介质类型（Tunnel-Medium-Type） = 802
  - [81] 隧道私有组 ID（Tunnel-Private-Group-ID）= VLAN 名或 VLAN ID
  - [83] 隧道偏好（Tunnel-Preference）

  属性[64]必须包含 *VLAN*（类型 13）值。属性[65]必须包含值 *802*（类型 6）。属性[81]指定分配给 IEEE 802.1x 认证用户的 *VLAN 名称或 VLAN ID*。

使用基于用户 ACL 的 802.1x 认证

可以启用基于用户的访问控制列表（ACL），为经过 802.1x 认证的用户提供不同等级的网络访问及服务。当 RADIUS 服务器认证了一个连接到 802.1x 端口的用户时，服务器会基于用户的身份获取 ACL 属性并将其发送给交换机。交换机会把这些属性在用户会话的持续时间内应用到 802.1x 端口上。当会话结束，认证失败或者链路 down 发生时，交换机会移除基于用户的 ACL 配置。交换机不会在运行配置中保存 RADIUS 指定的 ACL。端口为未授权状态时，交换机会把 ACL 从端口移除。

可以在相同的交换机上同时配置路由器 ACL 及输入端口 ACL。然而，端口 ACL 优先于路由器 ACL。如果把输入端口 ACL 应用到属于某个 VLAN 的端口，端口的 ACL 会优先于应用在 VLAN 接口上啊的输入路由器 ACL。在应用了端口 ACL 的端口上收到的入向数据包会被端口 ACL 进行过滤。在其他端口上收到的入向被路由数据包会被路由器 ACL 过滤。出向的被路由数据包会被路由器 ACL 过滤。为了避免配置冲突，管理员应该小心规划存储在 RADIUS 服务器上的用户配置。

RADIUS 支持基于用户的属性，包括厂商特定的属性。这些厂商特定的属性（vendor-specific attribute，VSA）在认证过程中以八位字节格式传递给交换机。对于 VSA 中基于用户的 ACL，入方向是 inacl#<n>，出方向是 outacl#<n>。MAC ACL 仅在入方向支持。交换机仅在入方向支持 VSA，不支持二层端口的出方向端口 ACL。

应只使用扩展 ACL 语法风格定义存储在 RADIUS 服务器上的基于用户的配置。当收到 RADIUS 服务器传输的这些属性时，交换机会按照扩展的命名方式创建 ACL。然而，如果使用 Filter-Id 属性，则可以指向一个标准的 ACL。

可以使用 Filter-Id 属性指定一个已经在交换机上配置了的入向或出向 ACL。此属性包含 ACL 编号，以及入向过滤的*.in* 或者出向过滤的*.out*。如果 RADIUS 服务器不支持*.in* 或*.out* 语法，访问列表默认被应用为出向 ACL。因为交换机上的 Inspur INOS 仅支持有限数量的访问列表，所以只支持 Filter-Id 属性编号从 1 到 199 以及 1300 到 2699 的 IP ACL（IP 标准及 IP 扩展 ACL）。基于用户 ACL 的最大尺寸的 4000 个 ASCII 字符，但受限于 RADIUS 服务器基于用户 ACL 的最大尺寸。

要配置基于用户的 ACL：

- 启用 AAA 认证
- 启用 AAA 授权，使用 **network** 关键字允许 RADIUS 服务器进行接口配置
- 启用 802.1x 认证
- 在 RADIUS 服务器上配置用户配置及 VSA

- 将 802.1x 端口为单主机模式

注释： 基于用户的 ACL 仅在单主机模式中支持。

使用可下载 ACL 以及重定向 URL 的 802.1x 认证

可以在 802.1x 认证或者 MAC 旁路认证期间从 RADIUS 服务器向交换机下载 ACL 或者重定向 URL。也可以在网页认证期间下载 ACL。

注释： 可下载的 ACL 也被称为 *dACL*。

如果有多台主机被认证且主机在单主机、MDA 或多认证模式中，交换机会把 ACL 中的源地址改为主机的 IP 地址。

可以把 ACL 以及重定向 URL 应用到连接到 802.1x 端口的所有设备上。

如果 802.1x 认证期间没有下载 ACL，交换机会为主机在端口上应用静态默认 ACL。在配置为多认证或 MDA 模式的语音 VLAN 端口上，交换机只会把 ACL 当作授权策略的一部分应用给电话。

从 Inspur INOS 12.2(55)SE 版开始，如果端口上没有静态 ACL，交换机会创建一个动态的认证默认 ACL，在可下载 ACL 应用之前执行策略。

注释： 认证默认 ACL 不会出现在运行配置中。

当在端口上检测到至少有一台主机有授权策略时，授权默认 ACL 会被创建。

当最后一个认证的会话结束时，授权默认 ACL 会被移除。可以使用全局配置命令 **ip access-list extended auth-default-acl** 配置授权默认 ACL。

注释： 单主机模式中的认证默认 ACL 不支持 Inspur 发现协议（CDP）旁路模式。为了支持 CDP 旁路，必须在接口上配置静态 ACL。

802.1x 和 MAB 认证方式支持两种认证模式，*开放（open）*及*闭合（closed）*。如果*闭合*认证模式的端口上没有静态 ACL：

- 认证默认 ACL 会被创建；
- 在执行策略之间，认证默认 ACL 只允许 DHCP 流量；
- 当第一台主机认证时，授权策略被应用且不插入 IP 地址；
- 当检测到第二台主机时，用于第一台主机的策略被刷新，首个及后续会话的策略会插入 IP 地址并执行。

如果*开放*认证模式的端口上没有静态 ACL：

- 开放认证默认 ACL 会被创建，允许所有流量通过；
- 为避免安全漏洞，将执行插入了 IP 地址的策略；
- 网页认证受制于开放认证默认 ACL。

为了控制没有授权策略的主机的访问，可以配置指令。支持的指令值是*开放（open）*和*默认（default）*。配置*开放*指令时，所有流量都被允许。*默认*指令让流量受限于端口提供的接入权限。可以在AAA服务器的用户配置中配置指令，也可以在交换机上配置。在AAA服务器上配置指令，请使用全局命令**authz-directive =<open/default>**。在交换机上配置指令，请使用全局配置命令**epmaccess-control open**。

注释： 指令的默认值是*默认*。

如果主机在没有配置ACL的端口上使用备用的网页认证：

- 如果端口是开放认证模式，交换机会创建开放认证默认ACL；
- 如果端口时闭合认证模式，交换机会创建认证默认ACL。

备用ACL中的访问控制条目（access control entries，ACE）会被转换为基于用户的条目。如果配置的备用配置不包括备用ACL，主机会受限于与端口关联的认证默认ACL。

注释： 如果网页认证使用了自定义的logo且存储在外部服务器上，端口的ACL必须允许在认证之前访问外部服务器。管理员必须配置静态端口ACL或者更改认证默认ACL，以提供到外部服务器的连接。

用于重定向 URL 的 Inspur 安全 ACS 及属性-值对

交换机使用以下*inspur-av-pair* VSA：

- URL重定向（url-redirect）是HTTP或HTTPS URL；
- URL重定向ACL（url-redirect-acl）是交换机ACL名称或编号。

交换机使用Inspur安全定义ACL属性-值（AV）对来截获终端的HTTP或HTTPS请求。交换机之后将客户端的网页浏览器跳转到特定的重定向地址。Inspur安全ACS上的url-redirect AV包含浏览器被重定向到的URL。url-redirect-acl属性值对包含要进行特定HTTP或HTTPS流量重定向的ACL名称或编号。

注释：

- 匹配ACL中permit ACE的流量被重定向；
- 在交换机上定义URL重定向ACL以及默认端口ACL。

如果认证服务器上为客户端配置了重定向URL，必须在客户端连接的交换机端口上配置默认端口ACL。

用于可下载 ACL 的 Inspur 安全 ACS 及属性-值对

用户可以在 Inspur 安全 ACS 上设置 Inspur 安全定义的 ACL 属性-值（AV）对，使用厂商特定

属性（VSA）：RADIUS Inspur AV 对。这一对值使用#ACL#-IP-name-number 属性，指定了 Inspur 安全 ACS 上的可下载 ACL 的名称。

- *name* 是 ACL 的名称
- *number* 是版本号（如 3f783768）

如果认证服务器上为客户端配置了可下载ACL，必须在客户端连接的交换机端口上配置默认端口ACL。

如果在交换机上配置了默认的ACL，且Inspur安全ACS给交换机发送了主机访问策略，交换机会把策略应用到来自交换机端口连接的主机的流量上。如果不应用策略，交换机会应用默认ACL。如果Inspur安全ACS给交换机发送了可下载的ACL，此ACL优先于交换机端口上配置的默认ACL。然而，如果交换机从Inspur安全ACS接收了一个主机访问策略，但没有配置默认ACL，交换机会声明授权失败。

基于 VLAN ID 的 MAC 认证

如果希望基于静态的 VLAN ID 而不是可下载的 VLAN 来认证主机，可以使用基于 VLAN ID 的 MAC 认证特性。在交换机上配置静态 VLAN 策略时，VLAN 信息会和每台请求认证的主机的 MAC 地址一同发给 IAS（Microsoft）RADIUS 服务器。配置在连接端口上的 VLAN ID 会被用来进行 MAC 认证。通过同时使用基于 VLAN ID 的 MAC 认证和 IAS 服务器，网络中可以有固定数量的 VLAN。

此特性也限制了 STP 监控及处理的 VLAN 数量。可以把网络中的 VLAN 当作固定的来管理。

注释： Inspur ACS 服务器不支持此特性（ACS 服务器会忽略发来的新主机的 VLAN ID，并仅基于 MAC 地址进行认证）。

使用访客 VLAN 的 802.1x 认证

可以为交换机上的每个 802.1x 端口配置一个访客 VLAN，给客户端提供有限的服务，比如下载 802.1x 客户端软件。这些客户机可以升级系统以进行 802.1x 认证，而一些主机可能不兼容 IEEE 802.1x，如运行 Windows 98 系统的主机。

在 802.1x 端口上启用了访客 VLAN 时，交换机没有收到发送的 EAP 请求/身份帧的应答，或者客户端没有发送 EAPOL 包时，交换机会把访客 VLAN 分配给客户端。

交换机会维护 EAPOL 包的历史。如果在链路的生存时间内在接口上检测到了 EAPOL 包，交换机会认为连接到该接口的设备是兼容 IEEE 802.1x 的，接口也就不会变为访客 VLAN 的状态。如果接口的链路状态变为 down，EAPOL 历史会被清空。如果未在接口上检测到 EAPOL 包，

接口会变为访客 VLAN 状态。

如果交换机尝试授权一台兼容 802.1x 的语音设备，而此时 AAA 服务器不可用，授权尝试会失败，但检测到 EAPOL 包的事件会被保存在 EAPOL 历史中。当 AAA 服务器可用时，交换机会授权该语音设备。然而，交换机不再允许其他设备接入访客 VLAN。为了避免这样的情况发生，可以使用以下命令之一：

- 输入接口配置命令 **authentication event no-response action authorize vlan** *vlan-id*，允许访问访客 VLAN；
- 输入接口配置命令 **shutdown**，接着再输入接口配置命令 **no shutdown** 以重启端口。

如果在链路的生存时间内设备给交换机发送了 EAPOL 包，交换机不再允许认证失败的客户端访问访客 VLAN。

注释： 如果在接口更改为访客 VLAN 之后检测到了 EAPOL 包，接口会返回到未授权状态，而 802.1x 认证会重启。

当交换机端口变为访客 VLAN 后，会允许任意数量的不兼容 802.1x 的客户端进行访问。如果一台兼容 802.1x 的客户端加入了配置了访客 VLAN 的端口，端口会变为未授权状态并被置入用户配置的接入 VLAN，而认证过程会重启。

802.1x 端口的访客 VLAN 在单主机、多主机、多认证以及多域模式中支持。

可以把除了 RSPAN VLAN、私有 VLAN 或语音 VLAN 之外的任意活跃 VLAN 配置为 802.1x 访客 VLAN。访客 VLAN 特性不被内部 VLAN（被路由端口）或中继端口支持，仅被接入端口支持。

交换机支持 MAC 旁路认证。在 802.1x 端口上启用了 *MAC 旁路认证*时，交换机会在 IEEE 802.1x 认证等待 EAPOL 消息交换超时的情况下基于 MAC 地址对客户端进行授权。在 802.1x 端口上检测到客户端时，交换机会等待来自客户端的以太网数据包。交换机会给认证服务器发送一个 RADIUS 访问/请求帧，其中带有基于 MAC 地址生成的用户名和密码。如果授权成功，交换机会允许客户端访问网络。如果授权失败，交换机会把端口分配到指定的访客 VLAN 中。

使用受限 VLAN 的 802.1x 认证

可以为每个交换机堆栈或者交换机的 IEEE 802.1x 端口配置受限 VLAN（也称为*认证失败* VLAN），给不能访问访客 VLAN 的客户端提供有限的服务。这些客户端兼容 802.1x，但因为认证失败而不能访问其他的 VLAN。受限 VLAN 允许在认证服务器上没有合法凭据的用户（通常是企业的访客）访问有限的服务。管理员可以控制对受限 VLAN 可用的服务。

注释： 如果希望给访客 VLAN 的用户以及受限 VLAN 的用户提供相同的服务，可以配置一个 VLAN 同时作为两种 VLAN 使用。

不使用此特性时，客户端会无限次地尝试认证并失败，而交换机端口会保持在生成树的阻塞

状态。使用此特性时，可以让交换机端口在指定次数的认证尝试（默认值是 3 次）之后进入受限 VLAN 中。

认证程序会记录客户端认证失败的次数。当次数超过了配置的最大尝试次数，端口会被移动至受限 VLAN 中。当 RADIUS 服务器回复了 *EAP 失败*包或者不使用 EAP 包的空应答时，失败尝试计数会增加。当端口移动至受限 VLAN 时，失败尝试计数重置。

认证失败的用户会保持在受限 VLAN 中，直到下一次重新尝试认证。受限 VLAN 中的端口会按照配置的间隔（默认为 60 秒）重新尝试认证。如果重新认证失败，端口会保留在受限 VLAN 中。如果重新认证成功，端口会被移动到配置的 VLAN 或者 RADIUS 服务器发来的 VLAN 中。

可以禁用重新认证功能。如果执行了此操作，重启认证过程的唯一方式是在端口上接收到*链路 down* 或 *EAP 登出*事件。建议在客户端可能通过集线器连接的情况下保持重新认证功能启用。因为当客户端断开到集线器的连接时，端口可能无法收到*链路 down* 或 *EAP 登出*事件。

在端口移动到受限 VLAN 之后，交换机会给客户端发送一个假的 EAP 成功消息。此行为会防止客户端无限期地尝试认真就。一些客户端（如运行 Windows XP 的设备）收不到 EAP 成功消息就无法进行 DHCP 的操作。

受限 VLAN 在所有主机模式的 802.1x 端口以及二层端口上支持。

可以把除了 RSPAN VLAN、主私有 VLAN 或语音 VLAN 之外的任意活跃 VLAN 配置为 802.1x 受限 VLAN。受限 VLAN 特性不被内部 VLAN（被路由端口）或中继端口支持，仅被接入端口支持。

其他的端口安全特性，如动态 ARP 监测、DHCP 侦听以及 IP 源防护，可以在受限 VLAN 独立配置。

使用不可访问旁路认证的 802.1x 认证

当交换机无法连通配置的 RADIUS 服务器且新主机无法被认证时，可以使用不可访问旁路认证特性（也称*临界认证*或 *AAA 失败策略*）。可以配置交换机把这些主机连接到临界端口。

当新主机尝试连接到临界端口时，该主机会被移动至用户特定的接入 VLAN，即*临界 VLAN* 中。管理员可以给这些主机授予有限的认证。

当交换机尝试认证连接到临界端口的主机时，交换机会检查配置的 RADIUS 服务器状态。如果服务器可用，交换机就可以认证主机。然而，如果所有的 RADIUS 服务器都不可用，交换机会授予主机网络访问权限，并把交换机端口置入*临界认证*状态中，这是认证状态的一种特殊情况。

注释： 如果在接口上配置了临界认证，交换机用于临界授权的 VLAN（*临界 VLAN*）应该是

活跃的。如果*临界 VLAN* 的状态为不活跃或 down，*临界认证*会话会不断尝试启用不活跃的 VLAN 并一直失败。这可能导致大量的内存占用。

多认证端口对不可访问旁路认证的支持

当端口被配置在任意的主机模式且AAA服务器不可用时，端口会被配置为多主机模式，并被移动至临界VLAN中。要在多认证模式的端口上支持不可访问旁路认证，可以使用 **authentication event server dead action reinitialize vlan** *vlan-id*命令。当有新主机尝试连接临界端口时，该端口会被重新初始化，所有连接的主机都会被移动到用户指定的接入VLAN中。此命令在所有主机模式中都支持。

不可访问旁路认证的认证结果

不可访问旁路认证特性的行为取决于端口的授权状态：

- 当连接到临界端口的主机尝试进行认证而所有服务器都不可用时，如果端口是未授权状态，交换机会把端口置为临界认证状态，并放在 RADIUS 配置的或用户指定的接入 VLAN 中；
- 如果端口已经是授权状态且发生了重新认证，交换机会把临界端口置为临界认证状态并放在当前 VLAN 中，此 VLAN 可能是之前由 RADIUS 服务器指定的 VLAN；
- 如果在认证交换期间 RADIUS 服务器变为不可用状态，当前的交换过程会超时，交换机会在进行下一次认证尝试时把临界端口置为临界认证状态。

可以配置临界端口在 RADIUS 服务器重新可用时重新初始化主机，并把它们从临界 VLAN 中移出。配置此操作时，所有在临界认证状态中的临界端口都会自动重新进行认证。

不可访问旁路认证的特性相互影响

不可访问旁路认证会与以下特性相互影响：

- 访客 VLAN——不可访问旁路认证兼容访客 VLAN。在 802.1x 端口上启用访客 VLAN 时，特性间的相互作用如下：
  - 如果至少有一台 RADIUS 服务器可用，在交换机没有收到对其发送的 EAP 请求/身份帧的应答或者客户端没有发送 EAPOL 包时，交换机会把客户端分配到访客 VLAN 中；
  - 如果所有的 RADIUS 服务器都不可用且客户端连接到临界端口，交换机会认证客户

端，把临界端口置于临界认证状态，并放在 RADIUS 配置或用户指定的接入 VLAN 中；

- 如果所有的 RADIUS 服务器都不可用且客户端未连接到临界端口，交换机可能不会把客户端分配到访客 VLAN 中；
- 如果所有的 RADIUS 服务器都不可用且客户端连接到之前分配到访客 VLAN 的临界端口，交换机会把端口保留在访客 VLAN 中。

- 受限 VLAN——如果 RADIUS 服务器不可用且端口已经被授权在受限 VLAN 中，交换机会把临界端口置为临界认证状态，并放在受限 VLAN 中；
- 802.1x 审计——如果 RADIUS 服务器不可用，审计不受影响；
- 私有 VLAN——可以在私有 VLAN 主机端口上配置不可访问旁路认证。接入 VLAN 必须是次级私有 VLAN；
- 语音 VLAN——不可访问旁路认证与语音 VLAN 兼容，但是 RADIUS 配置的或用户指定的接入 VLAN 必须与语音 VLAN 不同；
- 远程交换端口分析器（Remote Switched Port Analyzer，RSPAN）——不要把 RSPAN VLAN 配置为 RADIUS 或用户为不可访问旁路认证配置的接入 VLAN。

在交换机堆栈中：

- 堆栈主用设备会通过发送保活包检查 RADIUS 服务器的状态。当 RADIUS 服务器的状态改变时，堆栈主用设备会把此信息发送给堆栈成员。堆栈成员可以在重新认证临界端口的时候检查 RADIUS 服务器的状态；
- 如果选举出了新的堆栈主用设备，交换机堆栈与 RADIUS 服务器之间的链路可能改变，新的堆栈主用设备会立即发送保活包来更新 RADIUS 服务器的状态。如果服务器的状态从 *dead* 变为 *alive*，交换机会重新认证所有临界认证状态的端口。

当成员添加到堆栈时，堆栈主用设备会给成员发送服务器的状态。

注释： 交换机堆栈只在运行 LAN Base 镜像的 Inspur 2960X 交换机上支持。

802.1x 临界语音 VLAN

当一台 IP 电话连接到一个已被访问控制服务器（access control server，ACS）认证的端口上，IP 电话会被放在语音域中。如果 ACS 不可达，交换机不能确定设备是否是语音设备，电话也就不能访问语音网络，进而也不能工作。

对于数据流量，可以配置不可访问旁路认证或临界认证，当服务器不可达的时候允许流量通过本征 VLAN。如果 RADIUS 服务器不可用且启用了不可用旁路认证，交换机会授予客户端访问网络的权利，并把临界认证状态的端口放在 RADIUS 配置或者用户指定的接入 VLAN 中。

当交换机不能连通配置的 RADIUS 服务器时，新的主机不能被认证，交换机会把这些主机连接到临界端口上。尝试连接临界端口的新主机会被移动到用户指定的接入 VLAN（临界 VLAN）中，并被授予有限的认证权限。

可以输入接口配置命令 **authentication event server dead action authorize voice** 来配置临界语音 VLAN 特性。当 ACS 不响应时，端口会进入临界认证模式。当来自主机的流量打了语音 VLAN 的标签时，连接的设备（IP 电话）会被置入为端口配置的语音 VLAN 中。IP 电话会通过 CDP（Inspur 设备）、LLDP 或 DHCP 学习语音 VLAN 的标识信息。

可以输入接口配置命令 **switchport voice vlan** *vlan-id* 为端口配置语音 VLAN。

此特性在多域及多认证主机模式上支持。虽然也可以在单主机或多主机模式的交换机上输入此命令，但是除非交换机改为多域或多认证主机模式，否则命令不会生效。

802.1x 用户分配

可以配置 802.1x 用户分配特性，把群组名相同用户在多个不同的 VLAN 上进行负载均分。这些 VLAN 可以由 RADIUS 服务器提供，也可以通过交换机的 CLI 配置在一个 VLAN 群组名下：

- 配置 RADIUS 服务器为用户发送多个 VLAN 名称。多个 VLAN 名称可以作为发往用户的应答的一部分。802.1x 用户分配特性会追踪特定 VLAN 中的所有用户，并把已授权的用户移动到流量最少的 VLAN 中。
- 配置 RADIUS 服务器为用户发送一个 VLAN 群组名。VLAN 群组名可以作为发往用户的应答的一部分。可以使用交换机的 CLI 查询在配置的 VLAN 群组名中哪个群组名被选用。如果交换机找到了这样的 VLAN 群组名，就会查询这个 VLAN 群组名下的流量最少的 VLAN。负载均分通过把对应的已授权用户移动到这个 VLAN 实现。

注释： RADIUS 服务器发送的 VLAN 信息可以是任意 VLAN ID、VLAN 名称及 VLAN 群组的组合。

802.1x 用户分配配置指南

- 确认至少有一个 VLAN 映射到了 VLAN 群组；
- 可以把多个 VLAN 映射到一个 VLAN 群组中；
- 可以添加或删除 VLAN 群组中的 VLAN；
- 当管理员清除一个 VLAN 群组中已有的 VLAN 时，该 VLAN 中已认证的端口不会被清除，但映射会被从现有的 VLAN 群组中移除；
- 如果清除了 VLAN 群组中的最后一个 VLAN，VLAN 群组也会被清除；

- 即使有活跃的 VLAN 映射到 VLAN 群组，也可以清除该群组。清除群组时，群组内任意 VLAN 中已认证状态的端口或用户都不会被清除，但是 VLAN 到 VLAN 群组的映射会被清除。

语音 VLAN 端口与 IEEE 802.1x 认证

一个语音 VLAN 端口是特殊的接入端口，它关联了两个 VLAN 标识符：

- VVID，承载 IP 电话收发的语音流量。VVID 被用于配置连接到端口的 IP 电话；
- PVID，承载通过 IP 电话连接到交换机的工作站收发的数据流量。PVID 是端口的本征 VLAN。

无论端口授权状态如何，IP 电话都会使用 VVID 传输其语音流量。这使得电话可以独立于 IEEE 802.1x 认证工作。

在单主机模式中，语音 VLAN 上只允许有 IP 电话。在多主机模式中，请求者在 PVID 上认证后其他客户端可以在语音 VLAN 上发送流量。启用多主机模式时，请求者的认证会同时影响 PVID 以及 VVID。

当存在链路，且在来自 IP 电话的第一个 CDP 消息后出现了设备的 MAC 地址时，语音 VLAN 端口变为活跃状态。Inspur IP 电话不会中继来自其他设备的 CDP 包。因此，如果有多台 IP 电话串联在一起，交换机只能识别直接相连的一台。在语音 VLAN 的端口上启用 IEEE 802.1x 认证后，交换机会丢弃来自一跳之外的未识别 IP 电话发来的数据包。

在交换机端口上启用 IEEE 802.1x 认证时，可以把接入端口的 VLAN 同时配置为语音 VLAN。当 IP 电话连接到单主机模式的 802.1x 交换机端口时，交换机无需认证 IP 电话就会授予其网络访问的权利。建议在既认证数据设备也认证语音设备的端口上使用多域认证（MDA）。

注释： 如果在接入端口上启用了 IEEE 802.1x 认证，且该端口已经配置了语音 VLAN 并有 Inspur IP 点相连，Inspur IP 电话会失去与交换机的连通性至多 30 秒。

端口安全与 IEEE 802.1x 认证

通常来说，Inspur 不建议在启用了 IEEE 802.1x 的端口上启用端口安全特性。因为 IEEE 802.1x 强制一个端口只有一个 MAC 地址（为 IP 电话配置 MDA 时，强制一个 VLAN 只有一个 MAC 地址），端口安全特性就冗余了，而且有时还可能干扰 IEEE 802.1x 的操作。

LAN 唤醒与 IEEE 802.1x 认证

使用 IEEE 802.1x 认证以及 LAN 唤醒（wake-on-LAN，WoL）特性，可以在交换机收到特定的

以太网帧时启动休眠的主机，这样的数据帧也被称为*魔力包（magic packet）*。可以在管理员需要连接到已经关机的系统时使用此特性。

当一台使用 WoL 的主机连接到 IEEE 802.1x 端口上，且主机已关机，IEEE 802.1x 端口变为未授权状态。这样的端口只能收发 EAPOL 包，所以 WoL 的魔力包就不能到达主机。PC 关机，不被授权，则交换机端口不开放。

当交换机使用 IEEE 802.1x 认证以及 WoL 时，交换机会向未授权端口转发流量，其中就包含魔力包。因为端口仍是未授权状态，交换机会继续阻塞除了 EAPOL 包之外的入向流量。这时的主机可以接收数据包，但不能向网络上的其他设备发送数据包。

注释： 如果在端口上启用了 PortFast，端口强制为双向状态。

使用接口配置命令 **authentication control-direction in** 把端口配置为单向时，端口会变为生成树的转发状态。该端口可以给主机发送数据包，但是不能接收来自主机的数据包。

使用接口配置命令 **authentication control-direction both** 把端口配置为双向时，端口的两个方向都会进行访问控制。该端口不会向主机收发数据包。

MAC 旁路认证与 IEEE 802.1x 认证

可以配置交换机使用 MAC 旁路认证特性，让交换机基于客户端的 MAC 地址进行授权。例如，可以在连接了打印机的 IEEE 802.1x 端口上启用此特性。

如果 IEEE 802.1x 认证等待客户端 EAPOL 应答超时，交换机会尝试使用 MAC 旁路认证特性授权客户端。

在启用了 IEEE 802.1x 的端口上启用 MAC 旁路认证特性时，交换机会把 MAC 地址作为客户端的身份。认证服务器上的数据库中有允许访问网络的客户端的 MAC 地址。在 IEEE 802.1x 端口上检测到客户端之后，交换机会等待来自客户端的以太网数据包。交换机会给认证服务器发送一个 RADIUS 访问/请求帧，带有基于 MAC 地址生成的用户名和密码。如果授权成功，交换机会授予客户端访问网络的权限。如果授权失败，交换机会把端口分配给配置的访客 VLAN。此过程适用于多数客户端设备，但不适用于其他 MAC 地址格式的客户端。当客户端 MAC 地址与标准格式不同，或者 RADIUS 配置要求用户名与密码不同时，可以配置使用 MAB 认证。

如果在链路的生存时间内在接口上检测到了 EAPOL 包，交换机可以确定连接到接口上的设备兼容 802.1x，就会使用 802.1x 认证方式（而不是 MAC 旁路认证）来授权接口。如果接口的链路状态变为 down，EAPOL 历史会被清除。

交换机已经使用 MAC 旁路认证授权了一个端口，如果此时检测到了 IEEE 802.1x 认证请求者，交换机不会授权连接到端口的客户端。重新认证发生时，如果之前的会话因为终止操作

RADIUS 属性的值是 DEFAULT 而结束，交换机会使用端口配置的认证或重新认证方式执行操作。

使用 MAC 旁路认证方式授权的客户端可以被重新认证。该客户端的重新认证过程与使用 IEEE 802.1x 认证的客户端相同。在重新认证期间，端口会保持在之前分配的 VLAN 中。如果重新认证成功，交换机会把端口保留在相同 VLAN 中。如果重新认证失败，交换机会把端口分配给配置的访客 VLAN。

如果重新认证基于会话超时 RADIUS 属性（属性[27]）以及终止操作 RADIUS 属性（属性[29]），而且终止操作 RADIUS 属性的操作是*初始化（Initialize）*（属性默认值为 *DEFAULT*），MAC 旁路认证的会话将结束，且重新认证期间的连通性会丢失。如果启用了 MAC 旁路认证且 IEEE 802.1x 认证超时，交换机会使用 MAC 旁路认证特性来初始化重新认证过程。更多有关 AV 对的信息，参见 RFC 3580“IEEE 802.1X 远程验证拨入用户服务（RemoteAuthentication Dial In User Service，RADIUS）使用指南”。

MAC 旁路认证会与以下特性相互影响：

- IEEE 802.1x 认证——只有在端口上启用 802.1x 认证时才可以启用 MAC 旁路认证
- 访客 VLAN——如果客户端 MAC 地址身份非法，交换机会把客户端分配到配置的访客 VLAN 中
- 受限 VLAN——当连接到 IEEE 802.1x 端口的客户端使用 MAC 旁路认证时，此特性不被支持
- 端口安全
- 语音 VLAN
- 私有 VLAN——可以将客户端分配给私有 VLAN。
- 网络边缘接入拓扑（Network Edge Access Topology，NEAT）——MAB 和 NEAT 特性是互斥的。在接口上启用 NEAT 时不能启用 MAB，反之亦然

Inspur INOS 12.2（55）SE 以及之后版本支持过滤详细的 MAB 系统消息。

网络接入控制二层 IEEE 802.1x 验证

交换机支持网络接入控制（Network Admission Control，NAC）二层 IEEE 802.1x 验证特性，会在授予设备网络访问权限之前检查终端系统或客户端的防病毒状态或*态势（posture）*。使用 NAC 二层 IEEE 802.1x 验证时，可以执行以下操作：

- 从认证服务器上下载会话超时 RADIUS 属性（属性[27]）以及终止操作 RADIUS 属性（属性[29]）；
- 把进行重新认证尝试之间的秒数设置为会话超时 RADIUS 属性（属性[27]）值，并通过

RADIUS 服务器获取客户端的访问策略；

- 使用终止操作 RADIUS 属性（属性[29]）设置交换机尝试重新认证客户端时采取的操作。如果此值为 *DEFAULT* 或未设置，重新认证时会话结束。如果值是 RADIUS 请求，重新认证过程开始；
- 把 VLAN 编号或名称的列表、VLAN 群组名称设置为隧道组私有 ID（属性[81]）的值，并让隧道偏好（属性[83]）值使用这些 VLAN。如果不配置隧道偏好，首个隧道组私有 ID（属性[81]）会从列表中选取；
- 使用特权 EXEC 命令 **show authentication** 查看客户端的 NAC 态势令牌，获知客户端的态势；
- 把次级私有 VLAN 配置为访客 VLAN。

配置 NAC 二层 IEEE 802.1x 验证的过程与配置 IEEE 802.1x 基于端口认证的过程相似，除了必须要在 RADIUS 服务器上配置态势令牌。

灵活的认证顺序

可以使用灵活认证顺序功能配置端口认证新主机时采用的方法的顺序。IEEE 802.1x 灵活认证（Flexible Authentication）特性支持三种认证方式：

- dot1X——IEEE 802.1x 认证时二层认证方式
- mab——MAC 旁路认证时二层认证方式
- webauth——网页认证是三层认证方式

使用此特性时，可以控制哪些端口使用哪些认证方式，而且可以控制这些端口上认证方式故障转移的顺序。例如，MAC 旁路认证以及 802.1x 可以是认证的主要方式或次要方式，如果尝试这些认证方式失败，网页认证可以作为备用方式使用。

IEEE 802.1x 灵活认证特性支持以下主机模式：

- 多认证——多认证允许在一个语音 VLAN 上进行一次认证，在数据 VLAN 上进行多次认证
- 多域认证——多域认证允许进行两次认证：一次在语音 VLAN 上，一次在数据 VLAN 上

Open1x 认证

Open1x 认证允许设备在被认证之前访问端口。配置开放认证时，新主机可以根据端口上定义的访问控制列表（ACL）传输流量。主机被认证之后，在 RADIUS 服务器上配置的策略会

应用给主机。

可以在以下场景中配置开放认证：

- 单主机模式——认证前后只允许一个用户访问网络
- MDA 模式——只允许语音域中有一个用户，数据域中有一个用户
- 多认证模式——与 MDA 相似，但可以认证多台主机

注释： 如果配置了开放认证，该方式优先于其他的认证控制特性。这意味着如果使用了接口配置命令**authentication open**，无论接口配置命令**authentication port-control**配置如何，端口都会允许主机访问。

多域认证

交换机支持进行多域认证（multidomain authentication，MDA），允许在一个交换机端口上同时认证一台数据设备和一台语音设备（如 Inspur 或非 Inspur 的 IP 电话）。端口被分为一个数据域和一个语音域。

注释： 对于所有的主机模式，配置基于端口的认证时，认证之前线路协议保持 up 状态。MDA 不强制设备认证的顺序。然而在启用 MDA 的端口上，建议在认证数据设备之前认证语音设备。

按照以下指南配置 MDA：

- 必须把交换机端口配置为 MDA；
- 当主机模式设置为多域时，必须配置 IP 电话使用的语音 VLAN；
- 启用 MDA 端口上的语音 VLAN 分配功能在 Inspur INOS 12.2（40）SE 及之后版本上支持；
- 如需认证语音设备，必须配置 AAA 服务器发送属性值为 *device-traffic-class=voice* 的 Inspur 属性值（AV）对。否则，交换机会把语音设备当作数据设备；
- 访客 VLAN 和受限 VLAN 特性只适用于 MDA 端口上的数据设备。交换机会把授权失败的语音设备当作数据设备；
- 如果有多台设备尝试在端口的语音域或数据域上进行授权，端口会被错误禁用；
- 在设备被授权之前，端口会丢弃其流量。允许在语音以及数据 VLAN 中使用非 Inspur 的 IP 电话或语音设备。数据 VLAN 允许该设备联系 DHCP 服务器，获取 IP 地址以及语音 VLAN 的信息。当语音设备开始在语音 VLAN 上发送数据后，其对数据 VLAN 的访问将被阻止；
- 对于端口安全特性的 MAC 地址数量限制，绑定到语音 VLAN 上的语音设备的 MAC 地址不会被计数；
- MDA 可以使用 MAC 旁路认证作为备用的认证机制，使不支持 IEEE 802.1x 认证的设备可

以连通交换机端口；

- 在端口上检测到一个*数据*设备或*语音*设备时，在认证成功前其 MAC 地址会被阻塞。如果认证失败，该 MAC 地址会保持阻塞 5 分钟；
- 未授权的端口上，如果在*数据* VLAN 中检测到了超过五台设备，或者在*语音* VLAN 上检测到超过一台设备，该端口会错误禁用；
- 当端口主机模式从单主机或多主机变为多域模式时，端口上已授权的数据设备仍保持授权状态。然而，端口语音 VLAN 已经允许的 Inspur IP 电话会被自动移除，且必须在端口上进行重新认证；
- 当端口主机模式从单主机或多主机变为多域模式时，如访客和受限 VLAN 这样的主动回退机制配置保持不变；
- 把端口主机模式从多域模式改为单主机或多主机模式会移除端口上所有已授权的设备；
- 如果数据域先被认证，且被置入访客 VLAN 中，不兼容 IEEE 802.1x 的语音设备需要把自己的数据包标记为语音 VLAN 数据包才能触发认证过程；
- 不建议在启用 MDA 的端口上使用基于用户的 ACL。使用基于用户 ACL 策略的已授权设备可能会同时影响端口上的语音 VLAN 和数据 VLAN。若使用这样的配置，只应给端口上的一台设备执行基于用户的 ACL。

802.1x 请求交换机、认证交换机以及网络边缘接入拓扑（NEAT）

网络边缘接入拓扑（Network Edge Access Topology，NEAT）特性把身份认证扩展到配线间之外的区域（比如会议室）。

- 802.1x 请求交换机：可以使用 802.1x 请求者特性，配置一台交换机作为另一台交换机的请求者。在配线间外的交换机通过中继端口连接到上行交换机这类场景中，此配置很有用。配置了 802.1x 请求者特性的交换机会与上行交换机认证以进行安全连接。当请求交换机认证成功时，认证交换机上的端口模式会从接入变为中继。启用 CISP 时，必须在请求交换机上手动配置中继；
- 如果在认证交换机上配置了接入 VLAN，该 VLAN 在成功认证之后会成为中继端口的本征 VLAN。

默认状态下，如果请求交换机连接到了一台启用了BPDU防护的认证交换机，且认证交换机的端口在请求交换机被认证之前收到了生成树协议（Spanning Tree Protocol，STP）的网桥协议数据单元（BridgeProtocol Data Unit，BPDU）数据包，该端口可能被错误禁用。从Inspur INOS 15.0(1) SE版开始，可以控制认证期间从请求端口上发出的流量。输入全局配置命令**dot1x supplicant controlled transient**会在认证期间临时阻塞请求交换机的端口，以保证认证交换机

端口不会在认证完成之前被关闭。如果认证失败，请求交换机端口会开放。输入全局配置命令**no dot1x supplicant controlledtransient**可以在认证期间打开请求交换机端口。这是交换机的默认行为。

当认证交换机上通过接口配置命令**spanning-tree bpduguard enable**启用了BPDU防护时，强烈建议在请求交换机上使用**dot1x supplicant controlled transient**命令。

注释： 如果在认证交换机上使用全局配置命令**spanning-tree portfastbpduguard default**启用了BPDU防护，输入**dot1x supplicant controlled transient**命令不能防止BPDU违规发生。

可以在认证交换机连接到多台请求交换机的接口上启用 MDA 或多认证模式。认证交换机接口不支持多主机模式。

当重启接口上使用单主机模式的认证交换机时，该接口可能在认证前变为错误禁用状态。要从错误禁用状态恢复，请重启认证交换机的接口以激活接口并发起认证过程。

要让网络边缘接入拓扑（NEAT）特性在所有主机模式中都能工作，请在请求交换机上配置全局配置命令 **dot1x supplicant force-multicast**。

- 主机授权：确保网络上只允许通过被授权主机（连接到请求交换机）的流量。交换机使用客户端信息信令协议（Client Information Signalling Protocol，CISP）把连接到请求交换机的 MAC 地址发送给认证交换机；
- 自动启用：在认证交换机上自动启用中继配置，允许传输来自请求交换机上多个 VLAN 的用户流量。请在 ACS 上把 inspur-av-pair 配置为 *device-traffic-class=switch*（可以在*组*或*用户*设置下配置此项）

图 124：使用 CISP 的认证交换机及请求交换机

Workstations (clients)	工作站（客户端）
Supplicant switch (outside wiring closet)	请求交换机（在配线间外）
Authenticator switch	认证交换机
Access control server (ACS)	访问控制服务器（ACS）
Trunk port	中继端口

注释： 使用 NEAT 的请求交换机和认证交换机上不支持 **switchport nonegotiate** 命令。此命令不应配置在拓扑的请求端。如果配置在认证端，交换机的内部宏会自动把此命令从端口移除。

语音感知的 802.1x 安全特性

注释： 要使用语音感知 IEEE 802.1x 认证功能，交换机必须运行 LAN Base 的镜像。

可以使用语音感知的 802.1x 安全特性，配置交换机在数据 VLAN 或语音 VLAN 安全违规事件发生时只禁用相关 VLAN。之前，当尝试认证的数据客户端造成了安全违规事件时，整个端口会被关闭，到了连通性完全丢失。

当 PC 连接到 IP 电话时，可以使用此特性。数据 VLAN 上的安全违规事件只会导致数据 VLAN 被关闭。经过交换机传输的语音 VLAN 流量不会被干扰。

通用会话 ID

无论采取何种认证方式，认证管理器都会对客户端使用一个会话 ID（称为通用会话 ID）。此 ID 被用于所有的报告功能，如 show 命令以及 MIB。此会话 ID 出现在会话前的 syslog 消息中。

会话 ID 包含：

- 网络接入设备（Network Access Device，NAD）的 IP 地址
- 一个单调递增的唯一的 32 位整数
- 会话开始时间戳（一个 32 位整数）

以下示例显示了命令 show authentication 输出中会话 ID 的显示方式。此例中的会话 ID 是 160000050000000B288508E5：

```
Device# show authentication sessions
Interface MAC Address Method Domain Status Session ID
Fa4/0/4 0000.0000.0203 mab DATA Authz Success 160000050000000B288508E5
```

以下是 syslog 输出中会话 ID 的显示方式。此例中的会话 ID 是 160000050000000B288508E5：

```
1w0d: %AUTHMGR-5-START: Starting 'mab' for client (0000.0000.0203) on Interface Fa4/0/4
AuditSessionID 160000050000000B288508E5
1w0d: %MAB-5-SUCCESS: Authentication successful for client (0000.0000.0203) on Interface
Fa4/0/4 AuditSessionID 160000050000000B288508E5
1w0d: %AUTHMGR-7-RESULT: Authentication result 'success' from 'mab' for client
(0000.0000.0203) on Interface Fa4/0/4 AuditSessionID 160000050000000B288508E5
```

此会话 ID 被 NAD、AAA 服务器以及其他汇报分析程序用来标识客户端。此 ID 自动生成，无需配置。

如何配置 802.1x 基于端口的认证

默认的 802.1x 认证配置

表 148：默认的 802.1x 认证配置

特性	默认设置
交换机上 802.1x 的启用状态	禁用
每个端口的 802.1x 启用状态	禁用（强制授权）。 端口会收发正常流量，无需对客户端进行基于 802.1x 的认证
AAA	禁用
RADIUS 服务器 • IP 地址 • UDP 认证端口 • 默认审计端口 • 秘钥	• 未指定 • 1645 • 1646 • 未指定
主机模式	单主机模式
控制方向	双向控制
周期性重新认证	禁用
尝试重新认证的时间间隔	3600 秒
重新认证次数	2 次（在端口变为未授权状态之前交换机重启认证过程的次数）
静默时长	60 秒（在与客户端认证交换失败之后交换机保持静默状态的秒数）
重传时间	30 秒（交换机在重新发送 EAP 请求/身份帧之前等待客户端应答的秒数）
最大重传次数	2 次（交换机在重启认证过程之前发送 EAP 请求/身份帧的次数）
客户端超时时间	30 秒（在中继认证服务器到客户端的请求时，交换机重发请求给客户端之前等待的时间）

认证服务器超时时间	30 秒（在中继客户端给认证服务器的应答时，交换机重发应答给服务器之前等待的时间）
不活跃超时	禁用
访客 VLAN	未指定
不可访问旁路认证	禁用
受限 VLAN	未指定
认证交换机模式	未指定
MAC 旁路认证	禁用
语音感知安全	禁用

802.1x 认证配置指南

802.1x 认证

以下是 802.1x 配置指南：

- 启用 802.1x 认证时，端口在任何其他二层或三层特性启用之前被认证；
- 如果启用了 802.1x 的端口分配的 VLAN 变化，变化对交换机透明且不会影响交换机配置。比如，端口可能分配给了 RADIUS 服务器指定的 VLAN，而重新认证之后又分配给了不同的 VLAN，此变化对交换机透明；
- 如果 802.1x 端口分配的 VLAN 关闭、禁用或被移除，端口会变为未授权状态。比如，端口分配的接入 VLAN 被关闭或移除之后，端口变为未授权；
- 802.1x 协议支持二层静态接入端口、语音 VLAN 端口以及三层被路由端口，但不支持以下端口类型：
  - 动态端口——动态模式中的端口可以与邻居协商并成为中继端口。如果尝试在动态端口上启用 802.1x 认证，会出现错误消息，且 802.1x 认证不会被启用。如果尝试把启用了 802.1x 的端口更改为动态模式，会出现错误消息，且端口模式不会变化；
  - EtherChannel 端口——不要把 EtherChannel 中的活跃成员或尚未活跃的成员配置为 802.1x 端口。如果尝试在一个 EtherChannel 端口上启用 802.1x 认证，会出现错误消息，且 802.1x 认证不会被启用；
  - 交换端口分析器（SPAN）和远程 SPAN（RSPAN）目的端口——可以在 SPAN 或 RSPAN 目的端口上启用 802.1x 认证。然而，直到端口从 SPAN 或 RSPAN 目的端口中移除，

802.1x 认证才会启用。可以在 SPAN 或 RSPAN 源端口上启用 802.1x 认证。

- 在交换机上使用全局配置命令全局启用 802.1x 认证之前，应移除同时配置了 802.1x 认证和 EtherChannel 的端口上的 EtherChannel 配置；
- Inspur INOS 12.2(55)SE 以及之后版本支持过滤与 802.1x 认证相关的系统消息。

VLAN 分配、访客 VLAN、受限 VLAN 以及不可访问旁路认证

以下是 VLAN 分配、访客 VLAN、受限 VLAN 以及不可访问旁路认证的配置指南：

- 在端口上启用 802.1x 认证时，不能把端口 VLAN 配置为语音 VLAN；
- 使用 VLAN 分配的 802.1x 认证特性不支持中继端口、动态端口或是使用 VMPS 分配的动态端口；
- 可以把除了 RSPAN VLAN 或语音 VLAN 之外的任意 VLAN 配置为 802.1x 访客 VLAN。访客 VLAN 特性不支持内部 VLAN（被路由端口）或中继端口，只支持接入端口；
- 为连接了 DHCP 客户端的 802.1x 端口配置访客 VLAN 时，客户端可能需要通过 DHCP 服务器获取主机 IP 地址。可以更改设置，在客户端上的 DHCP 进程超时并尝试从 DHCP 服务器获取 IP 地址之前，让交换机重启 802.1x 认证过程。可以减少对 802.1x 认证过程的设置（接口配置命令 authentication timer inactivity 和 authentication timer reauthentication）。需减少的设置数量取决于连接的 802.1x 设备类型；
- 按照以下指南配置不可访问旁路认证特性：
  - 此特性支持单主机模式和多主机模式的 802.1x 端口；
  - 如果客户端运行 Windows XP，且客户端连接到的端口在临界认证状态中，Windows XP 可能报告接口未被认证；
  - 如果 Windows XP 客户端配置进行 DHCP 且拥有来自 DHCP 服务器的 IP 地址，在临界端口上接收 EAP 成功消息可能不会重启 DHCP 配置过程；
  - 可以在 802.1x 端口上配置不可访问旁路特性以及受限 VLAN。如果交换机尝试重新认证受限 VLAN 中的临界端口，且所有 RADIUS 服务器都不可用，交换机会把端口状态改为临界认证状态，并保持在受限 VLAN 中；
  - 如果 CTS 链路在临界认证模式中且主用设备重启，SGT 在设备上配置的策略在新主用设备上不可用。这是因为 3750-X 交换机堆栈中的内部绑定不会被同步到备用交换机上。
- 可以把除了 RSPAN VLAN 或语音 VLAN 之外的任意 VLAN 配置为 802.1x 受限 VLAN。受限 VLAN 特性不支持内部 VLAN（被路由端口）或中继端口，只支持接入端口。

MAC 旁路认证

802.1x 端口上允许的最大设备数量如下：

- 在单主机模式中，接入 VLAN 上只允许有一台设备。如果端口也配置了语音 VLAN，无限数量的 Inspur IP 电话可以通过语音 VLAN 收发流量；
- 在多域认证（MDA）模式中，接入 VLAN 允许有一台设备，语音 VLAN 允许有一台设备；
- 在多主机模式中，端口上只允许有一个 802.1x 请求者，但是接入 VLAN 中允许有无限数量的非 802.1x 主机。语音 VLAN 允许有无限数量的设备。

配置 802.1x 就绪状态检查

802.1x 就绪状态检查特性会监控交换机所有端口上的 802.1x 活动，并显示端口连接的支持 802.1x 的设备信息。可以使用此特性确定连接到交换机端口的设备是否兼容 802.1x。

802.1x 就绪状态检查允许在配置 802.1x 的所有端口上使用。就绪状态检查在配置为 **dot1x force-unauthorized** 的端口上不可用。

按照以下步骤在交换机上启用 802.1x 就绪状态检查：

在开始前

以下是启用就绪状态检查的指南：

就绪状态检查通常在 802.1x 启用之前使用。

如果使用特权 EXEC 命令 **dot1x test eapol-capable** 且没有指定接口，交换机堆栈的所有端口都会被测试。

如果在启用 802.1x 的端口上配置了 the **dot1x test eapol-capable** 命令，链路启用时，端口会查询连接客户端的 802.1x 兼容性。客户端使用通知包应答，则其兼容 802.1x。如果客户端在超时时间内进行响应，交换机会产生 syslog 消息。如果客户端没有响应查询消息，则其不兼容 802.1x，此时不会产生 syslog 消息。

可以在处理多台主机的端口上（比如 PC 通过 IP 电话连接端口）发送就绪状态检查消息。对于每一个在超时时间内响应的客户端，交换机都会产生一条 syslog 消息。

总步骤

1. enable

2. configure terminal

3. dot1x test eapol-capable [**interface** *interface-id*]

4. dot1x test timeout *timeout*

5. end

6. show running-config

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式
步骤 3	**dot1x test eapol-capable** [**interface** *interface-id*] **示例:** `Device#` **dot1x test eapol-capable interface gigabitethernet1/0/13** `DOT1X_PORT_EAPOL_CAPABLE:DOT1X: MAC 00-01-02-4b-f1-a3 on gigabitethernet1/0/13 is EAPOL capable`	在交换机上启用就绪状态检查， （可选）使用 *interface-id* 指定检查哪个端口的 IEEE 802.1x 就绪状态。 **注释：**如果省略可选的 **interface** 关键字，交换机上的所有端口都会被测试
步骤 4	**dot1x test timeout** *timeout*	（可选）配置等待 EAPOL 应答的超时时间。范围从 1 到 65535 秒，默认值是 10 秒
步骤 5	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式
步骤 6	**show running-config** **示例：** `Device#` **show running-config**	验证配置的条目
步骤 7	**copy running-config startup-config** **示例：**	（可选）把配置保存在配置文件中

	Device# **copy running-config startup-config**	

配置语音感知的 802.1x 安全特性

注释： 要使用语音感知 IEEE 802.1x 认证功能，交换机必须运行 LAN Base 的镜像。

可以使用语音感知的 802.1x 安全特性，配置交换机在数据 VLAN 或语音 VLAN 安全违规事件发生时只禁用相关 VLAN。当 PC 连接到 IP 电话时，可以使用此特性。数据 VLAN 上的安全违规事件只会导致数据 VLAN 被关闭。经过交换机传输的语音 VLAN 流量不会被干扰。

按照以下指南在交换机上配置语音感知 802.1x 安全特性：

- 输入全局配置命令**errdisable detect cause security-violationshutdown vlan**启用语音感知802.1x安全特性。输入命令的**no**形式禁用语音感知802.1x安全。此命令会应用到交换机上所有配置了802.1x的端口。

  注释: 如果不包括**shutdown vlan**关键字，进入错误禁用状态时整个端口都会被关闭。
- 如果使用全局配置命令**errdisable recovery cause security-violation**配置错误禁用恢复，端口会被自动重启。如果没有为端口配置错误禁用恢复，可以使用**shutdown**和**no shutdown**接口配置命令重启端口。
- 可以使用特权EXEC命令**clear errdisable interface** *interface-id* **vlan** [*vlan-list*]重启单个VLAN。如果不指定范围，端口上的所有VLAN都会被启用。

在特权 EXEC 模式中按照以下步骤启用语音感知 802.1x 安全特性。

总步骤

1. configure terminal

2. errdisable detect cause security-violation shutdown vlan

3. errdisable recovery cause security-violation

4. clear errdisable interface*interface-id* **vlan** *[vlan-list]*

5. 输入以下命令：

• shutdown

• no shutdown

6. end

7. show errdisable detect

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal**	进入特权 EXEC 模式。在提示时输入密

		码
步骤 2	**errdisable detect cause security-violation shutdown vlan**	关闭发生了安全违规错误的 VLAN。 **注释**：如果未包含 **shutdown vlan** 关键字，整个端口都会进入错误禁用状态并被关闭
步骤 3	**errdisable recovery cause security-violation**	配置错误禁用恢复。
步骤 4	**clear errdisable interface***interface-id* **vlan** *[vlan-list]*	（可选）重新启用单个错误禁用的 VLAN。 • 使用 interface-id 指定要重新启用 VLAN 的端口。 • （可选）使用 vlan-list 指定要重新启用的 VLAN 列表。如果 vlan-list 不指定，所有 VLAN 都被重启
步骤 5	输入以下命令： **• shutdown** **• no shutdown**	（可选）重新启用错误禁用的 VLAN，清除错误禁用标志
步骤 6	**end**	返回特权 EXEC 模式
步骤 7	**show errdisable detect**	验证配置的条目

以下示例展示了如何配置交换机，关闭发生了安全违规错误的 VLAN：

```
Switch(config)# errdisable detect cause security-violation shutdown vlan
```

以下示例展示了如何在 Gigabit Ethernet 40/2 端口上启用所有错误禁用的 VLAN。

```
Switch# clear errdisable interface gigabitethernet4/0/2vlan
```

可以输入特权 EXEC 命令 **show errdisable detect** 验证设置。

配置 802.1x 违规模式

可以配置 802.1x 端口，使其在在以情况发生时关闭端口、生成 syslog 消息或者丢弃来自新设备的包：

- 设备连接到启用 802.1x 的端口
- 端口被认证的设备到达最大数量

在特权 EXEC 模式中按照以下步骤在交换机上配置安全违规操作。

总步骤

1. configure terminal

2. aaa new-model

3. aaa authentication dot1x {**default**} *method1*

4. interface *interface-id*

5. switchport mode access

6. authentication violation {**shutdown** | **restrict** | **protect** | **replace**}

7. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例:** `Device#` **configure terminal**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**aaa new-model** **示例:** `Device(config)#` **aaa new-model**	启用 AAA
步骤 3	**aaa authentication dot1x** {**default**} *method1* **示例:** `Device(config)#` **aaa authentication dot1x default group radius**	创建 802.1x 认证方式列表。 当*没有*在 **authentication** 命令中指定命名的列表时，可以在方法之后加上 **default** 关键字来创建默认列表，以在默认情况下使用。默认方式列表会自动应用到所有端口上。 对于 *method1* 字段，输入关键字，使用所有 RADIUS 服务器列表进行认证
步骤 4	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet1/0/4**	指定连接到客户端的将要启用 IEEE 802.1x 认证的端口，并进入接口配置模式
步骤 5	**switchport mode access** **示例:** `Device(config-if)#` **switchport mode access**	设置端口为接入模式

步骤 6	**authentication violation {shutdown | restrict | protect | replace}** **示例:** `Device(config-if)#` **authentication violation restrict**	配置违规模式。关键字含义如下： • **shutdown**——错误禁用端口 • **restrict**——生成 syslog 错误 • **protect**——丢弃任何新设备发给端口的流量 • **replace**——移除当前会话，对新主机进行认证
步骤 7	**end** **示例:** `Device(config-if)#` **end**	返回特权 EXEC 模式

配置 802.1x 认证

为使用基于用户的 ACL 或 VLAN 分配，必须配置交换机为所有网络相关的服务请求启用 AAA 授权。

以下是 802.1x AAA 过程。

在开始前

要配置 802.1x 基于端口的认证，必须启用认证、授权以及审计（AAA）并指定认证方式列表。方式列表描述了向认证服务器查询的认证方式顺序。

总步骤

1. **用户连接到交换机端口。**
2. **进行认证。**
3. **基于 RADIUS 服务器配置进行 VLAN 分配。**
4. **交换机给审计服务器发送开始消息。**
5. **按需执行重新认证。**
6. **基于重新认证结果，交换机给审计服务器发送中间审计更新消息。**
7. **用户断开端口。**
8. **交换机给审计服务器发送停止消息。**

具体步骤

	命令或操作	**目的**

配置 802.1x 基于端口的认证

在特权 EXEC 模式中按照以下步骤配置 802.1x 基于端口的认证。

总步骤

1. configure terminal

2. aaa new-model

3. aaa authentication dot1x {**default**} *method1*

4. dot1x system-auth-control

5. aaa authorization network {**default**} **group radius**

6. radius server *server name*

7. key *string*

8. interface *interface-id*

9. switchport mode access

10. authentication port-control auto

11. dot1x pae authenticator

12. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例:** `Device#` **configure terminal**	进入特权 EXEC 模式
步骤 2	**aaa new-model** **示例:** `Device(config)#` **aaa new-model**	启用 AAA
步骤 3	**aaa authentication dot1x** {**default**} *method1* **示例:** `Device(config)#` **aaa authentication dot1x default group radius**	创建 802.1x 认证方式列表。 当*没有*在 **authentication** 命令中指定命名的列表时，可以在方法之后加上 **default** 关键字来创建默认列表，以在默认情况下使用。默认方式列表会自动应用到所有端口上。 对于 *method1* 字段，输入关键字，使用所有 RADIUS 服务器列表进行认证。

		注释：虽然命令行帮助字符串中还有其他可见的关键字，但只支持 **group radius** 关键字
步骤 4	**dot1x system-auth-control** **示例:** `Device(config)#` **dot1x system-auth-control**	在交换机上全局启用 802.1x 认证
步骤 5	**aaa authorization network {default} group radius** **示例:** `Device(config)#` **aaa authorization network default group radius**	（可选）配置交换机使用用户 RADIUS 授权所有网络相关的服务请求，如基于用户的 ACL 或 VLAN 分配
步骤 6	**radius server** *server name* **示例:** `Device(config)#` **radius server rsim address ipv4 124.2.2.12**	（可选）指定 RADIUS 服务器的 IP 地址
步骤 7	**key** *string* **示例:** `Device(config-radius-server)#` **key rad123**	（可选）指定交换机和 RADIUS 服务器上 RADIUS 后台进程之间使用的认证及加密密钥
步骤 8	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet1/0/2**	指定连接到客户端的将要启用 IEEE 802.1x 认证的端口，并进入接口配置模式
步骤 9	**switchport mode access** **示例:** `Device(config-if)#` **switchport mode access**	（可选）如果在步骤 6、7 中配置了 RADIUS 服务器，设置端口为接入模式
步 骤 10	**authentication port-control auto** **示例:** `Device(config-if)#` **authentication port-control auto**	在端口上启用 802.1x 认证
步 骤	**dot1x pae authenticator**	设备端口接入实体（Port Access Entity）

11	**示例:** `Device(config-if)#` **dot1x pae authenticator**	只作为认证者，忽略所有发往请求者的消息
步 骤 12	**end** **示例:** `Device(config-if)#` **end**	返回特权 EXEC 模式

配置交换机到 RADIUS 服务器的通信

管理员也需要在 RADIUS 服务器上进行一些设置，包括交换机的 IP 地址，以及服务器和交换机共享的密钥穿。更多信息参见 RADIUS 服务器文档。

按照以下步骤在交换机上配置 RADIUS 服务器参数。此步骤是必须的。

在开始前

必须启用认证、授权以及审计（AAA）并指定认证方式列表。方式列表描述了向认证服务器查询认证方式的顺序。

总步骤

1. enable

2. configure terminal

3. radius server *server name*

4. address {**ipv4 | ipv6**} *ip address* **auth-port** *port number* **acct-port** *port number*

5. key *string*

6. end

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式
步骤 3	**radius server** *server name* **示例:**	指定 RADIUS 服务器名称，并进入 RADIUS 服务器配置模式

	Device(config)# **radius server rsim**	
步骤 4	**address** {**ipv4 | ipv6**} *ip address* **auth-port** *port number* **acct-port** *port number* **示例:** Device(config-radius-server)# **address ipv4 124.2.2.12**	指定 RADIUS 服务器的 IP 地址。 使用 **auth-port** *port-number*,指定认证请求的 UDP 目的端口。默认值是 1645，范围从 0 到 65536。 使用 **acct-port** *port-number* 指定认证请求的 UDP 的目的端口。默认值是 1646
步骤 5	**key** *string* **示例:** Device(config-radius-server)# **key rad123**	指定设备以及 RADIUS 服务器运行的 RADIUS 后台程序之间使用的认证以及加密密钥。 **注释：**此密钥是明文密钥，且必须与 RADIUS 服务器使用的加密密钥相同。把密钥配置在 **radius server** 命令的最后一项。密钥前的空格会被忽略，但是密钥中间以及之后的空格会被使用。如果密钥使用空格，不要把密钥放在引号之间，除非引号是密钥的一部分
步骤 6	**end** **示例:** Device(config)# **end**	返回特权 EXEC 模式

配置主机模式

对于接口配置命令 **authentication port-control** 设置为 **auto** 的 IEEE 802.1x 授权端口，可以在特权 EXEC 模式中按照以下步骤配置，以允许有多台主机。配置关键字 **multi-domain** 可以启用多域认证（MDA），允许在一个交换机端口上同时有主机和语音设备（如 Inspur 或非 Inspur 的 IP 电话）。此步骤是可选的。

总步骤

1. configure terminal

2. interface *interface-id*

3. authentication host-mode [**multi-auth** | **multi-domain** | **multi-host** | **single-host**]

4. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet2/0/1**	指定多台主机直连的端口，并进入接口配置模式
步骤 3	**authentication host-mode** [**multi-auth** | **multi-domain** | **multi-host** | **single-host**] **示例:** `Device(config-if)#` **authentication host-mode multi-host**	允许 802.1x 授权的端口上有多台主机。 关键字含义如下： • **multi-auth**——允许语音 VLAN 有一台客户端，数据 VLAN 有多台客户端被认证。 **注释：multi-auth** 关键字只在 **authentication host-mode** 命令中可用。 • **multi-host**——在一台主机被认证之后，允许 802.1x 授权端口上存在多台主机。 • **multi-domain**——允许 IEEE 802.1x 授权端口上有一台主机以及一台语音设备。 **注释：**当主机模式设置为时必须为 IP 电话配置语音 VLAN。 确保特定接口的接口配置命令 **authentication port-control** 设置为 **auto**
步骤 4	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式

配置周期性重新认证

可以启用周期性的 802.1x 客户端重新认证功能并指定执行频率。如果不指定时间周期，尝试重新认证的周期是 3600 秒。

在特权 EXEC 模式中，按照以下步骤启用客户端的周期性重新认证，并指定尝试重新认证的周期。此过程是可选的。

总步骤

1. configure terminal

2. interface *interface-id*

3. authentication periodic

4. authentication timer {{[**inactivity** | **reauthenticate** | **restart**]} {*value*}}

5. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet2/0/1**	指定要配置的端口，并进入接口配置模式
步骤 3	**authentication periodic** **示例:** `Device(config-if)#` **authentication periodic**	启用客户端的重新认证，该设置默认被禁用。 **注释：**默认周期是3600秒。如需更改重新认证计时器值或者使用RADIUS服务器提供的会话超时时间，输入命令 **authenticationtimer reauthenticate**
步骤 4	**authentication timer** {{[**inactivity** | **reauthenticate** | **restart**]} {*value*}} **示例:** `Device(config-if)#` **authentication timer reauthenticate 180**	设置尝试重新认证的周期描述。 **authentication timer** 关键字含义如下： • **inactivity**——在设置的间隔秒数后如果客户端无活动，则其变为未授权。

		• **reauthenticate**——尝试进行自动重新认证的秒数间隔。 • **restart** *value*——尝试认证未授权端口的秒数间隔。 如果启用了周期性重新认证，此命令会影响交换机的行为
步骤 5	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式

更改静默周期

当交换机无法认证客户端时，交换机会保持闲置一段时间然后再次尝试认证。接口配置命令 **authentication timer inactivity** 控制着闲置时长。认证失败可能是因为客户端提供了非法的密码。可以设置比默认值更小的时长，为用户提供更快的响应时间。

在特权 EXEC 模式中按照以下步骤更改静默周期。此步骤是可选的。

总步骤

1. configure terminal

2. interface *interface-id*

3. authentication timer inactivity *seconds*

4. end

5. show authentication sessions interface *interface-id*

6. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet2/0/1**	指定要配置的端口，并进入接口配置模式

步骤 3	**authentication timer inactivity** *seconds* **示例:** `Device(config-if)#` **authentication timer inactivity** **30**	设置交换机与客户端认证交换失败之后保持静默状态的秒数。 范围从1到65536秒，默认值是60秒
步骤 4	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式
步骤 5	**show authentication sessions interface** *interface-id* **示例:** `Device#` **show authentication sessions interface gigabitethernet2/0/1**	验证配置的条目
步骤 6	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中

更改交换机到客户端的重传时间

客户端会使用 EAP 应答/身份帧响应来自交换机的 EAP 请求/身份帧。如果交换机没收收到应答，它会等待一定的时长（称为重传时间）然后重新发送数据帧。

注释： 只应在不寻常的情况下更改此命令的默认值，比如存在不可靠的链路或者客户端、认证服务器存在特定的行为问题。

在特权 EXEC 模式中，按照以下步骤更改交换机等待客户端通知的时长。此过程是可选的。

总步骤

1. configure terminal

2. interface *interface-id*

3. authentication timer reauthenticate *seconds*

4. end

5. show authentication sessions interface *interface-id*

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet2/0/1**	指定要配置的端口，并进入接口配置模式
步骤 3	**authentication timer reauthenticate** *seconds* **Example:** `Device(config-if)#` **authentication timer reauthenticate 60**	设置交换机在重新发送请求之前等待客户端回应EAP请求/身份帧的时长。范围从1到65535秒，默认值是5秒
步骤 4	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式
步骤 5	**show authentication sessions interface** *interface-id* **示例:** `Device#` **show authentication sessions interface gigabitethernet2/0/1**	验证配置的条目
步骤 6	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中

设置交换机到客户端的帧重传次数

除了更改交换机到客户端的重传时间,还可以更改交换机在重启认证过程之前向客户端发送 EAP 请求/身份帧的次数（假设未收到响应）。

注释： 只应在不寻常的情况下更改此命令的默认值，比如存在不可靠的链路或者客户端、认证服务器存在特定的行为问题。

总步骤

1. configure terminal

2. interface *interface-id*

3. dot1x max-reauth-req *count*

4. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet2/0/1**	指定要配置的端口，并进入接口配置模式
步骤 3	**dot1x max-reauth-req** *count* **示例:** `Device(config-if)#` **dot1x max-reauth-req 5**	设置交换机在重启认证过程之前向客户端发送EAP请求/身份帧的次数。 范围从1到10，默认值是2
步骤 4	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式

设置重新认证次数

也可以更改端口变成未授权状态之前交换机重启认证过程的次数。

注释： 只应在不寻常的情况下更改此命令的默认值，比如存在不可靠的链路或者客户端、认证服务器存在特定的行为问题。

总步骤

1. configure terminal

2. interface *interface-id*

3. switchport mode access

4. dot1x max-req *count*

5. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet2/0/1**	指定要配置的端口，并进入接口配置模式。
步骤 3	**switchport mode access** **示例:** `Device(config-if)#` **switchport mode access**	仅在之前配置了 RADIUS 服务器的情况下把端口设置为接入模式。
步骤 4	**dot1x max-req** *count* **示例:** `Device(config-if)#` **dot1x max-req 4**	设置端口变成未授权状态之前交换机重启认证过程的次数。范围从0到10，默认值是2。
步骤 5	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。

启用 MAC 移动

MAC 移动特性允许已认证的主机从交换机上的一个端口移动到另一个端口。

在特权 EXEC 模式中按照以下步骤在交换机上全局启用 MAC 移动特性。此过程是可选的。

总步骤

1. configure terminal

2. authentication mac-move permit

3. end

4. show running-config

5. copy running-config startup-config

具体步骤

	命令或操作	目的

步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**authentication mac-move permit** **示例:** `Device(config)#` **authentication mac-move permit**	在交换机上启用 MAC 移动特性，该特性默认被禁用。 在会话感知网络模式中，默认设置是 **access-session mac-move deny**.。要在会话感知网络中启用MAC移动，使用全局配置命令**no access-sessionmac-move**。
步骤 3	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 4	**show running-config** **示例：** `Device#` **show running-config**	验证配置的条目。
步骤 5	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。

启用 MAC 替换

MAC 替换特性允许一台主机替代端口上另一台已认证的主机。

在特权 EXEC 模式中，按照以下步骤在接口上启用 MAC 替换。此过程是可选的。

总步骤

1. configure terminal

2. interface *interface-id*

3. authentication violation {**protect** | **replace** | **restrict** | **shutdown**}

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的

步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet2/0/2**	指定要配置的端口，并进入接口配置模式。
步骤 3	**authentication violation** {**protect** | **replace** | **restrict** | **shutdown**} **示例:** `Device(config-if)#` **authentication violation replace**	使用 **replace** 关键字在接口上启用 MAC 替换。 端口会移除当前会话并发起对新主机的认证。 其他关键词的效果如下： • **protect**：端口丢弃非预期 MAC 地址的数据包，且不会生成系统消息。 • **restrict**：违规的数据包会被 CPU 丢弃，且会生成系统消息。 • **shutdown**：端口在收到非预期 MAC 地址的数据包时会被错误禁用。
步骤 4	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 5	**show running-config** **示例：** `Device#` **show running-config**	验证配置的条目。
步骤 6	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。

配置 802.1x 审计

配置 AAA 系统审计功能进行 802.1x 的审计，会让系统重载发送给审计 RADIUS 服务器的事

件。服务器进而可以推测所有活跃的 802.1x 会话都已关闭。

因为 RADIUS 使用不可靠的 UDP 传输协议，审计消息可能因为网络状况差而丢失。在可配置次数的审计请求重传之后，如果交换机没有收到来自 RADIUS 服务器的审计应答消息，会显示以下系统消息：

```
Accounting message %s for session %s failed to receive Accounting Response.
```

当停止消息没有成功发送时，会出现以下消息：

```
00:09:55: %RADIUS-4-RADIUS_DEAD: RADIUS server 172.20.246.201:1645,1646 is not responding.
```

注释： 必须配置 RADIUS 服务器执行审计任务，比如审计开始、停止以及中间更新消息和时间戳。要启用这些功能，可以在 RADIUS 服务器网络配置页中启用“更新/Watchdog 来自此 AAA 客户端的数据包”记录功能。接着，在 RADIUS 服务器的系统配置页启用“CVS RADIUS 审计”。

在交换机上启用 AAA 之后，在特权 EXEC 模式中按照以下步骤配置 802.1x 审计功能。此过程是可选的。

总步骤

1. **configure terminal**
2. **interface** *interface-id*
3. **aaa accounting dot1x default start-stop group radius**
4. **aaa accounting system default start-stop group radius**
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例:** `Device(config)# interface gigabitethernet1/0/3`	指定要配置的端口，并进入接口配置模式。
步骤 3	**aaa accounting dot1x default start-stop group radius**	使用 RADIUS 服务器列表启用 802.1x 审计。

	示例: `Device(config-if)#` **aaa accounting dot1x default** **start-stop group radius**	
步骤 4	**aaa accounting system default start-stop group radius** **示例:** `Device(config-if)#` **aaa accounting system default** **start-stop group radius**	（可选）启用系统审计功能（使用 RADIUS 服务器列表）并在交换机重启时生成系统审计重启事件消息。
步骤 5	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 6	**show running-config** **示例：** `Device#` **show running-config**	验证配置的条目。
步骤 7	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。

配置访客 VLAN

配置访客 VLAN 时，如果服务器没收到 EAP 请求/身份帧的应答，不兼容 802.1x 的客户端会被置于访客 VLAN 中。兼容 802.1x 但是认证失败的客户端不被授权网络访问权限。交换机在单主机或多主机模式中支持使用访客 VLAN。

在特权 EXEC 模式中，按照以下步骤配置访客 VLAN。此过程是可选的。

总步骤

1. configure terminal

2. interface *interface-id*

3. 使用以下命令之一：

- **switchport mode access**
- **switchport mode private-vlan host**

4. authentication event no-response action authorize vlan *vlan-id*

5. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet1/0/3**	指定要配置的端口，并进入接口配置模式。
步骤 3	使用以下命令之一： **• switchport mode access** **• switchport mode private-vlan host** **示例:** `Device(config-if)#` **switchport mode private-vlan host**	• 设置端口为接入模式。 • 配置二层端口为私有 VLAN 主机端口。
步骤 4	**authentication event no-response action authorize vlan** *vlan-id* **示例:** `Device(config-if)#` **authentication event no-response action authorize vlan 2**	指定一个活跃的 VLAN 为 802.1x 访客 VLAN，范围从 1 到 4094。 可以把除了内部 VLAN（被路由端口）、RSPAN VLAN 或语音 VLAN 之外的任意活跃 VLAN 配置为 802.1x 访客 VLAN。
步骤 5	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。

配置受限 VLAN

在交换机堆栈或者交换机上配置受限 VLAN 时，如果认证服务器没有收到合法的客户端用户名及密码，这些兼容 IEEE 802.1x 的客户端会被移动到受限 VLAN 中。交换机只在单主机模式中支持使用受限 VLAN。

在特权 EXEC 模式中，按照以下步骤配置受限 VLAN。此过程是可选的。

总步骤

1. configure terminal

2. interface *interface-id*

3. 使用以下命令之一：

- **switchport mode access**
- **switchport mode private-vlan host**

4. authentication port-control auto

5. authentication event fail action authorize vlan *vlan-id*

6. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet1/0/3**	指定要配置的端口，并进入接口配置模式。
步骤 3	使用以下命令之一： **• switchport mode access** **• switchport mode private-vlan host** **示例:** `Device(config-if)#` **switchport mode private-vlan host**	• 设置端口为接入模式。 • 配置二层端口为私有 VLAN 主机端口。
步骤 4	**authentication port-control auto** **示例:** `Device(config-if)#` **authentication port-controlauto**	在端口上启用 802.1x 认证。
步骤 5	**authentication event fail action authorize vlan** *vlan-id* **示例:**	指定一个活跃的 VLAN 为 802.1x 受限 VLAN，范围从 1 到 4094。 可以把除了内部 VLAN（被路由端口）、

	Device(config-if)# **authentication event fail action authorize vlan 2**	RSPAN VLAN 或语音 VLAN 之外的任意活跃 VLAN 配置为 802.1x 受限 VLAN。
步骤 6	**end** **示例:** Device(config)# **end**	返回特权 EXEC 模式。

配置受限 VLAN 上的认证尝试次数

使用接口配置命令 **authentication event retry** *retry count*，可以配置给用户分配受限 VLAN 之前允许的最大认证尝试次数。允许的认证尝试次数范围从 1 到 3，默认值是 3。

在特权 EXEC 模式中，按照以下步骤配置允许的最大认证尝试次数。此过程是可选的。

总步骤

1. configure terminal

2. interface *interface-id*

3. 使用以下命令之一：

- **switchport mode access**
- **switchport mode private-vlan host**

4. authentication port-control auto

5. authentication event fail action authorize vlan *vlan-id*

6. authentication event retry *retry count*

7. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例:** Device(config)# **interface gigabitethernet2/0/3**	指定要配置的端口，并进入接口配置模式。
步骤 3	使用以下命令之一：	• 设置端口为接入模式。

	• **switchport mode access** • **switchport mode private-vlan host** **示例:** `Device(config-if)#` **switchport mode private-vlan host**	• 配置二层端口为私有 VLAN 主机端口。
步骤 4	**authentication port-control auto** **示例:** `Device(config-if)#` **authentication port-controlauto**	在端口上启用 802.1x 认证。
步骤 5	**authentication event fail action authorize vlan** *vlan-id* **示例:** `Device(config-if)#` **authentication event fail action authorize vlan 8**	指定一个活跃的 VLAN 为 802.1x 受限 VLAN，范围从 1 到 4094。 可以把除了内部 VLAN（被路由端口）、RSPAN VLAN 或语音 VLAN 之外的任意活跃 VLAN 配置为 802.1x 受限 VLAN。
步骤 6	**authentication event retry** *retry count* **示例:** `Device(config-if)#` **authentication event retry 2**	配置把用户移动到受限 VLAN 之前允许的最大认证尝试次数。范围从 1 到 3，默认值是 3。
步骤 7	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。

配置 802.1x 不可访问旁路认证以及临界语音 VLAN

在特权 EXEC 模式中，按照以下步骤在端口上配置临界语音 VLAN 并启用不可访问旁路认证特性。

总步骤

1. configure terminal

2. aaa new-model

3. radius-server dead-criteria{**time** *seconds* } [**tries** *number*]

4. radius-serverdeadtime*minutes*

5. radius-server host ip-address *address*[**acct-port** *udp-port*][**auth-port** *udp-port*] [**testusername** *name*[**idle-time** *time*] [**ignore-acct-port**][**ignore auth-port**]] [**key** *string*]

6. dot1x critical {**eapol** | **recovery delay** *milliseconds*}

7. interface *interface-id*

8. authentication event server dead action {**authorize** | **reinitialize**} **vlan** *vlan-id*]

9. switchport voice vlan *vlan-id*

10. authentication event server dead action authorize voice

11. show authentication interface *interface-id*

12. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**aaa new-model** **示例:** `Device(config)#` **aaa new-model**	启用 AAA。
步骤 3	**radius-server dead-criteria**{**time** *seconds* } [**tries** *number*] **示例:** `Device(config)#` **radius-server dead-criteria time 20 tries 10**	设置决定 RADIUS 服务器不可用或 down 的条件。 • **time**——1 到 120 秒。交换机会动态决定一个在 10 到 60 之间的默认 *seconds* 值。 • **number**——1 到 100 次尝试。交换机会动态决定一个在 10 到 100 之间的默认 *number* 值。
步骤 4	**radius-server deadtime** *minutes* **示例:** `Device(config)#` **radius-server deadtime 60**	（可选）设置 RADIUS 服务器不发送请求的分钟数。范围从 0 到 1440 分钟（24 小时）。默认值是 0 分钟。
步骤 5	**radius-server host ip-address** *address*[**acct-port** *udp-port*][**auth-port**	（可选）使用以下关键字配置 RADIUS 服务器参数：

	udp-port] [**testusername** *name*[**idle-time** *time*] [**ignore-acct-port**][**ignore auth-port**]] [**key** *string*] **示例:** `Device(config)#` **radius-server host 1.1.1.2 acct-port 1550 auth-port 1560 test username user1 idle-time 30 key abc1234**	• **acct-port** *udp-port*——指定 RADIUS 审计服务器的 UDP 端口。UDP 端口号范围从 0 到 65536，默认值是 1646。 • **auth-port** *udp-port* ——指定 RADIUS 认证服务器的 UDP 端口。UDP 端口号范围从 0 到 65536，默认值是 1645。 **注释：**可以把 RADIUS 认证服务器及审计服务器的 UDP 端口配置为非默认值。 • **test username** *name*——自动测试 RADIUS 服务器状态，并指定使用的用户名。 • **idle-time** *time*——设置交换机给服务器发送测试包的间隔分钟数。范围从 1 到 35791 分钟，默认值是 60 分钟（1 小时）。 • **ignore-acct-port**——禁用 RADIUS 服务器审计端口的测试。 • **ignore-auth-port**——禁用 RADIUS 服务器认证端口的测试。 • 使用 **key** *string* 指定交换机和 RADIUS 服务器上的 RADIUS 后台程序之间使用的认证及加密密钥。此密钥是明文密钥，且必须与 RADIUS 服务器使用的加密密钥相同。 **注释：**把密钥配置在 radius serverhost 命令的最后一项。密钥前的空格会被忽略，但是密钥中间以及之后的空格会被使用。如

		果密钥使用空格，不要把密钥放在引号之间，除非引号是密钥的一部分。 也可以使用全局配置命令 **radius-server key** {**0**string | **7***string* | *string*}配置认证及加密密钥。
步骤 6	**dot1x critical** {**eapol** | **recovery delay** *milliseconds*} **示例:** `Device(config)#` **dot1x critical eapol** `(config)#` **dot1x critical recovery delay 2000**	（可选）配置不可访问旁路认证参数： • **eapol**——指定交换机在成功认证临界端口时发送一个 EAPOL 成功消息。 • **recovery delay** *milliseconds*——指定在不可用的 RADIUS 服务器变为可用时交换机重新初始化临界端口要等待的恢复时延。范围从 1 到 10000 毫秒，默认值是 1000 毫秒（端口每秒都可以被重新初始化）。
步骤 7	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet 1/0/1**	指定要配置的端口，并进入接口配置模式。
步骤 8	**authentication event server dead action** {**authorize** | **reinitialize**} **vlan** *vlan-id*] **示例:** `Device(config-if)#` **authentication event server dead action reinitialicze vlan 20**	使用以下关键字在 RADIUS 服务器不可用时把端口上主机移动到临界 VLAN： • **authorize**——把任意尝试认证的新主机移动到用户指定的临界 VLAN 中。 • **reinitialize**——把端口上所有已授权的主机移动到用户指定的临界 VLAN 中。
步骤 9	**switchport voice vlan** *vlan-id* **Example:** `Device(config-if)#` **switchport voice**	为端口指定语音 VLAN。语音 VLAN 不能与配置的临界 VLAN 相同。

	vlan	
步 骤 10	**authentication event server dead action authorize voice** **示例:** `Device(config-if)#` **authentication event server dead action authorize voice**	配置临界语音 VLAN，如果 RADIUS 服务器不可用时，把端口上的数据流量移动到语音 VLAN 中。
步 骤 11	**show authentication interface** *interface-id* **示例:** `Device(config-if)#` **do show authentication interface gigabit 1/0/1**	（可选）验证配置的条目。
步 骤 12	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。

要返回RADIUS服务器的默认设置，使用全局配置命令**no radius-server dead-criteria**，**radius-serverdeadtime**，和**no radius-server host**。要禁用不可访问旁路认证，使用接口配置命令**no authentication event server dead action**。要禁用临界语音VLAN，使用接口配置命令**noauthentication event server dead action authorize voice**。

配置不可访问旁路认证的示例

以下示例展示了如果配置不可访问旁路认证特性：

```
Device(config)# radius-server dead-criteria time 30 tries 20
Device(config)# radius-server deadtime 60
Device(config)# radius-server host 1.1.1.2 acct-port 1550 auth-port 1560 test username user1
idle-time 30 key abc1234
Device(config)# dot1x critical eapol
Device(config)# dot1x critical recovery delay 2000
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# dot1x critical
Device(config-if)# dot1x critical recovery action reinitialize
Device(config-if)# dot1x critical vlan 20
Device(config-if)# end
```

配置 802.1x 认证以及 WoL

在特权 EXEC 模式中，按照以下步骤启用 802.1x 以及 WoL。此过程是可选的。

总步骤

1. configure terminal

2. interface *interface-id*

3. authentication control-direction {**both** | **in**}

4. end

5. show authentication sessions interface *interface-id*

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet1/0/3**	指定要配置的端口，并进入接口配置模式。
步骤 3	**authentication control-direction** {**both** | **in**} **示例:** `Device(config-if)#` **authentication control-direction both**	在端口上启用 802.1x 认证以及 WoL，使用以下关键字配置端口执行单向操作或双向操作。 • **both**——设置端口为双向。端口无法收发主机的数据包。默认情况下端口为双向。 • **in**——设置端口为单向。端口可以向主机发送数据包，但不能接收来自主机的数据包。
步骤 4	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 5	**show authentication sessions interface**	验证配置的条目。

	interface-id **示例:** `Device#` **show authentication sessions interface gigabitethernet2/0/3**	
步骤 6	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。

配置 MAC 旁路认证

在特权 EXEC 模式中，按照以下步骤启用 MAC 旁路认证。此过程是可选的。

总步骤

1. configure terminal

2. interface *interface-id*

3. authentication port-control auto

4. mab [**eap**]

5. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet2/0/1**	指定要配置的端口，并进入接口配置模式。
步骤 3	**authentication port-control auto** **示例:** `Device(config-if)#` **authentication port-control auto**	在端口上启用 802.1x 认证。

步骤 4	**mab** [**eap**] **示例:** `Device(config-if)#` **mab**	启用 MAC 旁路认证。 （可选）使用关键字 **eap** 配置交换机使用 EAP 进行认证。
步骤 5	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。

配置 802.1x 用户分配

在特权 EXEC 模式中，按照以下步骤配置 VLAN 群组并映射 VLAN 到其中。

总步骤

1. configure terminal

2. vlan group *vlan-group-name* **vlan-list** *vlan-list*

3. end

4. no vlan group *vlan-group-name* **vlan-list** *vlan-list*

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**vlan group** *vlan-group-name* **vlan-list** *vlan-list* **示例:** `Device(config)#` **vlan group eng-dept vlan-list 10**	配置 VLAN 群组，并把一个 VLAN 或一个范围的 VLAN 映射到其中。
步骤 3	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 5	**no vlan group** *vlan-group-name* **vlan-list** *vlan-list* **示例:**	清理 VLAN 群组配置或 VLAN 群组配置中的某个元素。

	Device(config)# **no vlan group eng-dept vlan-list 10**	

配置 VLAN 群组的示例

以下示例显示了如何配置 VLAN 群组，映射 VLAN 到群组以及验证 VLAN 群组的配置及映射。

```
Device(config)# vlan group eng-dept vlan-list 10
Device(config)# show vlan group group-name eng-dept
Group Name Vlans Mapped
------------ --------------
eng-dept 10
Device(config)# show dot1x vlan-group all
Group Name Vlans Mapped
------------ --------------
eng-dept 10
hr-dept
```

以下示例显示了如何把 VLAN 添加到现有的 VLAN 群组，并验证操作。

```
Device(config)# vlan group eng-dept vlan-list 30
Device(config)# show vlan group eng-dept
Group Name       Vlans Mapped
-------------    --------------
eng-dept         10,30
```

以下示例显示了如何从 VLAN 群组中移除 VLAN。

```
Device# no vlan group eng-dept vlan-list 10
```

以下示例显示了当 VLAN 群组中的所有 VLAN 都被清除时，VLAN 群组也会被清除。

```
Device(config)# no vlan group eng-dept vlan-list 30
Vlan 30 is successfully cleared from vlan group eng-dept.
Device(config)# show vlan group group-name eng-dept
```

以下示例显示了如何清除所有 VLAN 群组。

```
Device(config)# no vlan group end-dept vlan-list all
Device(config)# show vlan-group all
```

更多有关以上命令的信息，参见 *Inspur INOS 安全性命令手册*。

配置 NAC 二层 802.1x 验证

可以配置 NAC 二层 802.1x 验证特性 。

在特权 EXEC 模式中，按照以下步骤配置 NAC 二层 802.1x 验证。此过程是可选的。

总步骤

1. configure terminal

2. interface *interface-id*

3. switchport mode access

4. authentication event no-response action authorize vlan *vlan-id*

5. authentication periodic

6. authentication timer reauthenticate

7. end

8. show authentication sessions interface *interface-id*

9. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet1/0/3**	指定要配置的端口，并进入接口配置模式。
步骤 3	**switchport mode access** **示例:** `Device(config-if)#` **switchport mode access**	仅在配置了 RADIUS 服务器时设备端口为接入模式。
步骤 4	**authenticationevent no-response actionauthorize vlan** *vlan-id* **示例:** `Device(config-if)#` **authentication event**	指定一个活跃的 VLAN 为 802.1x 访客 VLAN，范围从 1 到 4094。 可以把除了内部 VLAN（被路由端口）、RSPAN VLAN 或语音 VLAN 之外的任意活跃 VLAN 配置为 802.1x 访客 VLAN。

	no-response action authorize vlan 8	
步骤 5	**authentication periodic** **示例:** `Device(config-if)#` **authentication periodic**	启用对客户端的周期性重新认证，该特性默认禁用。
步骤 6	**authentication timer reauthenticate** **示例:** `Device(config-if)#` **authentication timer reauthenticate**	设置尝试重新认证客户端（设置为 1 小时）。 如果启用周期性重新认证，此命令会影响交换机行为。
步骤 7	**end** **示例:** `Device(config-if)#` **end**	返回特权 EXEC 模式。
步骤 8	**show authentication sessions interface** *interface-id* **示例:** `Device#` **show authentication sessions interface gigabitethernet2/0/3**	验证配置的条目。
步骤 9	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。

配置认证交换机以及 NEAT

配置此特性要求配线间外的一台交换机被配置为请求者，且连接到认证交换机。

注释： 必须在 ACS 上把 inspur-av-pair 配置为 *device-traffic-class=switch*，此配置会在请求者认证成功后把连接接口设置为中继。

在特权 EXEC 模式中，按照以下步骤配置交换机为认证者。

总步骤

1. configure terminal

2. cisp enable

3. interface *interface-id*

4. switchport mode access

5. authentication port-control auto

6. dot1x pae authenticator

7. spanning-tree portfast

8. end

9. show running-config interface *interface-id*

10. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**cisp enable** **示例:** `Device(config)#` **cisp enable**	启用 CISP。
步骤 3	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet1/0/3**	指定要配置的端口，并进入接口配置模式。
步骤 4	**switchport mode access** **示例:** `Device(config-if)#` **switchport mode access**	设置端口模式为接入模式。
步骤 5	**authentication port-control auto** **示例:** `Device(config-if)#` **authentication port-control auto**	设置端口认证模式为 auto。
步骤 6	**dot1x pae authenticator** **示例:** `Device(config-if)#` **dot1x pae authenticator**	配置接口为端口接入实体（PAE）认证者。
步骤 7	**spanning-tree portfast** **示例:** `Device(config-if)#` **spanning-tree portfast**	在连接到一台工作站或服务器的接入端口上启用 Port Fast。

	trunk	
步骤 8	**end** **示例:** `Device(config-if)#` **end**	返回特权 EXEC 模式。
步骤 9	**show running-config interface** *interface-id* **示例:** `Device#` **show running-config interface gigabitethernet2/0/1**	验证配置的条目。
步 骤 10	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。

配置请求交换机以及 NEAT

在特权 EXEC 模式中，按照以下步骤配置交换机为请求者。

总步骤

1. configure terminal

2. cisp enable

3. dot1x credentials *profile*

4. username *suppswitch*

5. password *password*

6. dot1x supplicant force-multicast

7. interface *interface-id*

8. switchport trunk encapsulation dot1q

9. switchport mode trunk

10. dot1x pae supplicant

11. dot1x credentials *profile-name*

12. end

13. show running-config interface *interface-id*

14. copy running-config startup-config

15. 配置 NEAT 以及自动智能端口宏（Auto Smartports Macros）

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**cisp enable** **示例:** `Device(config)#` **cisp enable**	启用 CISP。
步骤 3	**dot1x credentials** *profile* **示例:** `Device(config)#` **dot1x credentials test**	创建 802.1x 凭据配置。此配置必须挂接到配置为请求者的端口上。
步骤 4	**username** *suppswitch* **示例:** `Device(config)#` **username suppswitch**	创建用户名。
步骤 5	**password** *password* **示例:** `Device(config)#` **password myswitch**	为新用户名创建密码。
步骤 6	**dot1x supplicant force-multicast** **示例:** `Device(config)#` **dot1x supplicant force-multicast**	强制交换机只发送组播 EAPOL 包，无论其收到了单播包还是组播包。 这也使 NEAT 可以适用于所有主机模式的请求交换机。
步骤 7	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet1/0/1**	指定要配置的端口，并进入接口配置模式。
步骤 8	**switchport trunk encapsulation dot1q** **示例:** `Device(config-if)#` **switchport trunk encapsulation dot1q**	甚至端口为中继模式。
步骤 9	**switchport mode trunk** **示例:** `Device(config-if)#` **switchport mode trunk**	配置接口为 VLAN 中继端口。
步 骤	**dot1x pae supplicant**	配置接口为端口接入实体（PAE）请求

10	**示例:** `Device(config-if)#` **dot1x pae supplicant**	者。
步 骤 11	**dot1x credentials** *profile-name* **示例:** `Device(config-if)#` **dot1x credentials test**	把 802.1x 凭据配置挂接到接口上。
步 骤 12	**end** **示例:** `Device(config-if)#` **end**	返回特权 EXEC 模式。
步 骤 13	**show running-config interface** *interface-id* **示例:** `Device#` **show running-config interface gigabitethernet2/0/1**	验证配置的条目。
步 骤 14	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。
步 骤 15	配置NEAT以及自动智能端口宏	配置授权交换机时，也可以使用自动智能端口用户定义宏来代替交换机的 VSA。更多信息，参见此版本的*自动智能端口配置指南*。

配置 802.1x 认证、可下载的 ACL 以及重定向 URL

除了需要在交换机上配置 802.1x 认证之外，还需要配置 ACS。更多信息参见 *Inspur 安全 ACS 4.2 配置指南*:

http://www.icntnetworks.com

注释： 在交换机下载 ACL 之前，必须在 ACS 上配置一个可下载的 ACL。

在端口认证之后，可以使用特权 EXEC 命令 **show ip access-list** 显示端口下载的 ACL。

配置可下载的 ACL

当客户端完成认证且客户端的 IP 地址被加入 IP 设备追踪表之后，该策略生效。交换机随后会把可下载 ACL 应用到端口上。

总步骤

1. configure terminal

2. ip device tracking

3. aaa new-model

4. aaa authorization network default local group radius

5. radius-server vsa send authentication

6. interface *interface-id*

7. ip access-group *acl-id* **in**

8. show running-config interface *interface-id*

9. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**ip device tracking** **示例:** `Device(config)#` **ip device tracking**	设置 IP 设备追踪表。
步骤 3	**aaa new-model** **示例:** `Device(config)#` **aaa new-model**	启用 AAA。
步骤 4	**aaa authorization network default local group radius** **示例:** `Device(config)#` **aaa authorization network default local group radius**	设备本地授权方式。要移除授权方式，使用命令**no aaa authorizationnetwork default local group radius**。
步骤 5	**radius-server vsa send authentication** **示例:** `Device(config)#` **radius-server vsa send authentication**	配置 RADIUS VSA 发送认证。
步骤 6	**interface** *interface-id* **示例:** `Device(config)#` **interface**	指定要配置的端口，并进入接口配置模式。

	gigabitethernet2/0/4	
步骤 7	**ip access-group** *acl-id* **in** **示例:** `Device(config-if)#` **ip access-group default_acl in**	在端口输入方向配置默认 ACL。 **注释：***acl-id* 是访问列表的命令或编号。
步骤 8	**show running-config interface** *interface-id* **示例:** `Device#` **show running-config interface gigabitethernet2/0/1**	验证配置。
步骤 9	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。

配置可下载的策略

在特权 EXEC 模式中执行以下配置：

总步骤

1. configure terminal

2. access-list *access-list-number* **{ deny | permit } { hostname | any | host } log**

3. interface *interface-id*

4. ip access-group *acl-id* **in**

5. exit

6. aaa new-model

7. aaa authorization network default group radius

8. ip device tracking

9. ip device tracking probe [**count** | **interval** | **use-svi**]

10. radius-server vsa send authentication

11. end

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：**	进入全局配置模式。

	`Device#` **configure terminal**	
步骤 2	**access-list** *access-list-number* **{ deny | permit } { hostname | any | host } log** **示例:** `Device(config)#` **access-list 1 deny any log**	定义默认的端口 ACL。 *access-list-number* 是十进制数，从 1 到 99 或 1300 到 1999。 输入 **deny** 或 **permit** 指定匹配后拒绝或允许访问。 源是发送数据包的网络或主机源地址，如： • **hostname**：32 位的点分十进制数。 • **any**：此关键字是源以及源通配值 0.0.0.0 255.255.255.255 的缩写。无需输入源通配值。 • **host**：此关键字是源以及源通配 0.0.0.0 的缩写。 （可选）把源通配应用到源。 （可选）输入 log 把数据包匹配条目的通知信息发送给控制台。
步骤 3	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet2/0/2**	进入接口配置模式。
步骤 4	**ip access-group** *acl-id* **in** **示例:** `Device(config-if)#` **ip access-group default_acl in**	配置端口进入方向的默认ACL。 **注释：***acl-id*是访问列表的名称或编号。
步骤 5	**exit** **示例:** `Device(config-if)#` **exit**	返回全局配置模式。
步骤 6	**aaa new-model** **示例:** `Device(config)#` **aaa new-model**	启用 AAA。
步骤 7	**aaa authorization network default group**	设置本地授权方式。要移除授权方式，

	radius **示例:** `Device(config)#` **aaa authorization network default group radius**	使用命令 **no aaa authorization network default group radius**。
步骤 8	**ip device tracking** **示例:** `Device(config)#` **ip device tracking**	启用 IP 设备追踪表。 要禁用 IP 设备追踪表，使用全局配置命令 **no ip device tracking**。
步骤 9	**ip device tracking probe** [**count** | **interval** | **use-svi**] **示例:** `Device(config)#` **ip device tracking probe count**	（可选）配置 IP 设备追踪表： • **count** *count*——设置交换机发送 ARP 探测帧的次数。范围从 1 到 5，默认值是 3。 • **interval** *interval*——设置交换机重发 ARP 探测帧之前等待的秒数。范围从 30 到 300 秒，默认值是 30 秒。 • **use-svi**——使用交换机虚接口（SVI）的 IP 地址作为 ARP 探测帧的源地址。
步 骤 10	**radius-server vsa send authentication** **示例:** `Device(config)#` **radius-server vsa send authentication**	配置网络接入服务器识别并使用厂商特定的属性。 **注释：**可下载的 ACL 必须可操作。
步 骤 11	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。

配置基于 VLAN ID 的 MAC 认证

在特权 EXEC 模式中按照以下步骤进行配置。

总步骤

1. configure terminal

2. mab request format attribute 32 vlan access-vlan

3. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**mab request format attribute 32 vlan access-vlan** **示例:** `Device(config)#` **mab request format attribute 32 vlan access-vlan**	启用基于 VLAN ID 的 MAC 认证。
步骤 3	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。

配置灵活认证顺序

以下示例的命令改变了灵活认证排序的顺序，让 MAB 在 IEEE 802.1x 认证（dot1x）之前尝试。MAB 被配置为第一个认证方式，所以将优先于所有其他的认证方式。

注释： 在更改默认认证方式的顺序以及优先级时，应该理解更改操作的潜在后果。详情参见http://www.icntnetworks.com

在特权 EXEC 模式中按照以下步骤进行配置。

总步骤

1. configure terminal

2. interface *interface-id*

3. switchport mode access

4. authentication order [**dot1x** | **mab**] | {**webauth**}

5. authentication priority [**dot1x** | **mab**] | {**webauth**}

6. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal**	进入全局配置模式。

	示例： `Device#` **configure terminal**	
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet 1/0/1**	指定要配置的端口，并进入接口配置模式。
步骤 3	**switchport mode access** **示例:** `Device(config-if)#` **switchport mode access**	仅在配置了 RADIUS 服务器后把端口设置为接入模式。
步骤 4	**authentication order** [**dot1x** | **mab**] | {**webauth**} **示例:** `Device(config-if)#` **authentication order mab dot1x**	（可选）设置端口上使用的认证方式顺序。
步骤 5	**authentication priority** [**dot1x** | **mab**] | {**webauth**} **示例:** `Device(config-if)#` **authentication priority mab dot1x**	（可选）为端口优先级列表添加认证方式。
步骤 6	**end** **示例:** `Device(config-if)#` **end**	返回特权 EXEC 模式。

配置 Open1x

在特权 EXEC 模式中，按照以下步骤手工控制端口的授权状态。

总步骤

1. configure terminal

2. interface *interface-id*

3. switchport mode access

4. authentication control-direction {**both** | **in**}

5. authentication fallback *name*

6. authentication host-mode [**multi-auth** | **multi-domain** | **multi-host** | **single-host**]

7. authentication open

8. authentication order [**dot1x** | **mab**] | {**webauth**}

9. authentication periodic

10. authentication port-control {**auto** | **force-authorized** | **force-un authorized**}

11. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet 1/0/1**	指定要配置的端口，并进入接口配置模式。
步骤 3	**switchport mode access** **示例:** `Device(config-if)#` **switchport mode access**	仅在配置了 RADIUS 服务器后把端口设置为接入模式。
步骤 4	**authentication control-direction** {**both** | **in**} **示例:** `Device(config-if)#` **authentication control-direction both**	（可选）配置单向或双向的端口控制。
步骤 5	**authentication fallback** *name* **示例:** `Device(config-if)#` **authentication fallback profile1**	（可选）配置端口为不支持 802.1x 认证的客户端使用网页认证作为备用方式。
步骤 6	**authentication host-mode** [**multi-auth** | **multi-domain** |	（可选）设置端口上的授权管理器模式。

	**multi-host	single-host]** **示例:** `Device(config-if)#` **authentication host-mode multi-auth**	
步骤 7	**authentication open** **示例:** `Device(config-if)#` **authentication open**	（可选）在端口上启用或禁用开放访问。
步骤 8	**authentication order [dot1x	mab]	{webauth}** **示例:** `Device(config-if)#` **authentication order dot1x webauth**	（可选）设置端口使用的认证方式顺序。
步骤 9	**authentication periodic** **示例:** `Device(config-if)#` **authentication periodic**	（可选）在端口上启用或禁用重新认证。
步 骤 10	**authentication port-control {auto	force-authorized	force-un authorized}** **示例:** `Device(config-if)#` **authentication port-control auto**	（可选）启用端口授权状态的手工控制。
步 骤 11	**end** **示例:** `Device(config-if)#` **end**	返回特权 EXEC 模式。

在端口上禁用 802.1x 认证

可以使用接口配置命令 **no dot1x pae** 在端口上禁用 802.1x 认证。

在特权 EXEC 模式中，按照以下步骤在端口上禁用 802.1x 认证。此过程是可选的。

总步骤

1. configure terminal

2. interface *interface-id*

3. switchport mode access

4. no dot1x pae authenticator

5. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet 2/0/1**	指定要配置的端口，并进入接口配置模式。
步骤 3	**switchport mode access** **示例:** `Device(config-if)#` **switchport mode access**	（可选）仅在配置了 RADIUS 服务器后把端口设置为接入模式。
步骤 4	**no dot1x pae authenticator** **示例:** `Device(config-if)#` **no dot1x pae authenticator**	在端口上禁用 802.1x 认证。
步骤 5	**end** **示例:** `Device(config-if)#` **end**	返回特权 EXEC 模式。

把 802.1x 认证配置重置为默认值

在特权 EXEC 模式中，按照以下步骤把 802.1x 认证配置重置为默认值。此过程是可选的。

总步骤

1. configure terminal

2. interface *interface-id*

3. dot1x default

4. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet 1/0/2**	指定要配置的端口，并进入接口配置模式。
步骤 3	**dot1x default** **示例:** `Device(config-if)#` **dot1x default**	重置 802.1x 参数为默认值。
步骤 5	**end** **示例:** `Device(config-if)#` **end**	返回特权 EXEC 模式。

监控 802.1x 统计信息及状态

表 149：特权 EXECshow 命令

命令	目的
show dot1x all statistics	显示所有端口的 802.1x 统计信息。
show dot1x interface *interface-id* **statistics**	显示特定端口的 802.1x 统计信息。
show dot1x all [**count** | **details** | **statistics** | **summary**]	显示交换机的 802.1x 管理状态及运行状态。
show dot1x interface *interface-id*	显示特定端口的 802.1x 管理状态及运行状态。

表 150：全局配置命令

命令	目的
no dot1x logging verbose	过滤详细的 802.1x 认证消息（从 Inspur INOS 12.2(55)SE 版本之后支持）。

有关显示字段的详细信息，查看此版本的命令手册。

其他参考资料

相关文档

相关主题	文档标题
为会话感知网络配置身份控制策略以及身份服务模板。	Inspur INOS 会话感知网络配置指南（Inspur 6850 交换机） http://www.icntnetworks.com
配置 RADIUS、TACACS+、安全 Shell、802.1x 以及 AAA。	Inspur INOS 保护用户服务配置指南库（Inspur 6850 交换机） http://www.icntnetworks.com

错误信息解释

描述	链接
为帮助管理员搜索并解决该版本中的系统错误信息，管理员可使用错误信息解释工具。	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	http://www.icntnetworks.com

IPv4 访问控制列表的特性信息

版本	特性信息
Inspur INOS 11.3.1	IPv4 访问控制列表执行数据包过滤，控制数据包在网络之间的移动。其控制功能限制网络流量，约束用户及设备的网络访问，并阻止流量离开网络，进而提供安全性。
Inspur INOS 11.3.1	命名的 ACL 允许访问控制条目使用非连续的端口号，让管理员可以在一个访问控制条目中指定非连续的端口，极大地减少了访问列表中源地址、目的地址以及协议都相同，但是端口不同的条目数量。

配置基于端口的流量控制

基于端口的流量控制概述

基于端口的流量控制是 Inspur 交换机上的一组二层特性，可以用来在端口级别过滤或阻塞满足特定流量条件的数据包。本配置指南所述的 Inspur INOS 版本支持的基于端口的流量控制特性如下：

- 风暴控制
- 保护端口
- 端口阻塞
- 端口安全
- 协议风暴保护

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于风暴控制的信息

风暴控制

风暴控制特性防止 LAN 上的流量因为物理端口上的广播、组播或者单播风暴而中断。LAN 风暴发生时，数据包在 LAN 中被泛洪，产生了过量的流量，并导致网络性能降级。造成风暴的原因可能是协议栈实现中的错误，网络配置的错误，还可能是用户发起了拒绝服务攻击。风暴控制（或称流量抑制）监控从接口发往交换总线的数据包，并确定数据包是单播、组播、还是广播的。交换机会记录 1 秒的间隔时间内接收的特定类型的数据包数量，并将其与预定义的抑制等级门限值进行比较。

如何测量流量活动

风暴控制特性使用以下方式之一来测量流量活动：

- 广播、组播以及单播流量能够使用的端口总可用带宽的百分比。
- 接收广播、组播以及单播流量的流量速率，单位是数据包每秒。
- 接收广播、组播以及单播流量的流量速率，单位是比特每秒。
- 小数据帧的流量速率，单位是数据包每秒。此特性全局启用。会为每个接口配置小数据帧的门限值。

对于每种方式，速率达到上升门限值时，端口会阻塞流量。端口保持阻塞，直到流量速率下降到下降门限值以下，端口恢复正常的转发。如果没有指定下降抑制等级，在流量速率下降到上升抑制等级以下之前，交换机会阻塞所有流量。通常来说，抑制等级越高，对广播风暴的防护效果越小。

注释： 达到组播流量的风暴控制门限值时，除了如网桥协议数据单元（BPDU）以及 Inspur 发现协议（CDP）这样的控制流量，所有的组播流量都会被阻塞。然而，交换机不会区分如 OSPF 这样的路由更新与常规的组播数据流量，所以这两类流量都会被阻塞。

流量模式

以下示例展示了特定时间段内接口上的广播流量模式。

图 130：广播风暴控制示例

Forwarded traffic	转发流量
Blocked traffic	阻塞流量
Threshold	门限值
Time	时间
Total number of broadcast packets or bytes	广播的数据包或字节总数

在 T1、T2 以及 T4、T5 时间间隔内，转发的广播流量超过了配置的门限值。当特定流量总量超过门限值时，该类型的所有流量在下一个时间段内都会被丢弃。因此，在之后的 T2 和 T5 的间隔内，广播流量被阻塞。在下一个时间间隔中（如 T3），如果广播流量没有超过门限值，它将再次被转发。

风暴控制抑制等级与 1 秒时间间隔组合起来控制了风暴控制算法的工作方式。更高的门限值允许更多的数据包通过。门限值 100%表示不对流量进行限制，0.0 表示端口上的所有广播、组播或单播流量都会被阻塞。

注释： 因为数据包不会按照均匀的间隔到达，在测量流量活动时 1 秒的时间间隔就可以影响风暴控制的行为。

可以使用接口配置命令 **storm-control** 设置每类流量的门限值。

如何配置风暴控制

配置风暴控制及门限值等级

可以在端口上配置风暴控制并输入希望为特定类型流量使用的门限值等级。

然而，因为硬件的限制以及对不同尺寸的数据包统计方式不同，门限值的百分比是近似值。根据组成入向流量的数据包大小不同，实际使用的门限值可与配置的等级有几个百分点的差别。

注释： 风暴控制支持在物理端口上使用。也可以在 EtherChannel 上配置风暴控制。对 EtherChannel 进行配置时，风暴控制设置会传播到 EtherChannel 物理接口上。

按照以下步骤配置风暴控制以及门限值等级。

在开始前

风暴控制支持在物理端口上使用。也可以在 EtherChannel 上配置风暴控制。对 EtherChannel 进行配置时，风暴控制设置会传播到 EtherChannel 物理接口上。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. storm-control {**broadcast** | **multicast** | **unicast**} **level** {*level* [*level-low*] | **bps** *bps*[*bps-low*] | **pps***pps*[*pps-low*]}

5. storm-control action {**shutdown** | **trap**}

6. end

7. show storm-control [*interface-id*] [**broadcast** | **multicast** | **unicast**]

8. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：**	进入全局配置模式。

	`Device#` **configure terminal**	
步骤 3	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet1/0/1**	指定要被配置的接口，并进入接口配置模式。
步骤 4	**storm-control** {**broadcast** | **multicast** | **unicast**} **level** {*level* [*level-low*] | **bps** *bps* [*bps-low*] | **pps***pps*[*pps-low*]} **示例:** `Device(config-if)#`**storm-control unicast level 87 65**	配置广播、组播或单播风暴控制。默认情况下，风暴控制被禁用。 关键字含义如下： • *level* 字段以带宽百分比的形式（最多到小数点后两位）指定广播、组播或单播流量的上升门限值等级。速率达到上升门限值时端口阻塞流量。范围从 0.00 到 100.00。 • （可选）*level-low* 字段以带宽百分比的形式（最多到小数点后两位）指定下降门限值等级。该值必须小于或等于上升抑制值。端口在流量降到该等级以下时进行流量转发。如果不配置下降抑制等级，其被设置为上升抑制等级。范围从 0.00 到 100.00。 如果设置门限值为最大值（100%），对流量不进行限制。如果设置门限值为 0.0，端口上的所有广播、组播以及单播流量都会被阻塞。 • **bps** *bps* 字段以比特每秒（最多到小数点后一位）指定广播、组播或单播流量的上升门限值等级。速率达到上升门限值时端口阻塞流量。范围从 0.0 到 10000000000.0。

		• （可选）*bps-low* 字段以比特每秒（最多到小数点后一位）指定下降门限值等级。该值必须小于或等于上升抑制值。端口在流量降到该等级以下时进行流量转发。范围从 0.0 到 10000000000.0。 • **pps***pps* 字段以数据包每秒（最多到小数点后一位）指定广播、组播或单播流量的上升门限值等级。速率达到上升门限值时端口阻塞流量。范围从 0.0 到 10000000000.0。 • （可选）*pps-low* 以数据包每秒（最多到小数点后一位）指定下降门限值等级。该值必须小于或等于上升抑制值。端口在流量降到该等级以下时进行流量转发。范围从 0.0 到 10000000000.0。 对于 BPS 及 PPS 设置，较大的门限值数可以使用度量后缀，如 k、m 和 g。
步骤 5	**storm-control action {shutdown | trap}** **示例:** `Device(config-if)#`**storm-control action trap**	指定检测到风暴时要采取的行为。默认行为是过滤流量且不发送陷阱（trap）。 • 选择 **shutdown** 关键字，在风暴期间设置端口为错误禁用。 • 选择 **trap** 关键字，在检测到风暴时生成 SNMP 陷阱。
步骤 6	**end** **示例:** `Device(config-if)#` **end**	返回特权 EXEC 模式。

步骤 7	**show storm-control** [*interface-id*] [**broadcast** | **multicast** | **unicast**] **示例:** `Device#` **show storm-control gigabitethernet1/0/1 unicast**	验证在端口上为特定的流量类型设置的风暴控制抑制等级。如果不输入流量类型，会显示广播风暴控制的设置。
步骤 8	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。

配置小数据帧的到达速率

入向有 VLAN 标记且小于 67 字节的数据包被认为是小数据帧。它们会被交换机转发，但不会使交换机风暴控制计数器增加。

可以在交换机上全局启用小数据帧到达特性，为每个接口上的数据包配置小数据帧门限值。小于最小大小且按照特定速率（门限值）到达的数据包会被丢弃，因为此时端口会被错误禁用。

总步骤

1. enable

2. configure terminal

3. errdisable detect cause small-frame

4. errdisable recovery interval *interval*

5. errdisable recovery cause small-frame

6. interface *interface-id*

7. small-frame violation-rate *pps*

8. end

9. show interfaces *interface-id*

10. show running-config

11. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：**	进入特权 EXEC 模式。在提示时输入密码。

	Device>**enable**	
步骤 2	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式。
步骤 3	**errdisable detect cause small-frame** **示例:** Device(config)# **errdisable detect cause small-frame**	在交换机上启用小数据帧到达速率特性。
步骤 4	**errdisable recovery interval** *interval* **示例:** Device(config)# **errdisable recovery interval60**	（可选）指定从错误禁用状态恢复的时间。
步骤 5	**errdisable recovery cause small-frame** **示例:** Device(config)# **errdisable recovery cause small-frame**	（可选）配置错误禁用端口的恢复时间，使端口因到达的小数据帧被错误禁用后能自动重新启用。 风暴控制支持在物理端口上使用。也可以在 EtherChannel 上配置风暴控制。对 EtherChannel 进行配置时，风暴控制设置会传播到 EtherChannel 物理接口上。
步骤 6	**interface** *interface-id* **示例:** Device(config)# **interface gigabitethernet1/0/2**	指定要配置的接口，并进入接口配置模式。
步骤 7	**small-frame violation-rate** *pps* **示例:** Device(config-if)#**small-frame violation rate 10000**	配置接口丢弃入向数据包并被错误禁用的门限值。范围从 1 到 10000 包每秒（pps）。
步骤 8	**end** **示例:** Device(config)# **end**	返回特权 EXEC 模式。

步骤 9	**show interfaces** *interface-id* **示例:** `Device#` **show interfaces gigabitethernet1/0/2**	验证配置。
步 骤 10	**show running-config** **示例:** `Device#` **show running-config**	验证配置的条目。
步 骤 11	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。

关于保护端口的信息

保护端口

一些应用会要求二层流量不在相同交换机的端口之间进行转发，使得一个邻居看不到另一个邻居产生的流量。在这样的环境中，使用保护端口能保证交换机的这些端口之间没有单播、组播或广播流量交换。

保护端口的特性如下：

- 保护端口不会把任何流量（单播、组播或广播）转发给其他的保护端口。数据流量在二层上不能在保护端口之间转发；只有如 PIM 包这样的控制流量会被转发，因为这些包会被 CPU 处理并在软件中转发。所有保护端口之间的数据流量必须通过三层设备进行转发。
- 保护端口和非保护端口之间的转发行为照常进行。

因为一个交换机堆栈代表一台逻辑交换机，所以二层流量不会在交换机堆栈中的保护端口之间转发，无论保护端口是否在堆栈中相同的交换机上。

保护端口的默认配置

默认无保护端口定义。

保护端口指南

可以在物理接口（如吉比特以太网端口 1）或者 EtherChannel 群组（如端口通道 5）上配置保护端口。对端口通道启用保护端口，该特性会为端口通道组中的所有端口启用。

如何配置保护端口

配置保护端口

在开始前

没有预定义的保护端口。以下将配置一个保护端口。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. switchport protected

5. end

6. show interfaces *interface-id* **switchport**

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**interface** *interface-id* **示例:** `Device(config)#` **interface**	指定要配置的接口，并进入接口配置模式。

	gigabitethernet1/0/1	
步骤 4	**switchport protected** **示例:** `Device(config-if)#` **switchport protected**	配置接口为保护端口。
步骤 5	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 6	**show interfaces** *interface-id* **switchport** **示例:** `Device#` **show interfaces gigabitethernet1/0/1 switchport**	验证配置。
步骤 7	**show running-config** **示例:** `Device#` **show running-config**	验证配置的条目。
步骤 8	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。

监控保护端口

表 153：显示保护端口设置

命令	目的
show interfaces [*interface-id*] **switchport**	显示所有交换（非路由）端口或特定端口的管理状态以及运行状态，包括端口阻塞和端口保护设置。

有关端口阻塞的信息

端口阻塞

默认情况下，交换机会把带有未知目的 MAC 地址的数据包从所有端口泛洪出去。如果未知的单播或组播流量被转发到了保护端口上，可能会造成安全问题。为了避免未知的单播或组播流量被从一个端口转发到另一个端口，管理员可以阻止一个端口（保护或非保护端口）把未知的单播或组播数据包发往其他端口。

注释： 对于组播流量，端口阻塞特性只会阻止纯二层的数据包。报头中包含 IPv4 或 IPv6 信息的组播数据包不会被阻塞。

如何配置端口阻塞

阻塞端口上泛洪流量

在开始前

接口可以是物理接口，也可以是 EtherChannel 组。配置阻塞端口通道上的组播或单播流量时，端口通道组中的所有端口都会进行阻塞。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. switchport block multicast

5. switchport block unicast

6. end

7. show interfaces *interface-id* **switchport**

8. show running-config

9. copy running-config startup-config

具体步骤

	命令或操作	目的

步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet1/0/1**	指定要配置的接口，并进入接口配置模式。
步骤 4	**switchport block multicast** **示例:** `Device(config-if)#` **switchport block multicast**	阻塞从端口转发出的未知组播流量。 **注释：**只阻塞纯二层组播流量。报头中包含 IPv4 或 IPv6 信息的组播数据包不会被阻塞。
步骤 5	**switchport block unicast** **示例:** `Device(config-if)#` **switchport block unicast**	阻塞从端口转发出的未知单播流量。
步骤 6	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 7	**show interfaces** *interface-id* **switchport** **示例:** `Device#` **show interfaces gigabitethernet1/0/1 switchport**	验证配置。
步骤 8	**show running-config** **示例:** `Device#` **show running-config**	验证配置的条目。
步骤 9	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。

监控端口阻塞

表 154：显示端口阻塞特性设置的命令

命令	目的
show interfaces [*interface-id*] **switchport**	显示所有交换（非路由）端口或特定端口的管理状态以及运行状态，包括端口阻塞和端口保护设置。

端口安全的前提条件

注释： 如果尝试设置的最大数值小于已经在端口上配置的安全地址数量，配置的命令会被拒绝。

端口安全的限制

可以在交换机或者交换机堆栈上配置的最大安全 MAC 地址数量由系统中允许的最大可用 MAC 地址数量决定。此数值由活跃的交换机数据库管理（Switch Database Management，SDM）模板决定。此数值是总的可用 MAC 地址数量，包括用于其他二层功能的地址以及接口上配置的任何其他安全 MAC 地址。

有关端口安全的信息

端口安全

使用端口安全特性，可以限制并标识允许访问端口的工作站的 MAC 地址，进而限制对端口的输入。在为安全端口分配安全 MAC 地址时，该端口不会转发源地址在定义的地址组之外的数据包。如果把安全 MAC 地址的数量限制为 1 并分配了一个安全 MAC 地址，能够确保连接到端口的工作站使用端口的全部带宽。

如果端口被配置为安全端口，且到达了其配置的最大安全 MAC 地址数量，当尝试访问端口的工作站的 MAC 地址与已标识的安全 MAC 地址都不同时，会发生安全违规事件。同时，如果在一个安全端口上配置或者学习到的工作站安全 MAC 地址尝试访问另一个安全端口时，违规事件会被标记。

安全 MAC 地址类型

交换机支持以下类型的安全 MAC 地址：

- 静态安全MAC地址——使用接口配置命令**switchport port-securitymac-address** *mac-address*手动配置，保存在地址表中，且被添加到交换机的运行配置中。
- 动态安全 MAC 地址——动态配置，仅保存在地址表中，且在交换机重启时会被移除。
- 粘性安全 MAC 地址——可以动态学习或手工配置，保存在地址表中，且被添加到运行配置。如果这些地址被保存在配置文件中，当交换机重启时，接口无需重新动态配置这些地址。

粘性安全 MAC 地址

可以启用粘性学习功能，配置接口把动态 MAC 地址转换为粘性安全 MAC 地址，并将其添加到运行配置中。接口会把所有动态安全 MAC 地址，包括启用粘性学习之前动态学习到的地址转换为粘性安全 MAC 地址。所有粘性安全 MAC 地址都会被添加到运行配置中。

粘性安全 MAC 地址不会自动保存到交换机重启时使用的启用配置文件中。如果把粘性安全 MAC 地址保存在配置文件中，当交换机重启时，接口无需重新学习这些地址。如果不保存粘性安全地址，它们会丢失。

如果禁用了粘性学习功能，粘性安全 MAC 地址会被转换为动态安全地址，且会被从运行配置中移除。

安全违规

以下情况之一发生时会造成安全违规：

- 已经向地址表中添加了最大数量的安全 MAC 地址，而有 MAC 地址不在地址表中的工作站尝试访问接口。
- 在一个安全接口上学习到或配置的地址在相同 VLAN 中的另一个安全接口上被发现。

可以基于违规发生时要采取的行为，将接口配置为以下三种违规模式之一：

- 保护——当安全 MAC 地址的数量达到了端口允许的最大限制时，源地址未知的数据包会被丢弃，直到移除了足量的安全 MAC 地址或者增加了允许的最大地址数量。安全违规事件发生时不会通知用户。

注释： 不建议在中继端口上配置保护违规模式。当任意 VLAN 达到了其最大限制时，即使端口没有达到最大限制，保护模式也会禁用学习功能。

- 限制——当安全 MAC 地址的数量达到了端口允许的最大限制时，源地址未知的数据包会被丢弃，直到移除了足量的安全 MAC 地址或者增加了允许的最大地址数量。在此模式中，安全违规事件发生时会通知用户。会发送 SNMP 陷阱，记录 syslog 消息，且违规计数器会增加。
- 关闭——端口安全违规事件会导致接口成为错误禁用状态或被立即关闭，且端口的 LED 灯会关闭。当安全端口在错误禁用状态时，可以输入全局配置命令 **errdisable recovery cause** *psecure-violation* 将端口移出此状态，也可以输入接口配置命令 **shutdown** 及 **no shutdown** 手动重新启用接口。这是默认的模式。
- 关闭 VLAN——基于 VLAN 设置安全违规模式。在此模式中，违规事件发生时 VLAN 会被错误禁用，而端口不会被禁用。

下表显示了端口安全配置的违规模式以及采取的行为。

表 155：安全违规模式行为

违规模式	**转发流量**[19]	**发送 SNMP 陷阱**	**发送 syslog 消息**	**显示错误消息**[20]	**增加违规计数器**	**关闭端口**
保护	否	否	否	否	否	否
限制	否	是	是	否	是	否
关闭	否	否	否	否	是	是
关闭 VLAN	否	否	是	否	是	否[21]

[19] 源地址未知的数据包会被丢弃，直到移除了足量的安全 MAC 地址。

[20] 如果手动配置了可能造成安全违规的地址，交换机会返回错误消息。

[21] 只关闭违规发生的 VLAN。

端口安全老化特性

可以使用端口安全老化特性设置端口上所有安全地址的老化时间。每个端口上支持两种类型的老化方式：

- 绝对——在特定的老化时间过后端口上的安全地址会被删除。

- 不活跃——如果在特定的老化时间内安全地址不活跃，端口上的该安全地址会被删除。

端口安全与交换机堆栈

当一台交换机加入堆栈时，新的交换机会获取配置的安全地址。新交换机会从其他堆栈成员上下载所有的动态安全地址。

当一台交换机离开堆栈时（活跃交换机或堆栈成员），其余的堆栈成员会被通知，且该交换机配置或学习到的安全 MAC 地址会被从安全 MAC 地址表中删除。

默认的端口安全配置

表 156：默认的端口安全配置

特性	**默认设置**
端口安全	在端口上禁用。
粘性地址学习	禁用。
每个端口的最大安全地址数量	1。
违规模式	关闭。超过最大安全 MAC 地址数量时端口会被关闭。
端口安全老化	禁用。老化时间是 0。 静态老化被禁用。 类型为绝对老化。

端口安全配置指南

- 只能在静态端口或中继端口上配置端口安全特性。安全端口不能是动态接入端口。
- 安全端口不能是交换端口分析器（Switched Port Analyzer，SPAN）的目的端口。

注释： 虽然允许进行配置，但语音 VLAN 支持接入端口，不支持中继端口。

- 在配置了语音 VLAN 的接口上启用端口安全时，把端口的最大安全地址数量设置为 2。当端口连接到 Inspur IP 电话时，IP 电话需要使用一个 MAC 地址。Inspur IP 电话的地址会在语音 VLAN 上学习到，不会在接入 CLAN 上学习到。如果把一台 PC 连接到 Inspur IP 电话上，无需额外的 MAC 地址。如果把多台 PC 连接到 Inspur IP 电话上，必须配置足够的安全地址数量，满足每台 PC 机一台电话使用。
- 当中继端口配置了端口安全，且为数据流量分配了一个接入 VLAN，为语音流量分配了

一个语音 VLAN 时，输入 **switchport voice** 以及 **switchport priority extend** 接口配置命令没有效果。

当连接的设备使用相同的 MAC 地址在接入 VLAN 中请求 IP 地址，又在语音 VLAN 中请求 IP 地址时，只会给接入 VLAN 分配 IP 地址。

- 输入接口的最大安全地址数量，且新输入的值大于之前的值时，新值会覆盖之前配置的值。如果接口上已配置的安全地址数量大于新值时，命令被拒绝。
- 交换机不支持对粘性安全 MAC 地址进行端口安全老化操作。

下表总结了端口安全与其他基于端口特性的兼容性。

端口类型或端口特性	**与端口安全的兼容性**
DTP[22] 端口 [23]	否
中继端口	是
动态接入端口 [24]	否
被路由端口	否
SPAN 源端口	是
SPAN 目的端口	否
EtherChannel	是
隧道端口	是
保护端口	是
IEEE 802.1x 端口	是
语音 VLAN 端口 [25]	是
IP 源防护	是
动态地址解析协议（ARP）监测	是
灵活链路	是

[22] DTP=动态中继协议（Dynamic Trunking Protocol）

[23] 使用接口配置命令 **switchport mode dynamic** 配置的端口。

[24] 使用接口配置命令 **switchport access vlan dynamic** 配置的 VLAN 查询协议（VLAN Query Protocol，VQP）端口。

[25] 必须把端口允许的最大安全地址数量设置为 2 加上接入 VLAN 允许的最大安全地址数量。

如何配置端口安全

启用并配置端口安全

在开始前

此设置通过限制并标识允许访问端口的工作站的 MAC 地址来限制接口的输入。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. switchport mode {**access** | **trunk**}

5. switchport voice vlan*vlan-id*

6. switchport port-security

7. switchport port-security [maximum *value* [**vlan**{*vlan-list* | {**access** | **voice**}}]]

8. switchport port-security violation {**protect** | **restrict** | **shutdown** | **shutdown vlan**}

9. switchport port-security [mac-address *mac-address*[**vlan**{*vlan-id* | {**access** | **voice**}}]

10. switchport port-security mac-address sticky

11. switchport port-security mac-address sticky [*mac-address* | **vlan**{*vlan-id* | {**access** | **voice**}}]

12. end

13. show port-security

14. show running-config

15. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**interface** *interface-id*	指定要配置的接口，并进入接口配置

	示例: `Device(config)#` **interface gigabitethernet1/0/1**	模式。
步骤 4	**switchport mode** {**access** | **trunk**} **示例:** `Device(config-if)#`**switchport mode access**	设置接口的模式为接入或中继；默认模式（dynamic auto）中的接口不能配置为安全端口。
步骤 5	**switchport voice vlan***vlan-id* **示例:** `Device(config-if)#`**switchport voice vlan 22**	在端口上启用语音 VLAN。 *vlan-id*——指定用于语音流量的 VLAN。
步骤 6	**switchport port-security** **示例:** `Device(config-if)#`**switchport port-security**	在接口上启用端口安全。
步骤 7	**switchport port-security [maximum** *value* [**vlan**{*vlan-list* | {**access** | **voice**}}]] **示例:** `Device(config-if)#`**switchport port-security maximum 20**	（可选）设置端口的最大安全 MAC 地址数量。可以在交换机或者交换机堆栈上配置的最大安全 MAC 地址数量由系统中允许的最大可用 MAC 地址数量决定。此数值由活跃的交换机数据库管理（Switch Database Management，SDM）模板决定。此数值是总的可用 MAC 地址数量，包括用于其他二层功能的地址以及接口上配置的任何其他安全 MAC 地址。 （可选）**vlan**——基于 VLAN 设置最大值。 输入 **vlan** 关键字之后输入以下选项： • *vlan-list*——在中继端口上可以基于 VLAN 设置最大值，输入连字符分隔的VLAN范围或一组由逗号分隔的 VLAN。对于为指定的 VLAN，

		将使用基于 VLAN 的最大值。 • **access**——在接入端口上指定 VLAN 为接入 VLAN。 • **voice**——在接入端口上指定 VLAN 为语音 VLAN。 **注释：**只有在端口上配置了语音 VLAN 且端口不在接入 VLAN 中时，**voice** 关键字才可用。如果接口配置了语音 VLAN，应配置最大 2 个安全 MAC 地址。
步骤 8	**switchport port-security violation {protect | restrict | shutdown | shutdown vlan}** **示例:** `Device(config-if)#`**switchport port-security violation restrict**	（可选）设置违规模式以及检测到安全违规时要采取的行为： • **protect**——当安全 MAC 地址的数量达到了端口允许的最大限制时，源地址未知的数据包会被丢弃，直到移除了足量的安全 MAC 地址或者增加了允许的最大地址数量。安全违规事件发生时不会通知用户。 **注释：** 不建议在中继端口上配置保护违规模式。当任意 VLAN 达到了其最大限制时，即使端口没有达到最大限制，保护模式也会禁用学习功能。 • **restrict**——当安全 MAC 地址的数量达到了端口允许的最大限制时，源地址未知的数据包会被丢弃，直到移除了足量的安全 MAC 地址或者增加了允许的最大地址数量。在此模式中，安全违规事件发生时会通知用户。会发送 SNMP 陷阱，记录 syslog 消息，且

		违规计数器会增加。 • **shutdown**——端口安全违规事件会导致接口成为错误禁用状态，且端口的 LED 灯会关闭。会发送 SNMP 陷阱，记录 syslog 消息，且违规计数器会增加。 • **shutdown vlan**——基于 VLAN 设置安全违规模式。在此模式中，违规事件发生时 VLAN 会被错误禁用，而端口不会被禁用。 **注释：**安全端口在错误禁用状态时，可以输入全局配置命令**errdisable recovery causepsecure-violation**将端口移出此状态。可以使用接口配置命令**shutdown**及**no shutdown**手动重启端口，或使用特权EXEC命令**clear errdisableinterface vlan**。
步骤 9	**switchport port-security [mac-address** *mac-address*[**vlan** {*vlan-id* | {**access** | **voice**}}] **示例:** `Device(config-if)#`**switchport port-security mac-address 00:A0:C7:12:C9:25 vlan 3 voice**	（可选）为接口输入安全 MAC 地址。可以使用此命令输入源 MAC 地址的最大数量。如果输入的值小于最大安全 MAC 地址数，其余的 MAC 地址可以动态学习。 **注释：**如果输入此命令后启用了粘性学习，动态学习到的安全地址会被转换为粘性安全 MAC 地址并被添加到运行配置中。 （可用）**vlan**——基于 VLAN 设置最大值。 输入 **vlan** 关键字之后输入以下选项： • *vlan-list*——在中继端口上可以指定 VLAN ID 以及 MAC 地址。如果

		不指定 VLAN ID，将使用本征 VLAN。 • **access**——在接入端口上指定 VLAN 为接入 VLAN。 • **voice**——在接入端口上指定 VLAN 为语音 VLAN。 **注释：**只有在端口上配置了语音 VLAN 且端口不在接入 VLAN 中时，**voice** 关键字才可用。如果接口配置了语音 VLAN，应配置最大 2 个安全 MAC 地址。
步骤 10	**switchport port-security mac-address sticky** **示例:** `Device(config-if)#`**switchport port-security mac-address sticky**	（可选）在接口上启用粘性学习。
步骤 11	**switchport port-security mac-address sticky** [*mac-address* | **vlan**{*vlan-id* | {**access** | **voice**}}] **示例:** `Device(config-if)#`**switchport port-security mac-address sticky 00:A0:C7:12:C9:25 vlan voice**	（可选）输入粘性安全 MAC 地址，按需重复输入命令。如果输入的数量小于最大安全 MAC 地址数量，其余 MAC 地址会自动学习，并被转换为粘性安全 MAC 地址，且被添加到运行配置中。 **注释：**如果在输入此命令前未启用粘性学习，会显示错误消息，且无法输入粘性安全 MAC 地址。 （可选）**vlan**——基于 VLAN 设置最大值。 输入 **vlan** 关键字之后输入以下选项： • *vlan-list*——在中继端口上可以指定 VLAN ID 以及 MAC 地址。如果不指定 VLAN ID，将使用本征 VLAN。 • **access**——在接入端口上指定 VLAN 为接入 VLAN。

		• **voice**——在接入端口上指定 VLAN 为语音 VLAN。 **注释：**只有在端口上配置了语音 VLAN 且端口不在接入 VLAN 中时，**voice** 关键字才可用。
步 骤 12	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步 骤 13	**show port-security** **示例:** `Device#` **show port-security**	验证配置的条目。
步 骤 14	**show running-config** **示例:** `Device#` **show running-config**	验证配置的条目。
步 骤 15	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。

启用并配置端口安全老化

使用此特性可以在安全端口上移除并添加设备，无需手动删除现有的安全 MAC 地址，且仍可以限制端口上安全地址的数量。可以基于端口启用或禁用安全地址的老化功能。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. switchport port-security aging {**static** | **time** *time*| **type** {**absolute** | **inactivity**}}

5. end

6. show port-security [**interface** *interface-id*] [**address**]

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet1/0/1**	指定要配置的接口，并进入接口配置模式。
步骤 4	**switchport port-security aging** {**static** | **time** *time* | **type** {**absolute** | **inactivity**}} **示例:** `Device(config-if)#` **switchport port-security aging time 120**	启用或禁用安全端口的静态老化，或设置老化时间及类型。 **注释：**交换机不支持粘性安全端口的端口安全老化。 在端口上输入 **static** 来启用静态配置的安全地址的老化。 *time* 字段指定端口的老化时间。合法的范围从 0 到 1440 分钟。 **type** 字段可以选择以下关键字： • **absolute**——设置老化类型为绝对老化。端口上的所有安全地址在指定的时间后会老化且被从安全地址列表中移除。 • **inactivity**——设置老化类型为不活跃老化。只有特定时间内没有来自安全源地址的数据流量，安全地址才会老化。
步骤 5	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 6	**show port-security** [**interface** *interface-id*]	验证配置。

	[**address**] **示例:** `Device#` **show port-security interface gigabitethernet1/0/1**	
步骤 7	**show running-config** **示例:** `Device#` **show running-config**	验证配置的条目。
步骤 8	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置保存在配置文件中。

端口安全配置示例

以下示例显示了如何在端口上启用端口安全，并设置最大安全地址数量为 50。违规模式为默认，未配置静态安全 MAC 地址，且启用了粘性学习。

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)#switchport mode access
Device(config-if)#switchport port-security
Device(config-if)#switchport port-security maximum 50
Device(config-if)#switchport port-security mac-address sticky
```

以下示例显示了如何在端口的 VLAN 3 上配置静态的安全 MAC 地址：

```
Device(config)# interface gigabitethernet1/0/2
Device(config-if)#switchport mode trunk
Device(config-if)#switchport port-security
Device(config-if)#switchport port-security mac-address 0000.0200.0004 vlan 3
```

以下示例显示了如何在端口上启用粘性端口安全特性，手动为数据 VLAN 以及语音 VLAN 配置 MAC 地址，并把安全地址的最大数量设置为 20（数据 VLAN 10 个，语音 VLAN10 个）。

```
Device(config)# interface tengigabitethernet1/0/1
Device(config-if)#switchport access vlan 21
Device(config-if)#switchport mode access
Device(config-if)#switchport voice vlan 22
Device(config-if)#switchport port-security
```

```
Device(config-if)#switchport port-security maximum 20
Device(config-if)#switchport port-security violation restrict
Device(config-if)#switchport port-security mac-address sticky
Device(config-if)#switchport port-security mac-address sticky 0000.0000.0002
Device(config-if)#switchport port-security mac-address 0000.0000.0003
Device(config-if)#switchport port-security mac-address sticky 0000.0000.0001 vlan voice
Device(config-if)#switchport port-security mac-address 0000.0000.0004 vlan voice
Device(config-if)#switchport port-security maximum 10 vlan access
Device(config-if)#switchport port-security maximum 10 vlan voice
```

其他参考资料

错误信息解释

描述	**链接**
为帮助管理员搜索并解决该版本中的系统错误信息，管理员可使用错误信息解释工具。	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	http://www.icntnetworks.com

配置 IPv6 第一跳安全

•

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

IPv6 第一跳安全的前提

- 已配置了必要的 IPv6 SDM 模板。
- 熟悉 IPv6 的邻居发现特性。

IPv6 第一跳安全的限制

- 把 FHS 策略应用到 EtherChannel 接口（端口通道）时存在以下限制：
  - 配置了 FHS 策略的物理端口不能加入 EtherChannel 组。
  - 物理端口是 EtherChannel 组成员时，不能为其配置 FHS 策略。
- 默认情况下，侦听策略有防护的安全等级。在接入层交换机上配置这样的侦听策略时，即使面向路由器或 DHCP 服务器/中继的上行链路端口被配置为可信端口，外部的 IPv6 路由器通告（Router Advertisement，RA）或 IPv6 的动态主机配置协议（Dynamic Host

Configuration Protocolfor IPv6，DHCPv6）服务器数据包也会被阻塞。要允许 IPv6 RA 或 DHCPv6 服务器消息，需执行以下操作：

- 在上行链路端口上应用 IPv6 RA 防护策略（对于 RA）或 IPv6 DHCP 防护策略（对于 DHCP 服务器消息）。
- 配置较低安全等级的侦听策略，如收集或监测。然而，不建议为这样的安全策略配置较低的安全等级，因为这时第一跳安全特性就不再有效。

关于 IPv6 第一跳安全的信息

IPv6 第一跳安全（First Hop Security in IPv6，FHS IPv6）是一组 IPv6 的安全特性，其策略可以配置到物理接口或 VLAN 上。IPv6 软件策略数据库服务存储并访问这些策略。配置或更新策略时，策略的属性会被存储或更新到软件策略数据库中，然后再按配置进行应用。当前支持以下 IPv6 策略：

- IPv6 侦听策略——IPv6 侦听策略作为策略容器，让 FHS IPv6 中大多数特性可以使用。
- IPv6 FHS 绑定表内容——通过来自邻居发现（Neighbor Discovery，ND）协议侦听等信息源的信息，可以创建一个连接到交换机的 IPv6 邻居数据库。这个数据库（或称绑定表）会被多种 IPv6 防护特性（如 IPv6 ND 监测）用来验证链路层地址（link-layeraddress，LLA）、IPv4 或 IPv6 地址以及邻居的前缀绑定信息，以防止伪造或重定向攻击。
- IPv6 邻居发现监测——IPv6 ND 监测特性会学习并保护二层邻居表中的无状态自动配置地址绑定信息。IPv6 ND 监测会分析邻居发现消息以构建可信绑定表数据库，而不合规的 IPv6 邻居发现消息会被丢弃。如果可以验证一个 ND 消息的 IPv6 到介质访问控制（Media AccessControl，MAC）映射信息，则该消息被认为是可信的。

  此特性缓解了 ND 机制的一些固有弱点，比如可能产生对 DAD、地址解析、路由器发现以及邻居缓存的攻击。
- IPv6 路由器通告防护——IPv6 路由器通告（RA）防护特性让管理员可以阻塞或拒绝到达网络交换机平台上的不希望的流氓 RA 消息。RA 被路由器用来在链路上通告自己的存在。RA 防护特性会分析 RA 消息并过滤掉未授权路由器发送的伪造 RA。在主机模式中，所有的路由器通告以及路由器重定向消息都不在端口上被允许。RA 防护特性会在二层设备上对比配置信息与在接收的 RA 帧中发现的信息。一旦二层设备对比配置验证了 RA 帧以及路由器重定向帧的内容，它会把 RA 转发给其单播或组播目的地址。如果 RA 帧的内容未被验证，RA 会被丢弃。
- IPv6 DHCP 防护——IPv6 DHCP 防护特性会阻塞来自未授权 DHCPv6 服务器及中继代理的

应答及通告消息。IPv6 DHCP 防护特性可以防止伪造的消息被输入到绑定表，并且可以阻塞在没有显式配置为面向 DHCPv6 服务器或 DHCP 中继的端口上接收的 DHCPv6 服务器消息。要使用此特性，需配置策略并将其配置到接口或 VLAN 上。要显示 DHCP 防护数据包的调试信息，使用特权 EXEC 命令 **debug ipv6 snooping dhcp-guard**。

- IPv6 源防护——与 IPv4 源防护相似，IPv6 源防护会验证源地址或前缀信息，以避免源地址伪造。

  源防护程序会基于源或目的地址信息，控制硬件允许或拒绝流量。该特性只处理数据包流量。

  IPv6 源防护特性允许把条目存储在硬件 TCAM 表中，以防止主机发送带有非法 IPv6 源地址的数据包。

  要显示源防护数据包的调试信息，使用特权 EXEC 命令 **debug ipv6 snooping source-guard**。

  注释： IPv6 源防护及前缀防护特性仅在入方向支持，不支持在出方向执行。

  该特性有以下限制：

  - 物理端口是 EtherChannel 组成员时，不能为其配置 FHS 策略。
  - 在交换机端口上启用 IPv6 源防护时，必须在交换机端口所属的接口上启用 NDP 或 DHCP 侦听。否则，来自该端口的所有数据流量都会被阻塞。
  - IPv6 源防护策略不能配置到 VLAN 上，进在接口级别支持。
  - 不能同时使用 IPv6 源防护以及前缀防护特性。把策略配置到接口上时，策略应该“验证地址”或“验证前缀”，而不能同时验证。
  - PVLAN 以及源/前缀防护不能同时应用。
  - IPv6 源防护及前缀防护支持在 EtherChannel 上使用。

  有关 IPv6 源防护的更多信息，参见 icntnetworks.com 网站 Inspur INOS IPv6 配置指南库的“IPv6 源防护”一章。

- IPv6 前缀防护——IPv6 前缀防护特性在 IPv6 源防护特性之下工作，让设备拒绝源自非拓扑正确地址的流量。IPv6 前缀防护通常在使用 DHCP 前缀授权功能给设备授权 IPv6 前缀时使用（如家庭网关）。该特性能够发现分配给链路的地址范围，并阻塞源地址在范围外的流量。

  有关 IPv6 前缀防护的更多信息，参见 icntnetworks.com 网站 Inspur INOS IPv6 配置指南库的“IPv6 前缀防护”一章。

- IPv6 目的防护——IPv6 目的防护特性在 IPv6 邻居发现之下工作，确保设备只对链路上已知的活跃地址进行地址解析。其依赖地址收集功能把链路上活跃的目的填充到绑定表中，且在解析绑定表中未发现的地址发生之前进行阻塞。

  注释： 建议把 IPv6 目的防护特性应用在配置了 SVI 的二层 VLAN 中。

有关 IPv6 目的防护的更多信息，参见 icntnetworks.com 网站 Inspur INOS IPv6 配置指南库的“IPv6 目的防护”一章。

关于基于 SISF 的 IPv4 及 IPv6 设备追踪的信息

基于交换机集成安全特性（Switch Integrated Security Features based，SISF-based）的 IP 设备追踪作为容器策略，支持在 IPv4 和 IPv6 中通过 IP 诊断 CLI 命令使用 FHS 提供的侦听及设备追踪特性。

所有现有的 IPv6 侦听命令都有对应的基于 SISF 的设备追踪命令，可以把配置同时应用在 IPv4 和 IPv6 地址族上。

对于设备上存在的传统 IP 设备追踪以及 IPv6 侦听配置，新 **device-tracking upgrade-cli** 的允许管理员把现有配置迁移成新的基于 SISF 的设备追踪 CLI 命令。更多信息参见*迁移 IPDT 以及 IPv6 侦听命令到基于 SISF 的设备追踪命令*。

迁移到基于 SISF 的设备追踪 CLI 时的限制

- 如果设备上没有传统的 IP 设备追踪（IPDT）或 IPv6 侦听 CLI 配置，对于未来的配置可以仅使用新的基于 SISF 的设备追踪 CLI 命令。老的 IP 设备追踪 CLI 以及 IPv6 侦听 CLI 不可用。
- 如果设备上配置了 IPv6 侦听，对于未来配置可以继续使用传统的 IPv6 侦听 CLI，也可以使用 **device-tracking upgrade-cli** 命令将其迁移到新的基于 SISF 的设备追踪 CLI。在所有传统的 IPv6 侦听命令都被转换之后，设备上只能运行新的设备追踪命令。如果不使用 **device-tracking upgrade-cli** 命令，设备上只可使用传统的 IPv6 侦听命令。
- 如果在设备上配置了 IPDT，可以继续使用传统的 IPDT 命令以及 IPv6 侦听命令。此选项限制用户使用传统模式，设备上只可以使用传统的 IPDT 以及 IPv6 侦听命令。然而，建议管理员将传统配置迁移到新的基于 SISF 的设备追踪命令。
- 要把传统的 IPDT 以及 IPv6 侦听配置迁移到新的基于 SISF 的设备追踪命令，需运行 **device-tracking upgrade-cl** 命令。在运行此命令后，设备上只可以使用新的设备追踪命令，且传统的 IPDT 或 IPv6 侦听命令都不被支持。

- 不能混用旧的 IPDT 和 IPv6 侦听 CLI 以及新的基于 SISF 的设备追踪 CLI。
- 如果在传统模式中启用了 **ip dhcp snooping vlan** 命令，当传统配置迁移到新的基于 SISF 的设备追踪配置时，一个称为 WL-DEV-TRACK-DHCP 的设备追踪策略会被自动创建，用来追踪启用了 IP 设备追踪的 IPv4 以及 IPv6 客户端。如果未启用 **ip dhcp snooping vlan**，确保在设备上启用设备追踪特性，以支持其他依赖于设备追踪的特性。

迁移 IPDT 以及 IPv6 侦听命令到基于 SISF 的设备追踪命令

建议使用 **device-tracking upgrade-cli** 命令，迁移传统的 IP 设备追踪（IPDT）以及 IPv6 侦听命令到新的设备追踪命令 CLI 命令。

配置情景及迁移结果

基于设备上现有的传统配置，**device-tracking upgrade-cli** 命令会使用不同方式升级 CLI。在迁移现有配置时，请考虑以下情景及对应的迁移信息。

只存在 IPDT 配置

如果设备只有 IP 设备追踪（IPDT）配置，运行 **device-tracking upgrade-cli** 命令会在设备内部把配置翻译成新的 SISF 策略并配置在接口上。可以之后更新此 SISF 策略。

只存在 IPv6 侦听配置

在有 IPv6 侦听配置的设备上，老的 IPv6 侦听命令可以用作以后的配置。存在以下选项：

- （推荐）使用 **device-tracking upgrade-cli** 命令把传统配置迁移到新的基于 SISF 的设备追踪命令。在所有传统命令都被转换后，在设备上只能使用新的设备追踪命令。
- 在未来配置中使用传统的 IPv6 侦听命令，不运行 **device-tracking upgrade-cli** 命令。在此选项中，设备上只可以使用传统的 IPv6 侦听命令，且不能使用新的基于 SISF 的设备追踪 CLI 命令。

一个名为 Default 的设备追踪策略会在转换过程中被创建。无法手动把此策略配置到其他接口上。

同时存在 IPDT 以及 IPv6 侦听配置

在同时存在传统 IPDT 配置以及 IPv6 侦听配置的设备上，可以使用 **device-tracking upgrade-cli** 命令把传统命令转换为新的设备追踪 CLI 命令。然而，要注意只能给接口配置一个侦听策略，且 IPv6 侦听策略的参数会覆盖 IPDT 的设置。

注释： 如果不迁移到新的基于 SISF 的命令，且继续使用传统的 IPv6 侦听或 IPDT 命令，设备上的 IPv4 设备追踪配置信息可能会在 IPv6 侦听命令的输出中显示，因为该命令作为统一的特性，会同时处理 IPv4 和 IPv6 的配置。为了避免这样的情况，建议迁移传统配置并使用新的设备追踪命令。

不存在 IPDT 或 IPv6 侦听配置

如果设备上没有传统的 IP 设备追踪或 IPv6 侦听配置，管理员在以后的配置中只能使用基于 SISF 的 **device-tracking** 命令。传统的 IPDT 命令以及 IPv6 侦听命令不可用。

IPDT、IPv6 侦听以及设备追踪 CLI 兼容性

下表显示了新的基于 SISF 的设备追踪命令以及对应的 IPDT 和 IPv6 侦听命令。

表 x

IP 设备追踪（IPDT）	**IPv6 侦听**	**基于 SISF 的设备追踪**
ip device tracking probe count	不支持	不支持
ip device tracking probe delay	**ipv6 neighbor bindingreachable-lifetime**	**device-tracking policyreachable-lifetime**
ip device tracking probe interval	**ipv6 snooping trackingretry-interval**	**device-tracking policyretry-interval**
ip device tracking probe use-svi	接受，解释为**ipdevice tracking probeauto-source override**	接受，解释为 **ipdevice tracking probeauto-source override**
ip device tracking probeauto-source fallback	不支持	不支持
ip device tracking probeauto-source override	不支持	不支持
ip device tracking tracebuffer	不支持	不支持
ip device tracking maximum	**ipv6 snooping policy** <*name*>**limit**	**device-tracking snooping policy**<*name*>**limit**
ip device tracking probe count	不支持	不支持
ip device tracking probe interval	不支持	不支持
clear ip device tracking all	不支持	不支持

如何创建基于 SISF 的设备追踪及侦听策略

在特权 EXEC 模式中，按照以下步骤配置设备追踪策略。

总步骤

1. configure terminal

2. device-tracking policy *policy-name*

3. {[**device-role** {**node** | **switch**}] | [**limit address-count** *value*] | [**no**] | [**destination-glean**{**recovery**|**log-only**[**dhcp**]] | [**data-glean**{**recovery**|**log-only**{**dhcp** | **ndp**}] | **prefix-glean**] | [**security-level** {**glean** | **guard** | **inspect**}] | [**tracking** {**disable** [**stale-lifetime** [*seconds* | **infinite**] | **enable** [**reachable-lifetime** [*seconds* | **infinite**] }] | [**trusted-port**] }

4. end

5. show device-tracking policy *policy-name*

具体步骤

	命令或操作	目的
步 骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式。
步 骤 2	**device-tracking policy** *policy-name* 示例: `Device(config)#` **device-tracking policy example_policy**	进入设备追踪配置模式。
步 骤 3	{[**device-role** {**node**	**switch**}]	[**limit address-count** *value*]	[**no**]	[**destination-glean**{**recovery**|**log-only**[**dhcp**]]	[**data-glean**{**recovery**|**log-only**{**dhcp**	**ndp**}]	**prefix-glean**]	[**security-level** {**glean**	**guard**	**inspect**}]	[**tracking** {**disable** [**stale-lifetime**	为 IPv4 和 IPv6 启用以下选项： • （可选）**device-role**{**node**	**switch**}——指定连接到端口的设备角色。默认是 **node**。 • （可选）**limit address-count** *value*——限制每个目标允许的地址数量。 • （可选）**no**——取消命令或将其设置为默认配置。 • （可选）

	[*seconds* | **infinite**] | **enable** [**reachable-lifetime** [*seconds* | **infinite**] }] | [**trusted-port**] } **示例:** Device(config-device-tracking)# security-level inspect **示例:** Device(config-device-tracking)# trusted-port	**destination-glean**{**recovery**|**log-only**}[**dhcp**]}——通过数据流量源地址收集来恢复绑定表。 • （可选）**data-glean**{**recovery**|**log-only**}[**dhcp** | **ndp**]}——使用源或数据地址收集来恢复绑定表。 • （可选）**security-level**{**glean**|**guard**|**inspect**}——指定特性执行的安全等级。默认是 **guard**。 **glean**——从消息中收集地址，且无需认证就可以填充到绑定表。 **guard**——收集并监测消息。此外，拒绝路由器通告（RA）以及 DHCP 服务器消息。这是默认的选项。 **inspect**——收集地址，验证消息的一致性，并进行地址从属检查。 • （可选）**tracking** {**disable** | **enable**}——指定追踪选项。 • （可选）**trusted-port**——设置可信端口。禁用对适用目标的防护。通过可信端口学习到的绑定优先于通过其他端口学习到的绑定。在创建表条目发生冲突时，可信端口的条目有高优先级。
步骤 4	**end** **示例:** `Device(config-device-tracking)# exit`	推出配置模式。

步骤 5	**show device-tracking policy** *policy-name* **示例:** `Device#`**show device-tracking policy example_policy**	显示设备追踪策略配置。

如何将设备追踪策略配置到接口

在特权 EXEC 模式中，按照以下步骤将设备追踪策略配置到接口。

总步骤

1. configure terminal

2. interface *interface*

3. device-tracking attach-policy *policy name*

4. show device-tracking policies [**interface***interface*]

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface***interface* **示例:** `Device(config)#` **interface gigabitethernet 1/1/4**	指定接口并进入接口配置模式。
步骤 3	**device-tracking attach-policy** *policy name* **示例:** `Device(config-if)#` **device-tracking attach-policy example_policy**	将设备追踪策略配置到接口或接口的指定 VLAN 上。
步骤 4	**show device-tracking policies** [**interface** *interface*] **示例:** `Device#(config-if)#` **do show running-config**	显示匹配特定接口类型及编号的策略。

如何将设备追踪策略配置到 VLAN

在特权 EXEC 模式中，按照以下步骤将设备追踪策略配置到 VLAN 上。

总步骤

1. configure terminal

2. vlan configuration *vlan_list*

3. device-tracking [**attach-policy** *policy_name*]

4. do show running-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 2	**vlan configuration** *vlan_list* **示例：** `Device(config)# vlan configuration 333`	指定要配置设备追踪策略的 VLAN，进入 VLAN 接口配置模式。
步骤 3	**device-tracking** [**attach-policy** *policy_name*] **示例：** `Device(config-vlan-config)#device-tracking attach-policy example_policy`	将设备追踪策略配置到指定的 VLAN 上。
步骤 4	**do show running-config** **示例：** `Device#(config-if)# do show running-config`	在接口配置模式中验证策略应用到了指定的 VLAN 上。

如何向绑定表中添加设备范围的条目

在特权 EXEC 模式中，按照以下步骤配置绑定表内容。

总步骤

1. configure terminal

2. [**no**] **device-trackingDefault**| [**down-lifetime** *value*] | [**logging**] | [**max entries***value*] |

[**reachable-lifetime**

seconds | **retry-interval** *seconds*] | [**stale-lifetime**[*seconds*]

3. exit

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	[**no**] **device-tracking Default**| [**down-lifetime***value*] | [**logging**] | [**max entries***value*] |[**reachable-lifetime** *seconds* | **retry-interval** *seconds*] | [**stale-lifetime**[*seconds*] **示例：** `Device(config)#` **device-tracking Default**	使用以下选项创建设备范围的默认设备追踪策略，并将绑定表中添加条目。 • **down-lifetime**——设置条目在被删除之前保持 DOWN 状态的默认最大时间。 • **logging**——对绑定表事件启动系统日志记录。 • **max-entries**——定义绑定表的最大条目数量。 • **reachable-lifetime**——定义在无需证明可达性的情况下，一个可达的条目被认为是直接或间接可达的最长时间。 • **retry-interval**——定义两次探测的间隔。 • **stale-lifetime**——定义条目在被删除之前保持 Stale 状态的最大时间。
步骤 3	**exit** **示例：** `Device(config)#` **exit**	退出全局配置模式，返回特权 EXEC 配置模式。

如何配置 IPv6 侦听策略

在特权 EXEC 模式中，按照以下步骤配置 IPv6 侦听策略。

总步骤

1. configure terminal

2. ipv6 snooping policy *policy-name*

3. {[**default**] | [**device-role** {**node** | **switch**}] | [**limit address-count** *value*] | [**no**] | [**protocol** {**dhcp** | **ndp**}] | [**security-level** {**glean** | **guard** | **inspect**}] | [**tracking** {**disable** [**stale-lifetime** [*seconds* | **infinite**] |**enable** [**reachable-lifetime** [*seconds* | **infinite**] }] | [**trusted-port**] }

4. end

5. show ipv6 snooping policy *policy-name*

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**ipv6 snooping policy** *policy-name* **示例：** `Device(config)#` **ipv6 snooping policyexample_policy**	创建侦听策略并进入 IPv6 侦听策略配置模式。
步骤 3	{[**default**]	[**device-role** {**node**	**switch**}]	[**limit address-count** *value*]	[**no**]	[**protocol**{**dhcp**	**ndp**}]	[**security-level** {**glean**	**guard**| **inspect**}]	[**tracking** {**disable** [**stale-lifetime**[*seconds*	**infinite**]	**enable**[**reachable-lifetime** [*seconds*	**infinite**] }] |[**trusted-port**] } **示例：** Device(config-ipv6-snooping)# security-levelinspect **示例：** Device(config-ipv6-snooping)# trusted-port	启用数据地址收集，对比多项条件验证消息，指定消息的安全等级。 • （可选）**device-role**{**node**	**switch**}——指定连接到端口的设备角色。默认是 **node**。 • （可选）**limit address-count** *value*——限制每个目标允许的地址数量。 • （可选）**no**——取消命令或将其设置为默认配置。 • （可选）**protocol**{**dhcp**	**ndp**}——指定哪种协议应被重定向到侦听特性进行分析。默认设置是 **dhcp** 和 **ndp**。要更改默认设置，使用命令 **no protocol**。 • （可选）

		security-level{**glean**|**guard**|**inspect**}——指定特性执行的安全等级。默认等级是 **guard**。 **glean**——从消息中收集地址，且无需认证就可以填充到绑定表。 **guard**——收集并监测消息。此外，拒绝路由器通告（RA）以及 DHCP 服务器消息。这是默认的选项。 **inspect**——收集地址，验证消息的一致性，并进行地址从属检查。 • （可选）**tracking** {**disable** | **enable**}——覆盖默认追踪行为并指定追踪选项。 • （可选）**trusted-port**——设置可信端口。禁用对适用目标的防护。通过可信端口学习到的绑定优先于通过其他端口学习到的绑定。在创建表条目发生冲突时，可信端口的条目有高优先级。
步骤 4	**end** **示例：** `Device(config-ipv6-snooping)#` **exit**	返回特权 EXEC 模式。
步骤 5	**show ipv6 snooping policy** *policy-name* **示例：** `Device#`**show ipv6 snooping policy example_policy**	显示侦听策略配置。

接下来做什么？

配置 IPv6 侦听策略到接口或 VLAN。

如何把 IPv6 侦听策略配置到接口上

在特权 EXEC 模式中，按照以下步骤配置 IPv6 侦听策略到接口或接口上的 VLAN。

总步骤

1. configure terminal

2. interface Interface_type *stack/module/port*

3. switchport

4. ipv6 snooping [**attach-policy** *policy_name* [**vlan** {*vlan_id* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids*}] | **vlan** {*vlan_id* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]

5. do show running-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** Interface_type *stack/module/port* **示例：** `Device(config)#` **interface gigabitethernet 1/1/4**	指定接口类型及标识符，进入接口配置模式。
步骤 3	**switchport** **示例：** `Device(config-if)#` **switchport**	进入 switchport 模式。 **注释：**配置二层参数时，如果接口在三层模式，必须输入不带参数的 switchport 接口配置命令，将接口置为二层模式。此操作会关闭并重启接口，可能在设备上生成接口已连接的消息。把三层模式的接口置为二层模式时，接口之前的配置信息可能丢失，且接口会返回默认配置。交换机端口配置模式的命令提示符为(config-if)#。
步骤 4	**ipv6 snooping** [**attach-policy** *policy_name* [**vlan**{*vlan_id* | **add** *vlan_ids* | **except***vlan_ids* | **none** |**remove** *vlan_ids*}]	将定义的IPv6侦听策略配置到接口或接口上的VLAN。要给接口配置默认策略，使用不带**attach-policy**关键字的**ipv6**

	| **vlan** {*vlan_id* | **add** *vlan_ids*| **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] **示例：** `Device(config-if)#` **ipv6 snooping** 或 `Device(config-if)#` **ipv6 snooping attach-policy example_policy** 或 `Device(config-if)#` **ipv6 snooping vlan 111,112** 或 `Device(config-if)#` **ipv6 snooping attach-policy example_policy vlan 111,112**	**snooping**命令。要给接口上的VLAN配置默认策略，使用**ipv6snooping vlan**命令。默认策略中，安全等级是**guard**，设备角色是**node**，协议是**ndp**和**dhcp**。
步骤 5	**do show running-config** **示例：** `Device#(config-if)#` **do show running-config**	在接口配置模式中验证策略应用到了指定的接口上。

如何把 IPv6 侦听策略配置到二层 EtherChannel 接口上

在特权 EXEC 模式中，按照以下步骤配置 IPv6 侦听策略到 EtherChannel 接口或 VLAN。

总步骤

1. configure terminal

2. interface range *Interface_name*

3. ipv6 snooping [**attach-policy** *policy_name* [**vlan** {*vlan_ids* \| **add** *vlan_ids* \| **except** *vlan_ids* \| **none** \|**remove** *vlan_ids* \| **all**}] \| **vlan** [{*vlan_ids* \| **add** *vlan_ids* \| **except***vlan_ids* \| **none** \| **remove** *vlan_ids* \|**all**}]

4. do show running-config interface *portchannel_interface_name*

具体步骤

	命令或操作	目的
步骤 1	**configure terminal**	进入全局配置模式。

	示例： `Device#` **configure terminal**	
步骤 2	**interface range** *Interface_name* **示例：** `Device(config)#` **interface Po11**	指定创建 EtherChannel 时分配的端口通道接口名称。进入接口范围配置模式。 提示：输入 **do show interfaces summary** 命令快速查询接口名称以及类型。
步骤 3	**ipv6 snooping** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* |**add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** |**remove** *vlan_ids* | **all**}] **示例：** `Device(config-if-range)#` **ipv6 snooping attach-policyexample_policy** 或 `Device(config-if-range)#` **ipv6 snooping attach-policyexample_policy vlan 222,223,224** **或** `Device(config-if-range)#`**ipv6 snooping vlan 222,223,224**	将 IPv6 侦听策略配置到接口或接口上的 VLAN。如果不使用 **attach-policy** 选项，则配置默认策略。
步骤 4	**do show running-config interface** *portchannel_interface_name* **示例：** `Device#(config-if-range)#` **do show running-configint po11**	在当前模式中确认策略已配置在指定的接口上。

如何把 IPv6 侦听策略全局配置到 VLAN 上

在特权 EXEC 模式中，按照以下步骤把 IPv6 侦听策略配置到 VLAN 上。

总步骤

1. configure terminal

2. vlan configuration *vlan_list*

3. ipv6 snooping [**attach-policy** *policy_name*]

4. do show running-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**vlan configuration** *vlan_list* **示例：** `Device(config)#` **vlan configuration 333**	指定要配置 IPv6 侦听策略的 VLAN，进入 VLAN 接口配置模式。
步骤 3	**ipv6 snooping** [**attach-policy** *policy_name*] **示例：** `Device(config-vlan-config)#`**ipv6 snooping** **attach-policy example_policy**	将 IPv6 侦听策略配置到 VLAN 上，VLAN 可以覆盖所有交换机以及堆栈接口。如果不使用 **attach-policy** 选项，默认策略会被配置。默认策略中，安全等级是 **guard**，设备角色是 **node**，协议是 **ndp** 和 **dhcp**。
步骤 4	**do show running-config** **示例：** `Device#(config-if)#` **do show running-config**	在当前模式中确认策略已配置在指定的 VLAN 上。

如何配置 IPv6 绑定表内容

在特权 EXEC 模式中，按照以下步骤配置 IPv6 绑定表内容。

总步骤

1. configure terminal

2. [**no**] **ipv6 neighbor binding** [**vlan** *vlan-id* {*ipv6-address* **interface** interface_type *stack/module/porthw_address* [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**] | [**tracking**{ [default | disable] [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**] | [**enable** [**reachable-lifetimevalue** [*seconds* |**default** | **infinite**] | [**retry-interval** {*seconds*| **default** [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**]}]

3. [**no**] **ipv6 neighbor binding max-entries** *number* [**mac-limit** *number* | **port-limit** *number*

[**mac-limit***number*] | **vlan-limit** *number* [[**mac-limit** *number*] | [**port-limit** *number*

[**mac-limit***number*]]]]

4. ipv6 neighbor binding logging

5. exit

6. show ipv6 neighbor binding

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	[**no**] **ipv6 neighbor binding** [**vlan** *vlan-id* {*ipv6-address* **interface**interface_type *stack/module/port hw_address* [**reachable-lifetimevalue**[*seconds*	**default** | **infinite**]	[**tracking**{ [default	disable] [**reachable-lifetimevalue** [*seconds* | **default**	**infinite**] | [**enable**[**reachable-lifetimevalue** [*seconds* | **default**	**infinite**] | [**retry-interval**{*seconds*| **default** [**reachable-lifetimevalue** [*seconds* | **default**	**infinite**]}] **示例：** `Device(config)#` **ipv6 neighbor binding**	向绑定表中添加静态条目。
步骤 3	[**no**] **ipv6 neighbor binding max-entries** *number* [**mac-limit** *number*	**port-limit** *number* [**mac-limit** *number*]	**vlan-limit** *number* [[**mac-limit***number*]	[**port-limit** *number* [**mac-limit***number*]]]] **示例：** `Device(config)#` **ipv6 neighbor binding** **max-entries 30000**	指定允许添加到绑定表缓存的最大条目数量。

步骤 4	**ipv6 neighbor binding logging** **示例：** `Device(config)#` **ipv6 neighbor binding logging**	记录绑定表主要事件。
步骤 5	**exit** **示例：** `Device(config)#` **exit**	退出全局配置模式，返回特权 EXEC 模式。
步骤 6	**show ipv6 neighbor binding** **示例：** `Device#` **show ipv6 neighbor binding**	显示绑定表内容。

如何配置 IPv6 邻居发现监测策略

在特权 EXEC 模式中，按照以下步骤配置 IPv6 ND 监测策略。

总步骤

1. configure terminal

2. [**no**]**ipv6 nd inspection policy** *policy-name*

3. device-role {**host** | **monitor** | **router** | **switch**}

4. drop-unsecure

5. limit address-count *value*

6. sec-level minimum *value*

7. tracking {**enable** [**reachable-lifetime** {*value* | **infinite**}] | **disable** [**stale-lifetime** {*value* | **infinite**}]}

8. trusted-port

9. validate source-mac

10. no {**device-role** | **drop-unsecure** | **limit address-count** | **sec-level minimum** | **tracking** | **trusted-port** |**validate source-mac**}

11. default {**device-role** | **drop-unsecure** | **limit address-count** | **sec-level minimum** | **tracking** | **trusted-port**| **validate source-mac**}

12. do show ipv6 nd inspection policy *policy_name*

具体步骤

	命令或操作	目的

步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	[**no**]**ipv6 nd inspection policy** *policy-name* **示例：** `Device(config)#` **ipv6 nd inspection policyexample_policy**	指定 ND 监测策略名称，进入 ND 监测策略配置模式。
步骤 3	**device-role** {**host** | **monitor** | **router** | **switch**} **示例：** `Device(config-nd-inspection)#` **device-role switch**	指定连接到端口的设备角色。默认角色是 **host**。
步骤 4	**drop-unsecure** **示例：** `Device(config-nd-inspection)#` **drop-unsecure**	丢弃无选项、选项不合法或签名不合法的消息。
步骤 5	**limit address-count** *value* **示例：** `Device(config-nd-inspection)#` **limit address-count 1000**	输入 1 到 10000 的值。
步骤 6	**sec-level minimum** *value* **示例：** `Device(config-nd-inspection)#` **limit address-count 1000**	指定使用加密生成的地址（Cryptographically Generated Address，CGA）选项时的最小安全等级参数。
步骤 7	**tracking** {**enable** [**reachable-lifetime** {*value* | **infinite**}] |**disable** [**stale-lifetime** {*value* | **infinite**}]} **示例：** `Device(config-nd-inspection)#` **tracking disablestale-lifetime infinite**	覆盖端口上的默认追踪策略。
步骤 8	**trusted-port** **示例：**	配置端口为可信端口。

	Device(config-nd-inspection)# **trusted-port**	
步骤 9	**validate source-mac** **示例：** Device(config-nd-inspection)# **validate source-mac**	对比源介质访问控制（MAC）地址与链路层地址。
步骤 10	**no** {**device-role** | **drop-unsecure** | **limit address-count** |**sec-level minimum** | **tracking** | **trusted-port** | **validatesource-mac**} **示例：** Device(config-nd-inspection)# **no validate source-mac**	使用命令的 **no** 形式移除当前配置的参数。
步骤 11	**default** {**device-role** | **drop-unsecure** | **limit address-count** |**sec-level minimum** | **tracking** | **trusted-port** | **validatesource-mac**} **示例：** Device(config-nd-inspection)# **default limit address-count**	恢复配置为默认设置。
步骤 12	**do show ipv6 nd inspection policy** *policy_name* **示例：** Device(config-nd-inspection)# **do show ipv6 ndinspection policy example_policy**	在当前配置模式中验证配置的 ND 监测配置。

如何把 IPv6 邻居发现监测策略配置到接口上

在特权 EXEC 模式中，按照以下步骤配置 IPv6 ND 监测策略到接口或接口上的 VLAN。

总步骤

1. configure terminal

2. interface Interface_type *stack/module/port*

3. ipv6 nd inspection [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none**| **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* |**all**}]

4. do show running-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** Interface_type *stack/module/port* **示例：** `Device(config)#` **interface gigabitethernet 1/1/4**	指定接口类型及标识符，进入接口配置模式。
步骤 3	**ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan** {*vlan_ids*| **add** *vlan_ids*	**except** *vlan_ids*	**none**	**remove** *vlan_ids*	**all**}]	**vlan** [{*vlan_ids*	**add** *vlan_ids*	**except***vlan_ids*	**none** |**remove** *vlan_ids*	**all**}] **示例：** `Device(config-if)#` **ipv6 nd inspection attach-policyexample_policy** 或 `Device(config-if)#` **ipv6 nd inspection attach-policyexample_policy vlan 222,223,224** **或** `Device(config-if)#` **ipv6 nd inspection vlan 222, 223,224**	配置邻居发现监测策略到接口或接口上的指定 VLAN。如果不使用 **attach-policy** 选项，默认策略会被配置。
步骤 4	**do show running-config** **示例：** `Device#(config-if)#` **do show running-config**	在接口配置模式中验证配置到接口的策略。

如何把 IPv6 邻居发现监测策略配置到二层 EtherChannel 接口上

在特权 EXEC 模式中，按照以下步骤配置 IPv6 邻居发现监测策略到 EtherChannel 接口或 VLAN。

总步骤

1. configure terminal

2. interface range *Interface_name*

3. ipv6 nd inspection [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none**| **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* |**all**}]

4. do show running-config interface *portchannel_interface_name*

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface range** *Interface_name* **示例：** `Device(config)#` **interface Po11**	指定创建 EtherChannel 时分配的端口通道接口名称。进入接口范围配置模式。 **提示：** 输入 **do show interfaces summary** 命令快速查看接口名称及类型。
步骤 3	**ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan**{*vlan_ids*	**add** *vlan_ids*	**except** *vlan_ids*	**none**	**remove***vlan_ids*	**all**}]	**vlan** [{*vlan_ids*	**add** *vlan_ids* |**except***vlan_ids*	**none**	**remove** *vlan_ids*	**all**}] **示例：** `Device(config-if-range)#` **ipv6 nd inspectionattach-policy example_policy**	配置 ND 监测策略到接口或接口上的指定 VLAN。如果不使用 **attach-policy** 选项，默认策略会被配置。

	或 Device(config-if-range)# **ipv6 nd inspectionattach-policy example_policy vlan 222,223,224** 或 Device(config-if-range)#**ipv6 nd inspection vlan 222,223,224**	
步骤 4	**do show running-config interface***portchannel_interface_name* **示例：** Device#(config-if-range)# **do show running-config int po11**	在当前配置模式中确认指定接口的策略配置。

如何把 IPv6 邻居发现监测策略全局配置到 VLAN 上

在特权 EXEC 模式中，按照以下步骤把 IPv6 ND 监测策略配置到覆盖多个接口的 VLAN 上。

总步骤

1. configure terminal

2. vlan configuration *vlan_list*

3. ipv6 nd inspection [**attach-policy** *policy_name*]

4. do show running-config

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式。
步骤 2	**vlan configuration** *vlan_list* **示例：** Device(config)# **vlan configuration 334**	指定要配置 IPv6 侦听策略的 VLAN，进入 VLAN 接口配置模式。
步骤 3	**ipv6 nd inspection** [**attach-policy** *policy_name*]	把 IPv6 邻居发现策略配置到覆盖所有交换机及堆栈接口的指定 VLAN 上。如

	示例： `Device(config-vlan-config)#`**ipv6 ndinspection attach-policy example_policy**	果不使用 **attach-policy** 选项，默认策略会被使用。 默认策略中，主机角色是 **host**，无 drop-unsecure 设置，禁用地址计数限制，禁用最小安全等级设置，禁用追踪，无可信端口，不验证源 MAC 地址。
步骤 4	**do show running-config** **示例：** `Device#(config-if)#` **do show running-config**	在当前配置模式中确认指定 VLAN 的策略配置。

如何配置 IPv6 路由器通告防护策略

在特权 EXEC 模式中，按照以下步骤配置 IPv6 路由器通告防护策略。

总步骤

1. configure terminal

2. [**no**]**ipv6 nd raguard policy** *policy-name*

3. [**no**]**device-role** {**host** | **monitor** | **router** | **switch**}

4. [**no**]**hop-limit** {**maximum** | **minimum**} *value*

5. [**no**]**managed-config-flag** {**off** | **on**}

6. [**no**]**match** {**ipv6 access-list** *list* | **ra prefix-list** *list*}

7. [**no**]**other-config-flag** {**on** | **off**}

8. [**no**]**router-preference maximum** {**high** | **medium** | **low**}

9. [**no**]**trusted-port**

10. default {**device-role** | **hop-limit** {**maximum** | **minimum**} | **managed-config-flag** | **match** {**ipv6 access-list**| **ra prefix-list** } | **other-config-flag** | **router-preference maximum**| **trusted-port**}

11. do show ipv6 nd raguard policy *policy_name*

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：**	进入全局配置模式。

	Device# **configure terminal**	
步骤 2	[**no**]**ipv6 nd raguard policy** *policy-name* **示例：** Device(config)# **ipv6 nd raguard policy example_policy**	指定 RA 防护策略名称并进入 RA 防护策略配置模式。
步骤 3	[**no**]**device-role** {**host** | **monitor** | **router** |**switch**} **示例：** Device(config-nd-raguard)# **device-roleswitch**	指定连接到端口的设备角色。默认角色是 **host**。
步骤 4	[**no**]**hop-limit** {**maximum** | **minimum**} *value* **示例：** Device(config-nd-raguard)# **hop-limit maximum 33**	最大和最小跳数限制的范围（1-255）。使用跳数限制值过滤路由器通告消息。流氓 RA 消息可能有较低的跳数限制值（等同于 IPv4 的生存时间），该消息被主机接受时，能阻止主机生成发往流氓 RA 产生者以外的流量。跳数限制未指定的 RA 消息会被阻塞。 如果不进行配置，此项过滤会被禁用。 配置 **minimum** 来阻塞跳数限制值低于配置值的 RA 消息。配置 **maximum** 来阻塞跳数限制值高于配置值的 RA 消息。
步骤 5	[**no**]**managed-config-flag** {**off** | **on**} **示例：** Device(config-nd-raguard)# **managed-config-flag on**	使用管理地址配置标识（ManagedAddress Configuration，也称“M”标识字段）过滤路由器通告消息。M 字段为 1 的流氓 RA 消息可能导致主机使用流氓 DHCPv6 服务器。如果不进行配置，此项过滤被禁用。 **On**——接受并转发 M 值为 1 的 RA 消息，并阻塞 M 值为 0 的消息。 **Off**——接受并转发 M 值为 0 的 RA 消息，并阻塞 M 值为 1 的消息。

步骤 6	[**no**]**match** {**ipv6 access-list** *list* | **ra prefix-list***list*} **示例：** `Device(config-nd-raguard)#` **match ipv6access-list example_list**	匹配指定的前缀列表或访问列表。
步骤 7	[**no**]**other-config-flag** {**on** | **off**} **示例：** `Device(config-nd-raguard)#`**other-config-flag on**	使用其他配置（OtherConfiguration，也称“O”标识字段）过滤路由器通告消息。O 字段为 1 的流氓 RA 消息可能导致主机使用流氓 DHCPv6 服务器。如果不进行配置，此项过滤被禁用。 **On**——接受并转发 O 值为 1 的 RA 消息，并阻塞 O 值为 0 的消息。 **Off**——接受并转发 O 值为 0 的 RA 消息，并阻塞 O 值为 1 的消息。
步骤 8	[**no**]**router-preference maximum** {**high** |**medium** | **low**} **示例：** `Device(config-nd-raguard)#`**router-preference maximum high**	使用路由器优先级标志过滤路由器通告消息。如果不进行配置，此项过滤被禁用。 **high**——接受路由器优先级设置为高、中或低的 RA 消息。 **medium**——阻塞路由器优先级设置为高的 RA 消息。 **low**——阻塞路由器优先级设置为中和高的 RA 消息。
步骤 9	[**no**]**trusted-port** **Example:** `Device(config-nd-raguard)#` **trusted-port**	端口配置为可信时，所有连接的设备都被信任，且不再对其进行消息验证。
步 骤 10	**default** {**device-role** | **hop-limit** {**maximum** |**minimum**} | **managed-config-flag** | **match** {**ipv6access-list** | **ra prefix-list** } | **other-config-flag** |**router-preference maximum**| **trusted-port**} **示例：**	恢复命令为默认值。

	`Device(config-nd-raguard)#` **defaulthop-limit**	
步 骤 11	**do show ipv6 nd raguard policy** *policy_name* **示例：** `Device(config-nd-raguard)#` **do show ipv6nd raguard policy example_policy**	（可选）在 RA 防护策略配置模式中显示 ND 防护策略配置。

如何把 IPv6 路由器通告防护策略配置到接口上

在特权 EXEC 模式中，按照以下步骤配置 IPv6 路由器防护策略到接口或接口上的 VLAN。

总步骤

1. configure terminal

2. interface Interface_type *stack/module/port*

3. ipv6 nd raguard [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** |**remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* |**all**}]

4. do show running-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** Interface_type *stack/module/port* **示例：** `Device(config)#` **interface gigabitethernet 1/1/4**	指定接口类型及标识符，进入接口配置模式。
步骤 3	**ipv6 nd raguard** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* |**add** *vlan_ids*	**except** *vlan_ids*	**none**	**remove** *vlan_ids*	**all**}]| **vlan** [{*vlan_ids*	**add** *vlan_ids*	**except***vlan_ids*	**none**	**remove***vlan_ids*	**all**}] **示例：**	把 IPv6 邻居发现监测策略配置到接口或接口的指定 VLAN 上。如果不使用 **attach-policy** 选项，默认策略会被配置。

	Device(config-if)# **ipv6 nd raguard attach-policyexample_policy** 或 Device(config-if)# **ipv6 nd raguard attach-policyexample_policy vlan 222,223,224** **或** Device(config-if)# **ipv6 nd raguard vlan 222, 223,224**	
步骤 4	**do show running-config** **示例：** Device#(config-if)# **do show running-config**	在当前配置模式中确认策略配置到指定接口上。

如何把 IPv6 路由器通告防护策略配置到二层 EtherChannel 接口上

在特权 EXEC 模式中，按照以下步骤配置 IPv6 路由器通告防护策略到 EtherChannel 接口或 VLAN 上。

总步骤

1. configure terminal

2. interface range *Interface_name*

3. ipv6 nd raguard [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** |**remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* |**all**}]

4. do show running-config interface *portchannel_interface_name*

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式。
步骤 2	**interface range** *Interface_name* **示例：**	指定创建 EtherChannel 时分配的端口通道接口名称。进入接口范围配置模

	Device(config)# **interface Po11**	式。 提示：输入 **do show interfaces summary** 命令快速查询接口名称以及类型。
步骤 3	**ipv6 nd raguard** [**attach-policy** *policy_name* [**vlan** {*vlan_ids*| **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* |**all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none**| **remove** *vlan_ids* | **all**}] **示例：** Device(config-if-range)# **ipv6 nd raguardattach-policy example_policy** 或 Device(config-if-range)# **ipv6 nd raguardattach-policy example_policy vlan 222,223,224** **或** Device(config-if-range)#**ipv6 nd raguard vlan 222,223,224**	把 IPv6 邻居发现监测策略配置到接口或接口的指定 VLAN 上。如果不使用 **attach-policy** 选项，默认策略会被配置。
步骤 4	**do show running-config interface** *portchannel_interface_name* **示例：** Device#(config-if-range)# **do show running-config** **int po11**	在当前配置模式中确认策略配置到指定接口上。

如何把 IPv6 路由器通告防护策略全局配置到 VLAN 上

在特权 EXEC 模式中，按照以下步骤把 IPv6 路由器通告防护策略配置到 VLAN 上。

总步骤

1. configure terminal

2. vlan configuration *vlan_list*

3. ipv6 dhcp guard [**attach-policy** *policy_name*]

4. do show running-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 2	**vlan configuration** *vlan_list* **示例：** `Device(config)# vlan configuration 335`	指定要配置 IPv6 RA 防护策略的 VLAN，进入 VLAN 接口配置模式。
步骤 3	**ipv6 dhcp guard** [**attach-policy** *policy_name*] **示例：** `Device(config-vlan-config)#ipv6 nd raguardattach-policy example_policy`	把 IPv6 邻居发现监测策略配置到指定的覆盖所有交换机和堆栈接口 VLAN 上。如果不使用 **attach-policy** 选项，默认策略会被配置。
步骤 4	**do show running-config** **示例：** `Device#(config-if)# do show running-config`	在当前配置模式中确认策略配置到指定 VLAN 上。

如何配置 IPv6 DHCP 防护策略

在特权 EXEC 模式中，按照以下步骤配置 IPv6 DHCP（DHCPv6）防护策略。

总步骤

1. configure terminal

2. [**no**]**ipv6 dhcp guard policy** *policy-name*

3. [**no**]**device-role** {**client** | **server**}

4. [**no**] **match server access-list** *ipv6-access-list-name*

5. [**no**] **match reply prefix-list** *ipv6-prefix-list-name*

6. [**no**]**preference**{ **max** *limit* | **min** *limit* }

7. [**no**] **trusted-port**

8. default {**device-role** | **trusted-port**}

9. do show ipv6 dhcp guard policy *policy_name*

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	[**no**]**ipv6 dhcp guard policy** *policy-name* **示例：** `Device(config)#` **ipv6 dhcp guard policy example_policy**	指定 DHCPv6 防护策略名称，并进入 DHCPv6 防护策略配置模式。
步骤 3	[**no**]**device-role** {**client** | **server**} **示例：** `Device(config-dhcp-guard)#` **device-role server**	（可选）过滤掉端口上来自指定角色设备以外的 DHCPv6 应答和 DHCPv6 通告。默认角色是 **client**。 • **client**——默认值，指定连接的设备是客户端。此端口上的服务器消息会被丢弃。 • **server**——指定连接的设备是一台 DHCPv6 服务器。此端口上的服务器消息被允许。
步骤 4	[**no**] **match server access-list** *ipv6-access-list-name* **示例：** `;;`假设预配置的`IPv6`访问列表如下： `Device(config)#` **ipv6 access-list my_acls** `Device(config-ipv6-acl)#` **permit hostFE80::A8BB:CCFF:FE01:F700 any** `;;`配置`DHCPv6`防护，匹配允许的访问列表。 `Device(config-dhcp-guard)#` **match serveraccess-list my_acls**	（可选）验证被通告的 DHCPv6 服务器或中继地址在授权的服务器访问列表中（访问列表中的目的地址是“any”）。如果未配置，此检查会被略过。空访问列表被当作允许地址处理。
步骤 5	[**no**] **match reply prefix-list** *ipv6-prefix-list-name* **示例：** `;;`假设预配置的`IPv6`前缀列表如下：	（可选）验证 DHCPv6 应答消息通告的前缀在配置的授权前缀列表中。如果未配置，此检查会被略过。空前缀列表被当作允许处理。

	Device(config)# **ipv6 prefix-list my_prefixpermit 2001:0DB8::/64 le 128** ;; 配置DHCPv6防护匹配前缀 Device(config-dhcp-guard)# **match replyprefix-list my_prefix**	
步骤 6	[**no**]**preference**{ **max** *limit* | **min** *limit* } **示例：** Device(config-dhcp-guard)# **preference max250** Device(config-dhcp-guard)#**preference min 150**	**device-role** 是 **server** 时，配置 **max** 和 **min** 来使用服务器优先级值过滤 DHCPv6 服务器通告。默认设置允许所有的通告。 **max** *limit*——（0 到 255）（可选）验证通告的优先级（优先级选项中）小于指定的限制。默认值是 255。如果未指定，该检查会被略过。 **min** *limit*——（0 到 255）（可选）验证通告的优先级（优先级选项中）大于指定的限制。默认值是 0。如果未指定，该检查会被略过。
步骤 7	[**no**] **trusted-port** **示例：** Device(config-dhcp-guard)# **trusted-port**	（可选）**trusted-port**——设置端口为可信模式。端口上不再执行策略。 **注释：**如果配置了可信端口，则 device-role 选项不可用。
步骤 8	**default** {**device-role** | **trusted-port**} **示例：** Device(config-dhcp-guard)# **defaultdevice-role**	（可选）**default**——设置命令为默认值。
步骤 9	**do show ipv6 dhcp guard policy** *policy_name* **示例：** Device(config-dhcp-guard)# **do show ipv6 dhcpguard policy example_policy**	（可选）在配置子模式中显示 IPv6 DHCP 防护策略配置。省略 *policy_name* 变量会显示所有 DHCPv6 策略。

DHCPv6 防护配置示例

```
enable
```

```
configure terminal

ipv6 access-list acl1

permit host FE80::A8BB:CCFF:FE01:F700 any

ipv6 prefix-list abc permit 2001:0DB8::/64 le 128

ipv6 dhcp guard policy pol1

device-role server

match server access-list acl1

match reply prefix-list abc

preference min 0

preference max 255

trusted-port

interface GigabitEthernet 0/2/0

switchport

ipv6 dhcp guard attach-policy pol1 vlan add 1

vlan 1

ipv6 dhcp guard attach-policy pol1

show ipv6 dhcp guard policy pol1
```

如何把 IPv6 DHCP 防护策略配置到接口或接口的 VLAN 上

在特权 EXEC 模式中，按照以下步骤配置 IPv6 DHCP 防护策略到接口或接口的 VLAN 上。

总步骤

1. configure terminal

2. interface Interface_type *stack/module/port*

3. ipv6 dhcp guard [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** |**remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* |**all**}]

4. do show running-config interface Interface_type *stack/module/port*

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：**	进入全局配置模式。

	Device# **configure terminal**	
步骤 2	**interface** Interface_type *stack/module/port* **示例：** Device(config)# **interface gigabitethernet 1/1/4**	指定接口类型及标识符，进入接口配置模式。
步骤 3	**ipv6 dhcp guard** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* |**add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] |**vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove***vlan_ids* | **all**}] **示例：** Device(config-if)# **ipv6 dhcp guard attach-policyexample_policy** 或 Device(config-if)# **ipv6 dhcp guard attach-policyexample_policy vlan 222,223,224** **或** Device(config-if)# **ipv6 dhcp guard vlan 222, 223,224**	把 DHCP 防护策略配置到接口或接口上的指定 VLAN。如果不使用 **attach-policy** 选项，默认策略会被配置。
步骤 4	**do show running-config interface** Interface_type *stack/module/port* **示例：** Device#(config-if)# **do show running-config gig 1/1/4**	在当前配置模式中确认策略配置到指定接口上。

如何把 IPv6 DHCP 防护策略配置到二层 EtherChannel 接口上

在特权 EXEC 模式中，按照以下步骤配置 IPv6 的 DHCP 防护策略到 EtherChannel 接口或 VLAN 上。

总步骤

1. configure terminal

2. interface range *Interface_name*

3. ipv6 dhcp guard [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** |**remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* |**all**}]

4. do show running-config interface*portchannel_interface_name*

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface range** *Interface_name* **示例：** `Device(config)#` **interface Po11**	指定创建 EtherChannel 时分配的端口通道接口名称。进入接口范围配置模式。 **提示：** 输入 **do show interfaces summary** 命令快速查看接口名称及类型。
步骤 3	**interface range** *Interface_name* **示例：** `Device(config)#` **interface Po11**	把 DHCP 防护策略配置到接口或接口上的指定 VLAN。如果不使用 **attach-policy** 选项，默认策略会被配置。
步骤 4	**do show running-config interface***portchannel_interface_name* **示例：** `Device#(config-if-range)#` **do show running-config** **int po11**	在当前配置模式中确认策略配置到指定接口上。

如何把 IPv6 DHCP 防护策略全局配置到 VLAN 上

在特权 EXEC 模式中，按照以下步骤配置 IPv6 DHCP 防护策略到覆盖多个接口的 VLAN 上。

总步骤

1. configure terminal

2. vlan configuration *vlan_list*

3. ipv6 dhcp guard [**attach-policy** *policy_name*]

4. do show running-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 2	**vlan configuration** *vlan_list* **示例：** `Device(config)# vlan configuration 334`	指定要配置 IPv6 侦听策略的 VLAN，并进入 VLAN 接口配置模式。
步骤 3	**ipv6 dhcp guard** [**attach-policy** *policy_name*] **示例：** `Device(config-vlan-config)#ipv6 dhcp guard attach-policy example_policy`	把 IPv6 DHCP 防护策略配置到指定的覆盖所有交换机以及堆栈接口的 VLAN。如果不使用 **attach-policy** 选项，默认策略会被配置。默认策略中，设备角色是 **client**，无可信接口。
步骤 4	**do show running-config** **示例：** `Device#(config-if)# do show running-config`	在当前配置模式中确认策略配置到指定 VLAN 上。

如何配置 IPv6 源防护

总步骤

1. **enable**
2. **configure terminal**
3. [**no**] **ipv6 source-guard policy** *policy_name*
4. [**deny global-autoconf**] [**permit link-local**] [**default**{. . . }] [**exit**] [**no**{. . . }]
5. **end**
6. **show ipv6 source-guard policy** *policy_name*

具体步骤

	命令或操作	目的
步骤 1	**enable**	启用特权 EXEC 模式，在提示时输入密

	示例： `Device>`**enable**	码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	[**no**] **ipv6 source-guard policy** *policy_name* **示例：** `Device(config)#` **ipv6 source-guard policy example_policy**	指定 IPv6 源防护策略的名称，并进入 IPv6 源防护策略配置模式。
步骤 4	[**deny global-autoconf**] [**permit link-local**][**default**{. . . }] [**exit**] [**no**{. . . }] **示例：** `Device(config-sisf-sourceguard)#` **deny global-autoconf**	（可选）定义 IPv6 源防护策略。 • **deny global-autoconf**——拒绝来源于自动配置的全局地址的数据流量。当链路上的所有全局地址都由 DHCP 分配，且管理员希望阻塞自动配置地址的主机发送流量时，此特性很有用。 • **permit link-local**——允许来源于链路本地地址的数据流量。 **注释：**源防护策略下不支持可信选项。
步骤 5	**end** **示例：** `Device(config-sisf-sourceguard)#` **end**	退出 IPv6 源防护策略配置模式。
步骤 6	**show ipv6 source-guard policy** *policy_name* **示例：** `Device#` **show ipv6 source-guard policy example_policy**	显示策略配置以及应用策略的所有端口。

接下来做什么？

把 IPv6 源防护策略应用到接口。

如何把 IPv6 源防护策略配置到接口上

总步骤

1. enable

2. configure terminal

3. interface Interface_type *stack/module/port*

4. ipv6 source-guard [**attach-policy** <*policy_name*>]

5. show ipv6 source-guard policy *policy_name*

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**interface** Interface_type *stack/module/port* **示例：** `Device(config)#` **interface gigabitethernet 1/1/4**	指定接口类型及标识符，并进入接口配置模式。
步骤 4	**ipv6 source-guard** [**attach-policy** <*policy_name*>] **示例：** `Device(config-if)#` **ipv6 source-guard attach-policy example_policy**	把 IPv6 源防护策略配置到接口上。如果不使用 **attach-policy** 选项，默认策略将被配置。
步骤 5	**show ipv6 source-guard policy** *policy_name* **示例：** `Device#(config-if)#` **show ipv6 source-guard policy example_policy**	显示策略配置及应用策略的所有接口。

如何把 IPv6 源防护策略配置到二层 EtherChannel 接口上

总步骤

1. enable

2. configure terminal

3. interface port-channel *port-channel-number*

4. ipv6 source-guard [**attach-policy** <*policy_name*>]

5. show ipv6 source-guard policy *policy_name*

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**interface port-channel** *port-channel-number* **示例：** `Device (config)#` **interface Po4**	指定接口类型及端口号，并进入端口通道配置模式。
步骤 4	**ipv6 source-guard** [**attach-policy** <*policy_name*>] **示例：** `Device(config-if) #` **ipv6 source-guard** **attach-policy example_policy**	把 IPv6 源防护策略配置到接口上。如果不使用 **attach-policy** 选项，默认策略将被配置。
步骤 5	**show ipv6 source-guard policy** *policy_name* **示例：** `Device(config-if) #`**show ipv6** **source-guard policy** **example_policy**	显示策略配置及应用策略的所有接口。

如何配置 IPv6 前缀防护

注释： 在应用前缀防护特性时，为了让路由协议能控制源自链路本地地址的数据包，应在

源防护策略配置模式中启用 permit link-local 命令。

总步骤

1. enable

2. configure terminal

3. [**no**] **ipv6 source-guard policy** *source-guard-policy*

4. [**no**] **validate address**

5. validate prefix

6. exit

7. show ipv6 source-guard policy [*source-guard-policy*]

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	[**no**] **ipv6 source-guard policy** *source-guard-policy* **示例：** `Device (config)#` **ipv6 source-guard policy my_snooping_policy**	定义 IPv6 源防护策略名称，并进入交换机集成安全特性源防护策略配置模式。
步骤 4	[**no**] **validate address** **示例：** `Device (config-sisf-sourceguard)#` **no validateaddress**	禁用验证地址特性，让 IPv6 前缀防护特性可以配置。
步骤 5	**validate prefix** **示例：** `Device (config-sisf-sourceguard)#` **validate prefix**	启用 IPv6 源防护特性，执行 IPv6 前缀防护操作。

步骤 6	**exit** **示例：** `Device (config-sisf-sourceguard)#` **exit**	退出交换机集成安全特性源防护策略配置模式，返回特权 EXEC 模式。
步骤 7	**show ipv6 source-guard policy** [*source-guard-policy*] **示例：** `Device #` **show ipv6 source-guard policy policy1**	显示 IPv6 源防护策略配置。

如何把 IPv6 前缀防护策略配置到接口上

总步骤

1. enable

2. configure terminal

3. interface Interface_type *stack/module/port*

4. ipv6 source-guard attach-policy *policy_name*

5. show ipv6 source-guard policy *policy_name*

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**interface** Interface_type *stack/module/port* **示例：** `Device(config)#` **interface gigabitethernet 1/1/4**	指定接口类型及标识符，并进入接口配置模式。
步骤 4	**ipv6 source-guard attach-policy** *policy_name* **示例：**	把 IPv6 源防护策略配置到接口上。如果不使用 **attach-policy** 选项，默认策略将被配置。

	Device(config-if)# **ipv6 source-guard attach-policyexample_policy**	
步骤 5	**show ipv6 source-guard policy** *policy_name* **示例：** Device(config-if)# **show ipv6 source-guard policyexample_policy**	显示策略配置及应用策略的所有接口。

如何把 IPv6 前缀防护策略配置到二层 EtherChannel 接口上

总步骤

1. enable

2. configure terminal

3. interface port-channel *port-channel-number*

4. ipv6 source-guard [**attach-policy** <*policy_name*>]

5. show ipv6 source-guard policy *policy_name*

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** Device>**enable**	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式。
步骤 3	**interface port-channel** *port-channel-number* **示例：** Device (config)# **interface Po4**	指定接口类型及端口号，并进入端口通道配置模式。
步骤 4	**ipv6 source-guard** [**attach-policy** <*policy_name*>] **示例：** Device(config-if)# **ipv6**	把 IPv6 源防护策略配置到接口上。如果不使用 **attach-policy** 选项，默认策略将被配置。

	source-guardattach-policy example_policy	
步骤 5	**show ipv6 source-guard policy** *policy_name* **示例：** `Device(config-if)#` **show ipv6** **source-guard policyexample_policy**	显示策略配置及应用策略的所有接口。

IPv6 第一跳安全配置示例

示例：如何把 IPv6 源防护策略配置到二层 EtherChannel 接口上

以下示例展示了如何把 IPv6 源防护策略配置到二层 EtherChannel 接口上。

```
Switch# configure terminal
Switch(config)# ipv6 source-guard policy POL
Switch(config-sisf-sourceguard) # validate address
switch(config-sisf-sourceguard)# exit
Switch(config)# interface Po4
Switch(config)# ipv6 snooping
Switch(config-if)# ipv6 source-guard attach-policy POL
Switch(config-if)# exit
switch(config)#
```

示例：如何把 IPv6 前缀防护策略配置到二层 EtherChannel 接口上

以下示例展示了如何把 IPv6 前缀防护策略配置到二层 EtherChannel 接口上。

```
Switch# configure terminal
Switch(config)# ipv6 source-guard policy POL
Switch (config-sisf-sourceguard)# no validate address
```

```
Switch((config-sisf-sourceguard)# validate prefix

Switch(config)# interface Po4

Switch(config-if)# ipv6 snooping

Switch(config-if)# ipv6 source-guard attach-policy POL
```

其他参考资料

相关文档

相关主题	**文档标题**
部署 IPv6 编址以及基本的连通性	http://www.icntnetworks.com
IPv6 网络管理以及安全主题	IPv6 配置库，Inspur INOS（Inspur 6850 交换机） http://www.icntnetworks.com
IPv6 命令参考手册	IPv6 命令参考手册，Inspur INOS（Inspur 6850 交换机） http://www.icntnetworks.com

错误信息解释

描述	**链接**
为帮助管理员搜索并解决该版本中的系统错误信息，管理员可使用错误信息解释工具。	http://www.icntnetworks.com

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要	http://www.icntnetworks.com

提供 icntnetworks.com 的用户 ID 及密码。	

配置 InspurTrustSec

关于 InspurTrustSec 的信息

InspurTrustSec 对网络中的用户、主机以及网络设备有强大的识别能力，能够提升 Inspur 网络设备的安全性。TrustSec 能够唯一地区分特定角色的数据流量，进行拓扑无关且可扩展的访问控制。该特性能够为被认证的对端建立信任关系，并加密对端之间的链路，进而确保数据保密性及完整性。

InspurTrustSec 的关键组件是 Inspur 身份服务引擎（Inspur Identity Services Engine，ISE）。可以使用 Inspur ISE 的 TrustSec 身份及安全组 ACL（Security Group ACL，SGACL）规划交换机策略，也可以手动进行交换机配置。

查询特性信息

要在交换机上配置 InspurTrustSec，请在以下 URL 查阅“InspurTrustSec 交换机配置指南”：

http://www.icntnetworks.com

InspurTrustSec 通用可用性版的版本注释 URL 如下：

http://www.icntnetworks.com

有关在 Inspur 6850 以及 6650 上的限制，请通过以下 URL 查看注释：

http://www.icntnetworks.com

有关 InspurTrustSec 方案的概览、数据表、平台特性矩阵以及示例学习等其他信息，请查看以下 URL：

http://www.icntnetworks.com

InspurTrustSec 特性

下表列出了最终会在启用 TrustSec 的 Inspur 交换机上实现的 TrustSec 特性。以后的 TrustSec 通用可用性版本会增加支持的交换机数量，并扩展每种交换机支持的特性数量。

InspurTrustSec 特性	**描述**
802.1AE 标记(MACsec)	基于 IEEE 802.1AE 的线速率逐跳二层加密协议。 在支持 MACsec 的设备之间，数据包在传输设备的出方向进行加密，在接收设备的入方向进行解密，在设备中的形式是明文。 此特性仅在硬件支持 TrustSec 的设备之间可用。 **注释：**此特性不支持在 Inspur 6850 以及 Inspur 6650 的 Inspur INOS 上使用。 **注释：**此特性不支持 Inspur 5960x。
终端准入控制（Endpoint Admission Control，EAC）	EAC 是对连接到 TrustSec 域的终端用户或设备的认证过程。EAC 通常在接入层交换机上进行。如果 EAC 认证及授权过程成功，会给用户或设备分配安全组标签。当前的 EAC 方式可以是 802.1x、MAC 旁路认证（MAB）以及 Web 认证代理（WebAuth）。
网络设备准入控制（Network Device Admission Control，NDAC）	TrustSec 域中的每台网络设备都可以使用 NDAC 验证对端设备的凭据以及可信度。NDAC 使用 IEEE 802.1x 基于端口认证的认证框架，并使用 EAP-FAST 作为 EAP 方式。NDAC 认证及授权过程成功后，安全关联协议会协商 IEEE 802.1AE 的加密方式。 **注释：**此特性不支持 Inspur 2960x。
安全组访问控制列表（Security Group Access Control List，SGACL）	安全组访问控制列表（SGACL）对安全组标签以及策略进行关联。对 TrustSec 域出方向有 SGT 标签的流量执行策略。

InspurTrustSec SGACL 高可用性	在支持 InspurStackWise 技术的交换机上，InspurTrustSec 安全组访问控制列表（SGACL）支持高可用性功能。InspurStackWise 技术提供了状态化的冗余性，允许交换机堆栈执行并处理访问控制条目。 启用此功能没有特定的 InspurTrustSec 配置。此特性仅支持 Inspur 6850 以及 6650 系列交换机。
安全关联协议（Security Association Protocol，SAP）	在 NDAC 认证之后，安全关联协议（SAP）会自动协商密钥及加密套件，为后续 TrustSec 对端之间的 MACsec 链路加密使用。SAP 在 IEEE 802.11i 中定义。 **注释：**此特性不支持在 Inspur 6850 以及 Inspur 6650 的 Inspur INOS 上使用。 **注释：**此特性不支持 Inspur 5960x。
安全组标签（Security Group Tag，SGT）	SGT 是一个 16 位的标签，表示 TrustSec 域中源的安全等级。该标签会被附加到以太网帧或 IP 数据包之后。
SGT 交换协议（SGT Exchange Protocol，SXP）	使用 SXP 时，硬件上不支持 TrustSec 的设备可以接收 Inspur 身份服务引擎（Inspur Identity ServicesEngine，ISE）或 Inspur 安全访问控制系统（Inspur Secure Access ControlSystem，ACS）发给被认证用户或设备的 SGT 属性。该设备随后可以给硬件支持 TrustSec 的设备发送源 IP 到 SGT 的绑定信息，支持的设备可以由此标记源流量，以执行 SGACL 策略。

当链路两端都支持 802.1AE MACsec 时，SAP 协商过程会发生。请求者与认证者之间会进行 EAPOL 密钥交换，以协商加密套件，交换安全参数并管理密钥。这些任务成功完成后，安全关联（SA）会被建立。

根据软件版本、授权以及链路硬件支持的不同，SAP 协商可以使用以下操作模式之一：

- 伽罗瓦计数器模式（Galois Counter Mode，GCM）——认证及加密
- GCM 认证（GCMA）——GCM 认证，无加密

- 无封装——无封装（明文）
- 空——封装，无认证或加密

InspurTrustSec 的特性信息

表 160：Inspur TrustSec 的特性信息

特性名称	版本	特性信息
• NDAC • SXPv1、SXPv2 • SGT • 二层执行 SGACL • 接口到 SGT 映射以及 VLAN 到 SGT 映射 • 子网到 SGT 映射 • 三层端口映射（Port Mapping，PM） • 三层身份端口映射（Identity PortMapping，IPM） • 安全组名称下载 • SXP 环路检测 • 基于策略的 CoA	Inspur INOS 11.3.1	这些特性在 Inspur 6850 以及 6650 交换机上引入。
SXPv1 以及 SXPv2	Inspur INOS 11.3.1	SXR 在 Inspur 2960-X 交换机上引入。
SXPv1 以及 SXPv2	Inspur INOS 11.3.1	SXR 在 Inspur 2960-XR 交换机上引入。

配置控制层限速

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

CoPP 的限制

控制层限速（control plane policing，CoPP）的限制包含如下几点：

- 仅支持入向 CoPP。**system-cpp-policy** 策略映射仅在控制层接口的入向可用。
- 仅可以把 **system-cpp-policy** 策略映射安装在控制层接口上。
- **system-cpp-policy** 策略映射以及 17 个系统定义的类不能被修改或删除。
- **system-cpp-policy** 策略映射下仅允许有 **police** 行为。而且，**police rate** 仅能按照数据包每秒（packets per second，pps）来配置。
- 每个类映射有一个或多个 CPU 队列。对于多个 CPU 队列属于一个类映射的情况，更改一个类映射的限速器速率会影响术语该类映射的所有 CPU 队列。相似的，禁用一个类映射会禁用所有属于该类映射的队列。关于每个类映射有哪些 CPU 队列的信息，请查看表 161：CoPP 的系统定义值。

关于控制层限速的信息

本章描述了控制层限速（CoPP）如何在设备上工作，以及如何对其进行配置。

CoPP 概述

CoPP 特性通过优先处理控制层及管理流量，保护 CPU 不受不必要的流量或 DoS 流量的影响，进而提升设备的安全性。

设备通常被划分为三个操作层，每层目标不同：

- 数据层，转发数据包。
- 控制层，正确路由数据。
- 管理层，管理网元。

可以使用 CoPP 来保护多数 CPU 处理的流量，以确保路由的稳定性、可达性，保证数据包正常送达。更重要的是，可以使用 CoPP 来保护 CPU 免受 DoS 攻击。

CoPP 使用模块化的 QoS 命令行界面（MQC）以及 CPU 队列来实现这些目标。不同类型的控制层流量会被基于特定的条件分为一组，并分配给一个 CPU 队列。可以通过配置专用的硬件限速器来管理这些 CPU 队列。例如，可以修改特定 CPU 队列（流量类型）的限速器速率，也可以禁用特定类型流量的限速器。

虽然限速器在硬件上配置，但 CoPP 不会影响 CPU 性能或数据层性能。然而，因为其限制了进入 CPU 的数据包数量，所以 CPU 的负载被控制了。这意味着等待来自硬件的数据包的服务能处理的入向数据包速率会更受控制（该速率可以由用户配置）。

CoPP 的系统定义功能

第一次启动设备时，系统会自动执行以下操作：

- 查找策略映射 **system-cpp-policy**。如果未检测到此策略映射，系统会创建并将其安装到控制层。
- 系统在 **system-cpp-policy** 之下创建 17 个类映射。

  下一次启动设备时，系统会检测到已经创建了策略及类映射。
- 安装策略后，默认会启用（32 个队列中的）16 个 CPU 队列，各使用自己的默认速率。

  默认启用的 CPU 队列及其默认速率在表 161：CoPP 的系统定义值中列出。

下表列出了启动设备时系统创建的类映射。表中列出了每个类映射对应的限速器以及分组在每个类映射下的一个或多个 CPU 队列。类映射与限速器之间是一对一映射；类映射与 CPU 队列之间是一对多映射。

表 161：CoPP 的系统定义值

类映射名称	限速器索引（限速器编号）	CPU 队列（队列编号）	是	默

			否默认启用CPU队列？	认限速器速率（数据包每秒，pps）
system-cpp-police-data	WK_CPP_POLICE_DA TA(0)	WK_CPU_Q_ICMP_GEN(3) WK_CPU_Q_BROADCAST(12)	是	200
system-cpp-police-l2-control	WK_CPP_POLICE_L2_CONTROL(1)	WK_CPU_Q_L2_CONTROL(1)	否	500
system-cpp-police-routing-control	WK_CPP_POLICE_ROUTING_CONTROL(2)	WK_CPU_Q_ROUTING_CONTROL(4)	是	500
system-cpp-police-control-low-priority	WK_CPP_POLICE_CONTROL_LOW_PRI(3)	WK_CPU_Q_ICMP_REDIRECT(6) WK_CPU_Q_GENERAL_PUNT(25)	否	500
system-cpp-police-punt-webauth	WK_CPP_POLICE_PUNT_WEBAUTH(7)	WK_CPU_Q_PUNT_WEBAUTH(22)	否	1000
system-cpp-police-topology-control	WK_CPP_POLICE_TOPOLOGY_CONTROL(8)	WK_CPU_Q_TOPOLOGY_CONTROL(15)	否	13000
system-cpp-police-multicast	WK_CPP_POLICE_MULTICAST(9)	WK_CPU_Q_TRANSIT_TRAFFIC(18) WK_CPU_Q_MCAST_DATA(	是	500

		30)		
system-cpp-police-sys-data	WK_CPP_POLICE_SYS_DATA(10)	WK_CPU_Q_LEARNING_CACHE_OVFL(13) WK_CPU_Q_CRYPTO_CONTROL(23) WK_CPU_Q_EXCEPTION(24) WK_CPU_Q_EGR_EXCEPTION(28) WK_CPU_Q_NFL_SAMPLED_DATA(26) WK_CPU_Q_GOLD_PKT(31) WK_CPU_Q_RPF_FAILED(19)	是	100
system-cpp-police-dot1x-auth	WK_CPP_POLICE_DOT1X(11)	WK_CPU_Q_DOT1X_AUTH(0)	否	1000
system-cpp-police-protocol-snooping	WK_CPP_POLICE_PR	WK_CPU_Q_PROTO_SNOOPING(16)	否	500
system-cpp-police-sw-forward	WK_CPP_POLICE_SW_FWD(13)	WK_CPU_Q_SW_FORWARDING_Q(14) WK_CPU_Q_SGT_CACHE_FULL(27) WK_CPU_Q_LOGGING(21)	是	1000
system-cpp-police-forus	WK_CPP_POLICE_FORUS(14)	WK_CPU_Q_FORUS_ADDR_RESOLUTION(5) WK_CPU_Q_FORUS_TRAFFIC(2)	否	1000
system-cpp-police-multicast-end-station	WK_CPP_POLICE_MULTICAST_SNOOPING(15)	WK_CPU_Q_MCAST_END_STATION_SERVICE(20)	是	2000
system-cpp-default	WK_CPP_POLICE_DEFAULT_POLICER	WK_CPU_Q_DHCP_SNOOPING	否	1000

		WK_CPU_Q_SHOW_FORWARD		

CoPP 的用户可配置功能

可以执行以下操作来管理控制层流量：

- 启用或禁用 CPU 队列。

  要启用 CPU 队列，需配置策略映射 **system-cpp-policy** 之下对应类映射的限速器行为（数据包每秒）。

  要禁用 CPU 队列，需移除策略映射 **system-cpp-policy** 之下对应类映射的限速器行为（数据包每秒）。
- 要更改限速器速率，需配置策略映射 **system-cpp-policy** 之下对应类映射的限速器行为（数据包每秒）。
- 要设置 CPU 队列为默认值，需在全局配置模式中输入 **cpp system-default** 命令。

如何配置 CoPP

启用 CPU 队列或更改限速器速率

启用 CPU 队列与更改 CPU 队列的限速器速率的过程相同，按如下步骤进行。

总步骤

1. enable

2. configure terminal

3. policy-map *policy-map-name*

4. class *class-name*

5. police rate *rate* **pps**

6. end

7. show running-config | begin system-cpp-policy

具体步骤

	命令或操作	目的
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密

	示例： `Device>`**enable**	码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**policy-map** *policy-map-name* **示例:** `Device(config)#` **policy-map system-cpp-policy** `Device(config-pmap)#`	进入策略映射配置模式。
步骤 4	**class** *class-name* **示例:** `Device(config-pmap)#` **class system-cpp-police-protocol-snooping** `Device(config-pmap-c)#`	进入类行为配置模式。输入希望启用的 CPU 队列对应的类名。参见表 161：CoPP 的系统定义值。
步骤 5	**police rate** *rate* **pps** **示例:** `Device(config-pmap-c)#` **police rate 100 pps**	指定特定类型流量每秒处理的入向数据包数量上限。 **注释：**指定的速率会应用到属于该类映射的所有 CPU 队列上。
步骤 6	**end** **示例:** `Device(config-pmap-c)#` **end**	返回特权 EXEC 模式。
步骤 7	**show running-config | begin system-cpp-policy** **示例:** `Device#` **show running-config | begin system-cpp-policy**	显示为不同流量类型配置的速率。

禁用 CPU 队列

按照以下步骤禁用 CPU 队列。

总步骤

1. enable

2. configure terminal

3. policy-map *policy-map-name*

4. class *class-name*

5. no police rate *rate* **pps**

6. end

7. show running-config | begin system-cpp-policy

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**policy-map** *policy-map-name* **示例:** `Device(config)#` **policy-map system-cpp-policy** `Device(config-pmap)#`	进入策略映射配置模式。
步骤 4	**class** *class-name* **示例:** `Device(config-pmap)#` **class system-cpp-police-protocol-snooping** `Device(config-pmap-c)#`	进入类行为配置模式。输入希望启用的 CPU 队列对应的类名。参见表 161：CoPP 的系统定义值。
步骤 5	**no police rate** *rate* **pps** **示例:** `Device(config-pmap-c)#` **no police rate 100 pps**	禁用对特定类型流量的入向数据包处理。 **注释：**此操作会禁用属于指定类映射的所有 CPU 队列。
步骤 6	**end** **示例:** `Device(config-pmap-c)#` **end**	返回特权 EXEC 模式。
步骤 7	**show running-config	begin**	显示为不同流量类型配置的速率。

	system-cpp-policy **示例:** `Device#` **show running-config | begin system-cpp-policy**	

为所有 CPU 队列配置默认限速器速率

按照以下步骤把所有 CPU 队列的限速器速率设置为默认值。

总步骤

1. enable

2. configure terminal

3. cpp system-default

4. end

5. show platform hardware fed switch *switch-number* **qos que stat internal cpu policer**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**cpp system-default** **示例:** `Device(config)#` **cpp system-default** `Defaulting CPP : Policer rate for all classes willbe set to their defaults`	把所有类的限速器速率设置为默认速率。
步骤 4	**end** **示例:** `Device(config-pmap-c)#` **end**	返回特权 EXEC 模式。
步骤 5	**show platform hardware fed switch** *switch-number* **qos questat internal cpu**	显示为不同流量类型配置的速率。

	policer **示例:** `Device#` **show platform hardware fed switch 1** **qos questat internal cpu policer**	

CoPP 配置示例

示例：启用 CPU 队列或更改 CPU 队列的限速器速率

此示例展示了如何启用 CPU 队列或更改 CPU 队列的限速器速率。示例中 CPU 队列 system-cpp-police-protocol-snooping 的限速器速率被设置为 100pps。

```
Device>enable
Device# configure terminal
Device(config)# policy-map system-cpp-policy
Device(config-pmap)# class system-cpp-police-protocol-snooping
Device(config-pmap-c)# police rate 100 pps
Device(config-pmap-c)# end
Device# show running-config | begin system-cpp-policy
policy-map system-cpp-policy
class system-cpp-police-data
police rate 200 pps
class system-cpp-police-sys-data
police rate 100 pps
class system-cpp-police-sw-forward
police rate 1000 pps
class system-cpp-police-multicast
police rate 500 pps
class system-cpp-police-multicast-end-station
police rate 2000 pps
class system-cpp-police-punt-webauth
class system-cpp-police-l2-control
```

```
class system-cpp-police-routing-control
police rate 500 pps
class system-cpp-police-control-low-priority
class system-cpp-police-topology-control
class system-cpp-police-dot1x-auth
class system-cpp-police-protocol-snooping
police rate 100 pps
class system-cpp-police-forus
class system-cpp-default
<输出已删节>
```

示例：禁用 CPU 队列

此示例展示了如何禁用 CPU 队列。示例中 CPU 队列 system-cpp-police-protocol-snooping 被禁用。

```
Device>enable
Device# configure terminal
Device(config)# policy-map system-cpp-policy
Device(config-pmap)# class system-cpp-police-protocol-snooping
Device(config-pmap-c)# no police rate 100 pps
Device(config-pmap-c)# end
Device# show running-config | begin system-cpp-policy
policy-map system-cpp-policy
class system-cpp-police-data
police rate 200 pps
class system-cpp-police-sys-data
police rate 100 pps
class system-cpp-police-sw-forward
police rate 1000 pps
class system-cpp-police-multicast
police rate 500 pps
class system-cpp-police-multicast-end-station
police rate 2000 pps
```

```
class system-cpp-police-punt-webauth

class system-cpp-police-l2-control

class system-cpp-police-routing-control

police rate 500 pps

class system-cpp-police-control-low-priority

class system-cpp-police-topology-control

class system-cpp-police-dot1x-auth

class system-cpp-police-protocol-snooping

class system-cpp-police-forus

class system-cpp-default

<输出已删节>
```

示例：为所有 CPU 队列配置默认限速器速率

此示例展示了如何为所有 CPU 队列配置默认限速器速率并验证设置。

```
Device>enable

Device# configure terminal

Device(config)# cpp system-default

Defaulting CPP : Policer rate for all classes will be set to their defaults

Device(config)# end

Device# show platform hardware fed switch 1 qos queue stats internal cpu policer

                                          (default) (set)

QId PlcIdx   Queue Name                Enabled      Rate     Rate     Drop

------------------------------------------------------------------------

0   11       DOT1X Auth                   No       1000     1000     0

1   1        L2 Control                   No       500      500      0

2   14       Forus traffic                No       1000     1000     0

3   0        ICMP GEN                     Yes      200      200      0

4   2        Routing Control              Yes      1800     1800     0

5   14       Forus Address resolution     No       1000     1000     0

6   3        ICMP Redirect                No       500      500      0

7   6        WLESS PRI-5                  No       1000     1000     0

8   4        WLESS PRI-1                  No       1000     1000     0
```

9	5	WLESS PRI-2	No	1000	1000	0
10	6	WLESS PRI-3	No	1000	1000	0
11	6	WLESS PRI-4	No	1000	1000	0
12	0	BROADCAST	Yes	200	200	0
13	10	Learning cache ovfl	Yes	100	100	0
14	13	Sw forwarding	Yes	1000	1000	0
15	8	Topology Control	No	13000	13000	0
16	12	Proto Snooping	No	500	500	0
17	16	DHCP Snooping	No	1000	1000	0
18	9	Transit Traffic	Yes	500	500	0
19	10	RPF Failed	Yes	100	100	0
20	15	MCAST END STATION	Yes	2000	2000	0
21	13	LOGGING	Yes	1000	1000	0
22	7	Punt Webauth	No	1000	1000	0
23	10	Crypto Control	Yes	100	100	0
24	10	Exception	Yes	100	100	0
25	3	General Punt	No	500	500	0
26	10	NFL SAMPLED DATA	Yes	100	100	0
27	2	SGT Cache Full	Yes	1800	1800	0
28	10	EGR Exception	Yes	100	100	0
29	16	Show frwd	No	1000	1000	0
30	9	MCAST Data	Yes	500	500	0
31	10	Gold Pkt	Yes	100	100	0

监控 CoPP

按照以下步骤显示限速器设置，如流量类型以及 CPU 队列的限速器速率（用户配置及默认速率）。

总步骤

1. enable

2. show platform hardware fed switch *switch-number* **qos que stat internal cpu policer**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**show platform hardware fed switch** *switch-number***qos que stat internal cpu policer**	显示为不同流量类型配置的速率。

```
Device>enable
Device# show platform hardware fed switch 3 qos queue stats internal cpu policer
(default) (set)
QId PlcIdx Queue Name Enabled Rate Rate Drop
------------------------------------------------------------------------------
0 11 DOT1X Auth No 1000 1000 0
1 1 L2 Control No 500 500 0
2 14 Forus traffic No 1000 1000 0
3 0 ICMP GEN Yes 200 200 0
4 2 Routing Control Yes 1800 1800 0
5 14 Forus Address resolution No 1000 1000 0
6 3 ICMP Redirect No 500 500 0
7 6 WLESS PRI-5 No 1000 1000 0
8 4 WLESS PRI-1 No 1000 1000 0
9 5 WLESS PRI-2 No 1000 1000 0
10 6 WLESS PRI-3 No 1000 1000 0
11 6 WLESS PRI-4 No 1000 1000 0
12 0 BROADCAST Yes 200 200 0
13 10 Learning cache ovfl Yes 100 100 0
14 13 Sw forwarding Yes 1000 1000 0
15 8 Topology Control No 13000 13000 0
16 12 Proto Snooping No 500 500 0
17 16 DHCP Snooping No 1000 1000 0
18 9 Transit Traffic Yes 500 500 0
19 10 RPF Failed Yes 100 100 0
20 15 MCAST END STATION Yes 2000 2000 0
```

```
21 13 LOGGING Yes 1000 1000 0

22 7 Punt Webauth No 1000 1000 0

23 10 Crypto Control Yes 100 100 0

24 10 Exception Yes 100 100 0

25 3 General Punt No 500 500 0

26 10 NFL SAMPLED DATA Yes 100 100 0

27 2 SGT Cache Full Yes 1800 1800 0

28 10 EGR Exception Yes 100 100 0

29 16 Show frwd No 1000 1000 0

30 9 MCAST Data Yes 500 500 0

31 10 Gold Pkt Yes 100 100 0
```

其他参考资料

相关文档

相关主题	**文档标题**
MQC QoS 命令及 CoPP**show** 命令	统一平台命令参考手册，Inspur INOS（Inspur 6650 交换机）

错误信息解释

描述	**链接**
为帮助管理员搜索并解决该版本中的系统错误信息，管理员可使用错误信息解释工具。	http://www.icntnetworks.com

标准以及 RFC

标准/RFC	**标题**
无	-

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。	http://www.icntnetworks.com

为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	

CoPP 的特性历史与信息

下表提供了本模块描述特性的版本信息。表中仅列出了引入特定特性支持的软件版本。除非另有说明，否则后续软件版本同样支持该特性。

特性名称	版本	特性信息
控制层限速（CoPP）或 CPP	Inspur INOS 11.3.1	此特性被引入。
CoPP 的 CLI 配置		此特性可由用户配置。可以使用 CLI 配置选项来启用或禁用 CPU 队列，更改限速器速率，并把限速器速率设置为默认值。

第 14 部分 堆栈管理器与高可用性

管理交换机堆栈

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

交换机堆栈的前提条件

交换机堆栈中的所有交换机都需要与活跃交换机运行相同的许可证等级。有关许可证等级的更多信息，参见*系统管理配置指南（Inspur 6650 交换机）*。
交换机堆栈中的所有交换机都需要运行兼容的软件版本。
要启用堆栈，必须在堆栈端口上安装 StackWise 适配器。有关交换机堆栈硬件注意事项的更多信息，参见 *Inspur 6650 交换机硬件安装指南*。

交换机堆栈的限制条件

交换机堆栈配置存在以下限制条件：

- 运行 LAN Base 许可证等级的交换机堆栈不支持三层特性。
- 一个交换机堆栈至多可以有九台兼容堆栈特性的交换机通过 StackWise-160 端口相连。
- 一个交换机堆栈中不能混用 Inspur 6850 与 Inspur 6650 交换机。
- 一个交换机堆栈中不能混用不同许可证等级。

关于交换机堆栈的信息

交换机堆栈概述

一个交换机堆栈至多可以有九台兼容堆栈特性的交换机通过 StackWise-160 端口相连。堆栈成员一起作为一个统一的系统工作。通过二层和三层协议看，整个交换机堆栈在网络中就是一个实体。

一个交换机堆栈中总是有一台活跃交换机以及一台备用交换机。如果活跃交换机不可用，备用交换机会取得活跃交换机的角色，并继续维持堆栈的运行。

活跃交换机控制着交换机堆栈的操作，是整个堆栈的管理点。在活跃交换机上，可以配置：

- 应用到全体堆栈成员的系统级别（全局）特性
- 每个堆栈成员的接口级别特性

活跃交换机会包含交换机堆栈已保存以及正在运行的配置文件。配置文件中包含交换机堆栈的系统级别设置以及每个堆栈成员的接口级别设置。每个堆栈成员都有一份当前文件的拷贝，以进行备份。

交换机堆栈支持的特性

活跃交换机上支持的系统级别特性在整个交换机堆栈上都受支持。

加密特性

如果活跃交换机上运行的是加密通用软件镜像（支持加密），那么交换机堆栈上就支持加密特性。

StackWise-160

堆栈成员使用 StackWise-160 技术作为统一的系统一同工作。通过二层和三层协议看，整个交换机堆栈在网络中就是一个实体。

注释： 运行 LAN Base 镜像的交换机不支持三层特性。

StackWise-160 的堆栈带宽可达 160 Gbps，通过状态化切换（stateful switchover，SSO）来提供堆栈内的弹性。堆栈的行为像是成员交换机选举出的一台活跃交换机管理着一个交换单元。活跃交换机会自动在堆栈内选举备用交换机。活跃交换机会创建并更新所有的交换、路由信息并时常与备用交换机同步这些信息。主备切换的过程中接入点仍能保持连接，除非接入点直接连接到了主交换机，这种情况下接入点会掉电并重启。一个工作中的堆栈可以接收新成员，也可以在不中断服务的情况下删除旧成员。

交换机堆栈成员

一台单独的设备是有一个堆栈成员，且该成员也作为活跃交换机工作的设备堆栈。可以把两

台单独的设备连接起来，创建包含两个堆栈成员的设备堆栈，其中，一台设备是活跃交换机。可以把一台设备连接到一个现有的设备堆栈，增加堆栈成员的数量。
所有堆栈成员之间都会收发 hello 消息。
- 如果一个堆栈成员不回应，该成员会被从堆栈中移除。
- 如果备用设备不回应，会选举出新的备用设备。
- 如果活跃设备不回应，备用设备会成为活跃设备。

此外，活跃设备和备用设备之间会收发 keepalive 消息。
- 如果备用设备不回应，会选举出新的备用设备。
- 如果活跃设备不回应，备用设备会成为活跃设备。

更换交换机堆栈成员

如果使用相同型号的交换机替换了一个堆栈成员，假设新交换机（称为预备交换机）与被替换的交换机使用相同的成员编号，则新交换机会使用与被替换交换机完全相同的配置进行工作。
在成员更换期间，交换机堆栈的操作可以继续进行而不被打断，除非更换了活跃交换机，或者添加了已开机的单独交换机或交换机堆栈。
添加已开机的交换机（合并）会导致所有交换机重载并选取新的活跃交换机。新选举出的活跃交换机会保留其角色及配置。所有其他交换机会保留自己的堆栈成员编号，并使用新选举活跃交换机的堆栈配置。
移除已开机的堆栈成员会把交换机堆栈划分（分割）成两个或多个交换机堆栈，每个堆栈都使用相同的配置。这会导致：
- 网络中 IP 地址冲突。如果希望交换机堆栈保持独立，需更改新创建的交换机堆栈的 IP 地址。
- 堆栈中两个成员的MAC地址冲突。可以使用**stack-mac updateforce**命令来解决冲突。

如果新创建的交换机堆栈没有活跃交换机或者备用交换机，该堆栈会重载并选取出一个新的活跃交换机。
注释： 确保对添加到交换机堆栈或从堆栈移除的交换机进行关机。
在添加或移除堆栈成员之后，确保该交换机堆栈按照全带宽（160 Gbps）运行。按住堆栈成员的Mode键，直到Stack模式LED灯亮起。堆栈中所有交换机最右两个端口的LED灯应该是绿色的。根据交换机型号不同，最右两个端口可以是10吉比特以太网端口或者小型可插拔（small form-factor pluggable，SFP）模块端口（10/100/1000端口）。如果任意交换机上这两个LED灯有不是绿色的情况，则堆栈没有全带宽运行。
如果移除了已开机的成员，但是不想分割堆栈：
- 关闭新创建的交换机堆栈中的交换机。
- 把它们通过堆栈端口重连到原来的交换机堆栈。
- 交换机开机。

关于影响交换机堆栈的连线及供电注意事项，参见*Inspur 6650交换机硬件安装指南*。

堆栈成员编号

堆栈成员编号（1到9）标识了设备堆栈中的每个成员。成员编号也确定了堆栈成员使用的接口级别的配置。可以使用EXEC命令**show switch**显示堆栈成员编号。

一台新的开箱即用的设备（没有加入过设备堆栈或还没有手动指定堆栈成员编号）默认的堆栈成员编号是1。加入设备堆栈时，该设备的默认堆栈成员编号会更改为堆栈中的最低可用成员编号。

一个堆栈中的堆栈成员不能使用相同的堆栈成员编号。每个堆栈成员，包括单独的设备，都会保持使用自己的成员编号，直到手动进行更改，或者编号已经被堆栈中其他成员使用。

- 如果使用命令**switch** *current-stack-member-number***renumber** *new-stack-member-number*手动更改了堆栈成员编号，只有在堆栈成员重置（或者输入了特权EXEC命令**reload slot** *stack-member-number*）且该编号没有分配给堆栈中其他成员时，新的编号才会生效。另一种更改堆栈成员编号的方式是更改Device_NUMBER环境变量。

  如果该编号已经被堆栈中的其他成员使用，设备会选用堆栈中最低可用的编号。

  如果手动更改了堆栈成员编号，且新堆栈成员没有相关联的接口级别配置，该堆栈员会重置为默认配置。

  不能在规划设备上使用**switch** *current-stack-member-number* **renumber** *new-stack-member-number*命令。如果执行此命令，命令会被拒绝。
- 如果把一个堆栈成员移动到不同的设备堆栈中，只在设备成员编号没有被堆栈中其他成员使用时，该成员才会保留其编号。如果编号被使用，该设备会选用堆栈中的最低可用编号。
- 在合并设备堆栈时，加入设备堆栈的新活跃交换机会选择堆栈中的最低可用编号。

如硬件安装指南中所述，可以使用 Stack 模式中的设备端口 LED 来可视化地确定每个堆栈成员的编号。

在 **default** 模式中，只有堆栈 master 的 Stack LED 才会闪烁绿灯。把 Mode 键切换至 **Stack** 选项时，所有堆栈成员的 Stack LED 都会亮绿灯。

当模式键被切换到 Stack 选项时，每个堆栈成员的交换机编号都会通过交换机前五个端口的 LED 显示。所有堆栈成员的交换机编号都以二进制形式显示。在交换机上，琥珀色的 LED 等表示 0，绿色的 LED 表示 1。

交换机编号 5（二进制为 00101）的示例如下：

交换机编号为 5 号的堆栈成员上前五个 LED 颜色组合如下：

- 端口 1：琥珀色
- 端口 2：琥珀色
- 端口 3：绿色
- 端口 4：琥珀色
- 端口 5：绿色

类似的，根据堆栈成员的交换机编号不同，前五个 LED 灯会亮琥珀色灯或亮绿灯。

注释：

- 如果把水平堆栈端口连接到另一端的正常网络端口，如果在 30 秒内没有从另一端接收到 SDP 包，堆栈端口的传输和接收会被禁用。
- 堆栈端口不会关闭，但传输和接收会被禁用。控制台上会显示以下日志消息。当对端网络端口转换为堆栈端口时，该堆栈端口的传输和接收才会被启用。

  ```
  %STACKMGR-4-HSTACK_LINK_CONFIG: Verify peer stack port setting for
  hstack StackPort-1 switch 5 (hostname-switchnumber)
  ```

堆栈成员优先级

堆栈成员使用更高的优先级会增加其被选举为活跃交换机的概率，并有助于保留自己的堆栈

成员编号。优先级值可以是1到15之间。默认的优先级值是1。可以使用EXEC命令**show switch**显示堆栈成员优先级值。

注释： 建议把最高优先级值分配给期望成为活跃交换机的设备。这保证在重新选举发生时，设备会被选举为活跃交换机。

要更改堆栈成员的优先级值，使用**switch** *stack-member-number* **priority** *newpriority-value*命令。更多信息参见“设置堆栈成员优先级”一节。

新的优先级会立即生效，但不会影响当前的活跃交换机。在当前的活跃交换机或者交换机堆栈重置的时候，新的优先级值有助于决定哪个堆栈成员会被选举为新的活跃交换机。

交换机堆栈网桥 ID 以及 MAC 地址

交换机堆栈在网络中通过*网桥 ID（bridge ID）*进行标识，如果作为三层设备运行，则使用路由器 MAC 地址标识。网桥 ID 以及路由器 MAC 地址由活跃交换机的 MAC 地址决定。

如果活跃交换机变化，新活跃交换机的 MAC 地址会决定新的网桥 ID 以及路由器 MAC 地址。

如果整个交换机堆栈重载，交换机堆栈会使用活跃交换机的 MAC 地址。

交换机堆栈的持续 MAC 地址

可以使用持续 MAC 地址特性设置堆栈 MAC 地址改变的时延。在此时间之内，如果之前的活跃交换机重新加入了堆栈，即使该交换机现在是堆栈成员而不是活跃交换机，交换机堆栈也会继续使用其 MAC 地址作为堆栈的 MAC 地址。如果之前的活跃交换机在此时间段内没有重新加入堆栈，交换机堆栈会采用新活跃交换机的 MAC 地址作为堆栈的 MAC 地址。默认情况下，堆栈的 MAC 地址会是首个活跃交换机的 MAC 地址，即使有新的活跃交换机接替其角色。

也可以配置堆栈 MAC 地址的持久性，使堆栈 MAC 地址永远不会更改为新活跃交换机的 MAC 地址。

活跃及备用交换机的选举和重新选举

所有堆栈成员都有资格成为活跃交换机或备用交换机。如果活跃交换机不可用，备用交换机会成为活跃交换机。

活跃交换机会保持其角色，除非以下事件发生：

- 交换机堆栈被重置。
- 活跃交换机被从交换机堆栈移除。
- 活跃交换机被重置或关机。
- 活跃交换机故障。
- 添加了已开机的单独交换机或交换机堆栈，交换机堆栈成员增加。

活跃交换机会按序根据以下因素进行选举或重新选举：

1 交换机当前是活跃交换机。

2 交换机有最高的堆栈成员优先级。

注释： 建议把最高优先级值分配给期望成为活跃交换机的设备。这保证在重新选举发生时，设备会被选举为活跃交换机。

3 交换机有最短的启动时间。

4　交换机有最低的 MAC 地址。

注释：　选举或重新选举新的备用交换机的因素与活跃交换机的相同，且适用于除了活跃交换机之外的所有参与交换机。

在选举之后，新的活跃交换机会在几秒之后可用。在此期间，交换机堆栈使用内存中的转发表来最小化网络中断。在新的活跃交换机选举或重置期间，其他可用的堆栈成员的物理接口不受影响。

当之前的活跃交换机变为可用状态时，它*不会*假定自己是活跃交换机的角色。

如果开启或重置了整个交换机堆栈，一些堆栈成员可能不会参与到活跃交换机的选举过程中。在相同的 2 分钟时间范围内开机的堆栈成员会参与活跃交换机的选举过程，并有机会成为活跃交换机。在 120 秒时间范围之后启用的堆栈成员不会参与初始选举过程，因而成为堆栈成员。关于影响活跃交换机选举的启动注意事项，参见交换机硬件安装指南。

如硬件安装指南中所述，可以通过交换机上的 ACTV LED 来查看交换机是否是活跃交换机。

交换机堆栈配置文件

活跃交换机上有交换机堆栈已保存的以及正在运行的配置文件。备用交换机会自动接收同步过来的运行配置文件。当运行配置文件被保存在启动配置文件中时，堆栈成员会同步地进行拷贝。如果活跃交换机变为不可用状态，备用交换机会接替其角色，并使用当前的运行配置。

配置文件记录这些配置：

- 系统级别（全局）配置，如 IP、STP、VLAN 以及、SNMP 等应用于所有堆栈成员的设置。
- 针对每个堆栈成员的堆栈成员接口相关配置。

注释：　如果活跃交换机被替换，且没有把运行配置保存在启动配置中，活跃交换机的接口相关配置会被保存。

一台新的开箱即用的设备在加入交换机堆栈时会使用该堆栈的系统级别设置。如果一台设备在启动之前被移动到不同的交换机堆栈中，该设备会丢失已保存的配置文件，并使用新交换机堆栈的系统级别配置。如果设备在加入新交换机堆栈之前被启动作为一台单独的设备使用，交换机堆栈会重载。堆栈重载时，新设备可能成为活跃交换机，保留其配置并覆盖其他堆栈成员的配置文件。

每个堆栈成员的接口相关配置都与堆栈成员编号相关联。堆栈成员会保持使用其编号，除非手动更改或该编号已被相同交换机堆栈中的其他成员使用。如果交换机成员编号改变，新的编号会在堆栈成员重置以后生效。

- 如果不存在该成员编号的接口相关配置，该堆栈成员会使用自己默认的接口相关配置。
- 如果存在该成员编号的接口相关配置，该堆栈成员会使用与该成员编号关联的接口相关配置。

如果使用相同型号的设备替换了一个故障的成员，替换设备会自动使用与故障设备相同的接口相关配置，管理员无需重新配置接口设置。替换设备（称为规划设备）必须与故障设备使用相同的堆栈成员编号。

可以按照备份及恢复单独设备配置的方式来进行堆栈配置的备份和恢复。

通过离线配置规划堆栈成员

可以使用离线配置特性在新交换机加入堆栈之前对其进行*规划*（提供配置）。可以配置堆栈成员编号、交换机类型以及与当前不是堆栈一部分的交换机相关的接口。在交换机堆栈上创

建的配置被称为*规划配置*。被添加到交换机堆栈且接收此配置的交换机被称为*规划交换机*。管理员可以使用全局配置命令 **switch** *stack-member-number* **provision** *type* 手动创建规划配置。在把规划交换机添加到堆栈之前，必须更改 *stack-member-number*，且必须与为堆栈中新交换机创建的堆栈成员编号相同。规划配置中的交换机类型必须与新添加交换机的类型相同。当一台交换机被添加到交换机堆栈，且不存在规划配置时，规划配置会被自动创建。

在配置与规划交换机相关的接口时，交换机堆栈会接受配置，且此信息会出现在运行配置中。然而，因为该交换机不是活跃状态，任何与其接口相关的配置都不会运行，且与规划交换机相关的接口不会出现在特定特性的显示信息中。比如，与规划交换机相关的 VLAN 配置信息不会出现在交换机堆栈的 **show vlan** 用户 EXEC 命令输出中。

无论规划交换机是否是堆栈的一部分，交换机堆栈都会把规划配置保留在运行配置中。可以输入 **copy running-config startup-config** 特权 EXEC 命令把规划配置保存到启动配置文件中。启动配置文件确保交换机堆栈可以重载，且无论规划交换机是不是交换机堆栈的一部分，都能使用已保存的信息。

把规划交换机添加到交换机堆栈的影响

把规划设备添加到交换机堆栈时，堆栈会应用规划配置或默认配置。下表列出了交换机堆栈在对比规划配置以及规划交换机时会发生的事件。

表 169：比较规划配置以及规划交换机的结果

场景		**结果**
堆栈成员编号和设备类型匹配。	1 如果规划交换机的堆栈成员编号与堆栈中规划配置的堆栈成员编号相同，且 2 如果规划交换机的设备类型与堆栈中规划配置的设备类型相同。	交换机堆栈把规划配置应用到规划交换机上，并将其加入堆栈。
堆栈成员编号匹配但设备类型不匹配。	1 如果规划交换机的堆栈成员编号与堆栈中规划配置的堆栈成员编号相同，但是 2 规划交换机的设备类型与堆栈中规划配置的设备类型不同。	交换机堆栈会把默认配置应用到规划交换机上，并将其加入堆栈。 规划配置会被改变以反映新信息。
规划配置中未找到堆栈成员编号。		交换机堆栈会把默认配置应用到规划交换机上，并将其加入堆栈。 规划配置会被改变以反映新信息。
规划配置中未找到规划交换机的堆栈成员编号。		交换机堆栈会把默认配置应用到规划交换机上，并将其加入堆栈。

如果向已关机的堆栈中添加了一台和规划配置中型号不同的规划交换机，启动堆栈后，交换机堆栈会拒绝启动配置中（不正确的）的**switch***stack-member-number* **provision** *type*命令。然而，在堆栈初始化期间，启动配置文件中规划接口（可能有类型错误）的非默认接口配置信息会被执行。根据实际设备类型以及之前规划的交换机类型的差别不同，一些命令可能被拒绝，而一些命令会被接受。

注释： 如果交换机堆栈不含有新设备的规划配置，设备会使用默认接口配置加入堆栈。交换机堆栈随后会添加与新设备匹配的全局配置命令**switch** *stack-member-number* **provision** *type* 到运行配置中。更多配置信息，参见*规划交换机堆栈的新成员*一节。

替换交换机堆栈中规划交换机的影响

交换机堆栈中的规划交换机故障时，可以把它从堆栈中移除并使用另一台设备代替，堆栈对其应用规划配置或默认配置。交换机对比规划配置以及规划交换机时发生的事件与添加规划交换机到堆栈时的事件相同。

移除交换机堆栈中规划交换机的影响

如果从设备堆栈中移除了一台规划交换机，与移除的堆栈成员相关的配置会作为规划信息保留在运行配置中。要完全移除配置，使用**no switch** *stack-member-number* **provision**全局配置命令。

升级运行不兼容软件的交换机

自动升级以及自动建议特性让使用与交换机堆栈不兼容软件包的交换机可以升级到兼容的软件版本，以便加入交换机堆栈。

自动升级

自动升级特性的目的是让交换机升级到兼容的软件镜像，使交换机能加入交换机堆栈。

当一台新交换机尝试加入一个交换机堆栈时，每个堆栈成员都会进行与新交换机的兼容性检查。每个堆栈成员会把兼容性检查的结果发送给活跃交换机，由它来使用这些结果确定交换机是否能够加入交换机堆栈。如果新交换机上的软件与交换机堆栈不兼容，新交换机会进入版本不匹配（version-mismatch，VM）模式。

如果在现有交换机堆栈上启用了自动升级特性，活跃交换机会自动升级新交换机，让它使用与兼容的堆栈成员相同的软件镜像。检测到不匹配的软件几分钟之后，自动升级会开始。

自动升级默认被禁用。

自动升级包括自动拷贝过程以及自动提取过程。

- 自动拷贝过程会自动地把运行在任意堆栈成员上的软件镜像拷贝到新交换机上以自动升级。如果启用自动拷贝，新交换机有足够的闪存空间，而且交换机堆栈运行的软件镜像适用于新交换机时，自动拷贝会进行。

注释： VM模式中的交换机可能无法运行所有的软件版本。例如，新交换机的硬件在较早的软件版本中无法识别。

- 当自动升级过程不能在堆栈中找到合适的软件拷贝给新交换机时，自动提取过程会发生。此时，自动提取进程会搜索堆栈中的所有交换机，查找升级软件堆栈或新交换所需的bin文件。这个bin文件可以在交换机堆栈或新交换机的任意闪存文件系统中。如果在堆栈成员上找到了适用于新交换机的bin文件，此进程会提取文件并自动升级新交换机

自动升级特性在捆绑模式中不可用。交换机堆栈必须运行在安装模式中。如果交换机堆栈运行在捆绑模式中，使用特权EXEC命令**software expand**更改为安装模式。

可以在新交换机上使用全局配置命令**software auto-upgrade enable**启用自动升级特性。可以使用特权EXEC命令**show running-config**，通过查看输出中的*Auto upgrade*一行来检查自动升级状态。

可以使用全局配置命令**software auto-upgrade source url**，配置自动升级特性使用特定的软件包来升级新交换机。如果软件包不合法，新交换机会使用与兼容堆栈成员相同的软件镜像来进行升级。

自动升级过程完成时，新交换机会重启并作为全功能的成员加入堆栈。如果在重启期间两条堆栈线缆都连接上，则不会发生网络断开，因为交换机堆栈按照两路环模式工作。

关于升级运行不兼容软件交换机的更多信息，参见*Inspur INOS文件系统、配置文件及软件包文件附录（Inspur 6650交换机）*。

自动建议

以下情况发生时，自动建议特性会被触发：

- 自动升级特性被禁用。
- 新交换机在捆绑模式，而堆栈在安装模式。自动建议会显示syslog消息，说明使用特权EXEC命令**software auto-upgrade**把新交换机改为安装模式。
- 堆栈在捆绑模式。自动建议会显示syslog，说明在捆绑模式中重启新交换机，让其能够加入堆栈。
- 因为新交换运行不兼容的软件，自动升级尝试失败。在交换机堆栈对新交换机执行了兼容性检查之后，自动建议会显示syslog，说明新交换机是否可以自动升级。

自动建议不能被禁用。当交换机堆栈的软件和版本不匹配（VM）模式中的交换机软件不含有相同许可证等级时，自动建议特性*不会*给出建议。

自动建议消息示例

示例：自动升级被禁用且不兼容交换机尝试加入

以下自动建议示例展示了当自动升级特性被禁用，且不兼容的switch1尝试加入交换机堆栈时显示的系统消息：

```
*Oct 18 08:36:19.379: %INSTALLER-6-AUTO_ADVISE_SW_INITIATED: 2 installer: Auto advise initiated for switch 1
*Oct 18 08:36:19.380: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer: Searching stack for softwareto upgrade switch 1
*Oct 18 08:36:19.382: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer: Switch 1 with incompatiblesoftware has been
*Oct 18 08:36:19.382: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer: added to the stack. Thesoftware running on
*Oct 18 08:36:19.382: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer: all stack members was scanned and it has been
*Oct 18 08:36:19.382: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer: determined that the
```

```
'softwareauto-upgrade'
*Oct 18 08:36:19.382: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer: command can be used to
install compatible
*Oct 18 08:36:19.382: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer: software on switch 1.
```

示例：自动升级被禁用且新交换机在捆绑模式中

以下自动建议示例展示了当自动升级特性被禁用，且运行捆绑模式的交换机尝试加入运行安装模式的堆栈时显示的系统消息：

```
*Oct 18 11:09:47.005: %INSTALLER-6-AUTO_ADVISE_SW_INITIATED: 2 installer: Auto advise
initiated for switch 1
*Oct 18 11:09:47.005: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer: Switch 1 running
bundledsoftware has been added
*Oct 18 11:09:47.005: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer: to the stack that is
runninginstalled software.
*Oct 18 11:09:47.005: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer: The 'software
auto-upgrade'command can be used to
*Oct 18 11:09:47.005: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer: convert switch 1 to
theinstalled running mode by
*Oct 18 11:09:47.005: %INSTALLER-6-AUTO_ADVISE_SW: 2 installer: installing its running
software.
```

交换机堆栈管理连通性

管理员可以通过活跃交换机管理交换机堆栈以及堆栈成员接口。可以使用CLI、SNMP以及支持的网络管理应用，如InspurWorks。不能基于独立的设备管理堆栈成员。

注释： 可以使用SNMP管理支持的MIB中定义的堆栈网络特性。交换机不支持使用SNMP管理堆栈特定的特性，比如堆栈成员以及选举。

通过 IP 地址到交换机堆栈的连通性

交换机堆栈通过一个IP地址进行管理。这个IP地址是系统级别的设置，不针对活跃交换机或者任何其他的堆栈成员。只要有IP连通性，就算从堆栈中移除了活跃交换机或者任何其他的堆栈成员，管理员仍然可以通过相同的IP地址管理堆栈。

注释： 把堆栈成员从交换机堆栈中移除后，堆栈成员会保留堆栈IP地址。为了避免网络中两台设备使用相同IP地址造成的冲突，请更改从交换机堆栈中移除设备的IP地址。

有关交换机堆栈配置的信息，参见*交换机堆栈配置文件*一节。

通过控制台端口或以太网管理端口到交换机堆栈的连通性

可以使用以下方式之一连接到活跃交换机：

- 可以通过一个或多个堆栈成员的控制台端口，使用终端或PC连接活跃交换机。
- 可以通过一个或多个堆栈成员的以太网管理端口，使用PC连接活跃交换机。关于通过以太网管理端口连接交换机堆栈的更多信息，参见*使用以太网管理端口*一节。

可以通过一个或多个堆栈成员的控制台端口，使用终端或PC连接到堆栈master。
要当心使用多个CLI会话连接活跃交换机的情况。在一个会话中输入的命令不会在另一个会话中显示。因此，管理员可能无法分辨输入了命令的会话。
建议在管理交换机堆栈时只使用一个CLI会话。

如何配置交换机堆栈

启用持续 MAC 地址特性

注释： 输入命令配置此特性时，警告消息会显示此配置的后果。应该小心地使用此特性。在同一域中的其他位置使用老活跃交换机的 MAC 地址可能导致流量丢失。
按照以下步骤启用持续 MAC 地址特性：
总步骤
1. enable
2. configure terminal
3. stack-mac persistent timer [**0** | *time-value*]
4. end
5. copy running-config startup-config
具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**stack-mac persistent timer** [**0** | *time-value*] **示例：** `Device(config)#` **stack-mac persistent timer 7**	设置在活跃交换机变化时，堆栈 MAC 地址变为新活跃交换机地址的时延。如果在此期间之前的活跃交换机加入了堆栈，堆栈使用该地址作为堆栈 MAC 地址。 • 不输入值或值为 0 的命令，无限期地继续使用当前活跃交换机的 MAC 地址。 • 输入 1 到 60 分钟的 *time-value*，配置堆栈 MAC 地址变为新活跃交换机地址的时间。 堆栈会使用之前活跃交换机的 MAC 地址，直到配置的时间过期。
步骤 4	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。

步骤 5	**copy running-config startup-config** **示例：** `Device#` **copy running-configstartup-config**	（可选）把配置的条目保存到配置文件中。

接下来做什么？

使用全局配置命令 **no stack-mac persistent timer** 来禁用持续 MAC 地址特性。

分配堆栈成员编号

此选项仅可以在活跃交换机上执行。

按照以下步骤给堆栈成员分配成员编号：

总步骤

1. enable

2. configure terminal

3. switch *current-stack-member-number* **renumber** *new-stack-member-number*

4. end

5. reload slot *stack-member-number*

6. show switch

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**switch** *current-stack-member-number* **renumber***new-stack-member-number* **示例：** `Device(config)#` **switch 3 renumber 4**	指定当前的堆栈成员编号以及堆栈成员的新编号。范围从 1 到 9。 可以使用用户 EXEC 命令 **show switch** 显示当前的堆栈成员编号。
步骤 4	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 5	**reload slot** *stack-member-number* **示例：** `Device#` **reload slot 4**	重置堆栈成员。
步骤 6	**show switch** **示例：** `Device#`**show switch**	验证堆栈成员编号。
步骤 7	**copy running-config startup-config** **示例：** `Device#` **copy running-configstartup-config**	（可选）把配置的条目保存到配置文件中。

设置堆栈成员优先级

此选项仅可以在活跃交换机上执行。
按照以下步骤给堆栈成员分配优先级值。
总步骤
1. enable
2. switch *stack-member-number* **priority** *new-priority-number*
3. show switch *stack-member-number*
4. copy running-config startup-config
具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**switch** *stack-member-number* **priority***new-priority-number* **示例：** `Device#` **switch 3 priority 2**	指定堆栈成员编号以及堆栈成员的新优先级。堆栈成员编号范围从 1 到 9。优先级值范围从 1 到 15。 可以使用用户 EXEC 命令 **show switch** 显示当前的优先级值。 新优先级值会立刻生效，但不会影响当前的活跃交换机。在当前活跃交换机或堆栈重置时，新优先级值有助于确定哪个堆栈成员会被选为新的活跃交换机。
步骤 3	**show switch** *stack-member-number* **示例：** `Device#` **show switch**	验证堆栈成员的优先级值。
步骤 4	**copy running-config startup-config** **示例：** `Device#` **copy running-configstartup-config**	（可选）把配置的条目保存到配置文件中。

规划交换机堆栈的新成员

此选项仅可以在活跃交换机上执行。
总步骤
1. show switch
2. configure terminal
3. switch *stack-member-number* **provision** *type*
4. end
5. copy running-config startup-config
具体步骤

	命令或操作	目的

步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**switch** *stack-member-number* **provision** *type* **示例：** `Device(config)#` **switch 3 provision WS-xxxx**	指定预配置交换机的堆栈成员编号。默认情况下，无规划交换机。 *stack-member-number* 的范围从 1 到 9。指定交换机堆栈中未被使用的交换机成员编号，见步骤 1。 *type* 字段输入命令行帮助字符串中列出的支持交换机型号。
步骤 4	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 5	**copy running-config startup-config** **示例：** `Device#` **copy running-configstartup-config**	（可选）把配置的条目保存到配置文件中。

移除规划交换机信息

在开始前，必须把规划交换机从堆栈中移除。此选项仅可以在活跃交换机上执行。

总步骤

1. configure terminal

2. no switch *stack-member-number* **provision**

3. end

4. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**no switch** *stack-member-number* **provision** **示例：** `Device(config)#` **no switch 3 provision**	移除指定成员的规划信息。
步骤 3	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 4	**copy running-config startup-config** **示例：** `Device#` **copy running-configstartup-config**	（可选）把配置的条目保存到配置文件中。

如果要移除规划交换机的堆栈有如下配置：

- 堆栈有四个成员
- 堆栈成员 1 是活跃交换机
- 堆栈成员 3 是规划交换机

且希望移除规划信息并避免收到错误消息，可以移除堆栈成员3的电源，断开堆栈成员3号连接的StackWise-160线缆，重连其余交换机之间的线缆，并输入**no switch***stack-member-number* **provision**全局配置命令。

显示堆栈中的不兼容交换机

总步骤

1. show switch

具体步骤

	命令或操作	目的
步骤 1	**show switch** **示例：** `Device#` **show switch**	显示交换机堆栈中的不兼容交换机（ 'Current State' 为 'V-Mismatch' ）。V-Mismatch 状态标识出了软件不兼容的交换机。输出中显示的 Lic-Mismatch 表示与活跃交换机运行不同许可证级别的交换机。 有关管理许可证级别的更多信息，参见*系统管理配置指南（Inspur 6650 交换机）*。

升级交换机堆栈中的不兼容交换机

总步骤

1. software auto-upgrade

2. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**software auto-upgrade** **示例：** `Device#` **software auto-upgrade**	升级交换机堆栈中的不兼容交换机，或把捆绑模式中的交换机改为安装模式。
步骤 2	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置的条目保存到配置文件中。

交换机堆栈故障排除

临时禁用堆栈端口

如果堆栈端口抖动且造成堆栈环不稳定，要禁用端口，输入特权EXEC命令**switch***stack-member-number* **stack port** *port-number* **disable**。要重新启用端口，输入命令**switch** *stack-member-number* **stack port** *port-number* **enable**。

注释： 要小心使用**switch** *stack-member-number* **stack port** *port-number* **disable**命令。禁用堆栈端口时，堆栈会使用半速带宽工作。

当所有成员都通过堆栈端口连接，且都在就绪状态中时，堆栈为全环状态。

以下情况发生时，堆栈为部分环状态：

- 所有成员都通过堆栈端口连接，但是一些成员不是就绪状态。
- 一些成员未通过堆栈端口连接。

总步骤

1. **switch** *stack-member-number* **stack port** *port-number* **disable**
2. **switch** *stack-member-number* **stack port** *port-number* **enable**

具体步骤

	命令或操作	目的
步骤 1	**switch** *stack-member-number* **stack port** *port-number* **disable** **示例：** `Device# switch 2 stack port 1 disable`	禁用特定的堆栈端口。
步骤 2	**switch** *stack-member-number* **stack port** *port-number* **enable** **示例：** `Device# switch 2 stack port 1 enable`	重新启用堆栈端口。

要禁用堆栈端口且堆栈在全环状态时，只能禁用一个堆栈端口。会显示以下信息：

```
Enabling/disabling a stack port may cause undesired stack changes. Continue?[confirm]
```

要禁用堆栈端口且堆栈在部分环状态时，不能禁用端口。会显示以下信息：

```
Disabling stack port not allowed with current stack configuration.
```

在另一个成员启动时重新启用堆栈端口

交换机1上的堆栈端口1连接到了交换机4上的端口2。如果端口1抖动，可以使用特权EXEC命令**switch 1 stack port 1 disable**禁用端口1。当交换机1上的端口1被禁用且交换机1仍然开机时，按照以下步骤重新启用堆栈端口：

步骤1 断开交换机1上的堆栈端口1与交换机4上的端口2之间的连线。

步骤2 从堆栈中移除交换机4。

步骤3 添加一个交换机来代替交换机4并给它分配交换机编号4。

步骤4 重连交换机1上的堆栈端口1与交换机4（替换交换机）上的端口2之间的连线。

步骤5 重新启用交换机之间的链路。输入特权EXEC命令**switch 1 stack port 1 enable**启用交换

机1上的端口1。

步骤6　启动交换机4。

注意：　在启用交换机1的端口1之前启动交换机4可能导致一台交换机重启。如果先启动交换机4，可能需要输入**switch 1 stack port 1 enable**以及**switch4 stack port 2 enable**特权EXEC命令来启用链路。

监控设备堆栈

表170：显示堆栈信息的命令

命令	**描述**
show switch	显示堆栈的汇总信息，包括规划交换机状态以及版本不匹配模式中的交换机。
show switch *stack-member-number*	显示特定成员的信息。
show switch detail	显示堆栈的详细信息。
show switch neighbors	显示堆栈邻居。
show switch stack-ports [**summary**]	显示堆栈的端口信息。
show redundancy	显示冗余系统及当前处理器信息。冗余系统信息包括系统工作时长、备用故障、切换原因、硬件、配置冗余模式及运行冗余模式。显示的当前处理器信息包括活跃位置、软件状态以及处于当前状态中的时长。
show redundancy state	显示活跃及备用设备的所有冗余状态。

交换机堆栈配置示例

交换机堆栈配置场景

以下多数交换机堆栈配置场景都假设至少有两台设备通过其StackWise-160端口相连。

表 171：配置场景

场景		**结果**
活跃交换机选举由现有活跃交换机决定	通过 StackWise-160 端口连接两个开机的交换机堆栈。	两个活跃交换机中只有一个能成为新的活跃交换机。

活跃交换机选举由堆栈成员优先级值决定	1 通过 StackWise-160 端口连接两台交换机。 2 使用全局配置命令**switch***stack-member-number* **priority***new-priority-number*来设置一个交换机成员使用更高的成员优先级值。 3 同时重启两个堆栈成员。	有更高优先级值的堆栈成员会被选举为活跃交换机。
活跃交换机选举由配置文件决定	假设两个堆栈成员有相同的优先级值： 1 确保一个堆栈成员有默认配置，而另一个堆栈成员有保存（非默认）的配置文件。 2 同时重启两个堆栈成员。	有保存配置文件的堆栈成员会被选举为活跃交换机。
活跃交换机选举由MAC地址决定	假设两个堆栈成员有相同的优先级值、配置文件及特性集，同时重启两个堆栈成员股。	有较低MAC地址的堆栈成员会被选举为活跃交换机。
堆栈成员编号冲突	假设一个堆栈成员的优先级比另一个堆栈成员高： 1 确保两个堆栈成员有相同的堆栈成员编号。需要时可以使用全局配置命令**switch***current-stack-member-number***renumber***new-stack-member-number*。 2 同时重启两个堆栈成员。	有更高优先级的堆栈成员会保持自己的堆栈成员编号。另一个堆栈成员会使用新的堆栈成员编号。
添加堆栈成员	1 关闭新交换机。 2 通过 StackWise-160 端口把新交换机连接到已开机的交换机堆栈。 3 开启新交换机。	活跃交换机被保留。新交换机会被添加到交换机堆栈中。
活跃交换机故障	移除（或关闭）活跃交换机。	其余堆栈成员中的一个会成为新的堆

		栈 master。堆栈中的所有其他成员保持不变且不会重启。
添加超过九个堆栈成员	1 通过 StackWise-160 端口连接了 10 台设备。 2 打开所有设备。	两台设备会成为活跃交换机。一台活跃交换机有九个堆栈成员。另一台活跃交换机保持为单独的设备。 使用设备上的 Mode 按键以及端口 LED 灯来辨别哪台设备是活跃交换机以及每台活跃交换机有哪些所属设备。

示例：启用持续 MAC 地址特性

此示例展示了如何配置持续 MAC 地址特性使用 7 分钟的时延，并验证了配置。

```
Device(config)# stack-mac persistent timer 7
WARNING: The stack continues to use the base MAC of the old Master
WARNING: as the stack MAC after a master switchover until the MAC
WARNING: persistency timer expires. During this time the Network
WARNING: Administrators must make sure that the old stack-mac does
WARNING: not appear elsewhere in this network domain. If it does,
WARNING: user traffic may be blackholed.
Device(config)# end
Device# show switch
Switch/Stack Mac Address : 0016.4727.a900
```

```
Mac persistency wait time: 7 mins
H/W Current
Switch# Role Mac Address Priority Version State
----------------------------------------------------------
*1 Active 0016.4727.a900 1 P2B Ready
```

示例：规划交换机堆栈的新成员

此示例展示了如何为交换机堆栈规划一个堆栈成员编号为 2 的交换机。命令 **show running-config** 的输出显示了与规划交换机相关的接口。

```
Device(config)# switch 2 provision switch_PID
Device(config)# end
Device# show running-config | include switch 2
!
interface GigabitEthernet2/0/1
!
interface GigabitEthernet2/0/2
!
interface GigabitEthernet2/0/3
<输出已删节>
```

示例：show switch stack-ports summary 命令输出

只有堆栈成员 2 的端口 1 被禁用。

```
Device# show switch stack-ports summary
Device#/ Stack Neighbor Cable Link Link Sync # In
Port# Port Length OK Active OK Changes Loopback
Status To LinkOK
-------- ------ -------- -------- ---- ------ ---- --------- --------
1/1 OK 3 50 cm Yes Yes Yes 1 No
1/2 Down None 3 m Yes No Yes 1 No
2/1 Down None 3 m Yes No Yes 1 No
2/2 OK 3 50 cm Yes Yes Yes 1 No
3/1 OK 2 50 cm Yes Yes Yes 1 No
3/2 OK 1 50 cm Yes Yes Yes 1 No
```

表 172：show switch stack-ports summary 命令输出

字段	描述
Switch#/Port#	成员编号及其堆栈端口编号。
Stack Port Status	堆栈端口状态： • Absent——堆栈端口上未检测到连线。 • Down——检测到连线，但是无启用的直连邻居或堆栈端口被禁用。 • OK——检测到连线，且直连邻居启用。

Neighbor	堆栈连线另一端的活跃成员交换机编号。
Cable Length	合法长度是 50cm、1m 或 3m。 如果交换机无法检测到线缆长度，该字段值为 *nocable*。线缆可能未连接或链路不可靠。
Link OK	堆栈线缆是否连接且正常工作。线缆另一端可能连接了邻居。 是邻居交换机的堆栈端口。 • No——没有堆栈线缆连接到此端口或堆栈线缆不可用。 • Yes——有可用的堆栈线缆连接到此端口。
Link Active	堆栈线缆另一端是否连接了邻居。 • No——在另一端未检测邻居。端口不能通过此链路发送流量。 • Yes——在另一端检测到了邻居。端口可以通过此链路发送流量。
Sync OK	链路伙伴是否给堆栈端口发送了合法的协议消息。 • No——链路伙伴没有给堆栈端口发送合法的协议消息。 • Yes——链路伙伴给端口发送了合法的协议消息。
# Changes to LinkOK	链路的相对稳定性。 如果短时间内发生了大量改变，可能出现了链路抖动。
In Loopback	堆栈线缆是否连接到了成员的堆栈端口上。 • No——成员上至少有一个堆栈端口有连接的堆栈线缆。 • Yes——成员上没有堆栈端口连接了堆栈线缆。

示例：软件环回

在有三个成员的堆栈中，使用堆栈线缆连接所有成员：

```
Device# show switch stack-ports summary
Device#
Sw#/Port# Port Neighbor Cable Link Link Sync #Changes In
Status Length OK Active OK To LinkOK Loopback
-------- ------ -------- -------- ---- ------ ---- --------- --------
1/1 OK 3 50 cm Yes Yes Yes 1 No
1/2 OK 2 3 m Yes Yes Yes 1 No
2/1 OK 1 3 m Yes Yes Yes 1 No
2/2 OK 3 50 cm Yes Yes Yes 1 No
3/1 OK 2 50 cm Yes Yes Yes 1 No
```

```
3/2 OK 1 50 cm Yes Yes Yes 1 No
```

如果断开了交换机 1 上端口 1 的堆栈连线，会出现以下消息：

```
01:09:55: %STACKMGR-4-STACK_LINK_CHANGE: Stack Port 2 Switch 3 has changed to state DOWN
01:09:56: %STACKMGR-4-STACK_LINK_CHANGE: Stack Port 1 Switch 1 has changed to state DOWN
Device# show switch stack-ports summary
Device#
Sw#/Port# Port    Status   Neighbor Cable    Length   LinkOK LinkActiveSyncOK#ChangesTo
LinkOK InLoopback
-------- ------ -------- -------- ---- ------ ---- --------- --------
1/1 Absent None No cable No No No 1 No
1/2 OK 2 3 m Yes Yes Yes 1 No
2/1 OK 1 3 m Yes Yes Yes 1 No
2/2 OK 3 50 cm Yes Yes Yes 1 No
3/1 OK 2 50 cm Yes Yes Yes 1 No
3/2 Down None 50 cm No No No 1 No
```

如果断开了交换机 1 端口 2 的堆栈连线，堆栈会被分割。

交换机 2 和交换机 3 现在在堆栈线缆相连的两个成员的堆栈中：

```
Device# show sw stack-ports summary
Device#
Sw#/Port# Port Neighbor Cable Link Link Sync #Changes In
Status Length OK Active OK To LinkOK Loopback
-------- ------ -------- -------- ---- ------ ---- --------- --------
2/1 Down None 3 m No No No 1 No
2/2 OK 3 50 cm Yes Yes Yes 1 No
3/1 OK 2 50 cm Yes Yes Yes 1 No
   3/2 Down None 50 cm No No No 1 No
```

交换机 1 是单独交换机：

```
Device# show switch stack-ports summary
Device#
Sw#/Port# Port Neighbor Cable Link Link Sync #Changes In
Status Length OK Active OK To LinkOK Loopback
-------- ------ -------- -------- ---- ------ ---- --------- --------
1/1 Absent None No cable No No No 1 Yes
1/2 Absent None No cable No No No 1 Yes
```

示例：有连接堆栈线缆的软件环回

- 在交换机 1 的端口 1 上，端口状态是 *Down,*，连接了线缆。
  在交换机 1 的端口 2 上，端口状态是 *Absent*，没有连接线缆。

```
Device# show switch stack-ports summary
Device#
Sw#/Port# Port Neighbor Cable Link Link Sync #Changes In
Status Length OK Active OK To LinkOK Loopback
-------- ------ -------- -------- ---- ------ ---- --------- --------
```

```
1/1 Down None 50 Cm No No No 1 No
1/2 Absent None No cable No No No 1 No
```

- 在物理环回状态中，线缆连接了一台交换机上的两个堆栈端口。可以使用此配置来测试
  - 交换机的线缆正常工作
  - 连线的堆栈端口正常工作
  - `Device# show switch stack-ports summary`
  - `Device#`
  - `Sw#/Port# Port Neighbor Cable Link Link Sync #Changes In`
  - `Status Length OK Active OK To LinkOK Loopback`
  - `-------- ------ -------- -------- ---- ------ ---- --------- --------`
  - `2/1 OK 2 50 cm Yes Yes Yes 1 No`
  - `2/2 OK 2 50 cm Yes Yes Yes 1 No`

端口状态显示

  - 交换机 2 是单独的交换机。
  - 端口可以收发流量。

示例：没有连接堆栈线缆的软件环回

```
Device# show switch stack-ports summary
Device#
Sw#/Port# Port Neighbor Cable Link Link Sync #Changes In
Status Length OK Active OK To LinkOK Loopback
-------- ------ -------- -------- ---- ------ ---- --------- --------
1/1 Absent None No cable No No No 1 Yes
1/2 Absent None No cable No No No 1 Yes
```

示例：查找断开的堆栈线缆

堆栈线缆连接了所有堆栈成员。交换机 1 的端口 2 连接了交换机 2 的端口 1。
成员端口状态如下：

```
Device# show switch stack-ports summary
Device#
Sw#/Port# Port Neighbor Cable Link Link Sync #Changes In
Status Length OK Active OK To LinkOK Loopback
-------- ------ -------- -------- ---- ------ ---- --------- --------
1/1 OK 2 50 cm Yes Yes Yes 0 No
1/2 OK 2 50 cm Yes Yes Yes 0 No
2/1 OK 1 50 cm Yes Yes Yes 0 No
2/2 OK 1 50 cm Yes Yes Yes 0 No
```

如果断开了交换机 1 端口 2 的连线，会出现以下消息：

```
%STACKMGR-4-STACK_LINK_CHANGE: Stack Port 1 Switch 2 has changed to state DOWN
%STACKMGR-4-STACK_LINK_CHANGE: Stack Port 2 Switch 1 has changed to state DOWN
```

现在的端口状态如下：

```
Device# show switch stack-ports summary
Device#
Sw#/Port# Port Neighbor Cable Link Link Sync #Changes In
Status Length OK Active OK To LinkOK Loopback
-------- ------ -------- -------- ---- ------ ---- --------- --------
1/1 OK 2 50 cm Yes Yes Yes 1 No
1/2 Absent None No cable No No No 2 No
2/1 Down None 50 cm No No No 2 No
2/2 OK 1 50 cm Yes Yes Yes 1 No
```

线缆只有一端连接到了堆栈端口，即交换机 2 的端口 1。

- 交换机 1 端口 2 的 *Stack Port Status* 值是 *Absent*，交换机 2 端口 1 的值是 *Down*。
- *Cable Length* 值是 *No cable*.。

诊断问题：

- 验证交换机 1 端口 2 的线缆连接。
- 交换机 1 的端口 2 在以下状态时有端口或线缆问题
  - *In Loopback* 值是 *Yes*。

或

  - *Link OK*、*Link Active* 或 *Sync OK* 值是 *No*。

示例：修复堆栈端口的连接不良问题

堆栈线缆连接了所有成员。交换机 1 的端口 2 连接了交换机 2 的端口 1。

端口状态如下：

```
Device# show switch stack-ports summary
Device#
Sw#/Port# Port Neighbor Cable Link Link Sync #Changes In
Status Length OK Active OK To LinkOK Loopback
-------- ------ -------- -------- ---- ------ ---- --------- --------
1/1 OK 2 50 cm Yes Yes Yes 1 No
1/2 Down None 50 cm No No No 2 No
2/1 Down None 50 cm No No No 2 No
2/2 OK 1 50 cm Yes Yes Yes 1 No
```

诊断问题：

- 堆栈端口状态值是 *Down*。
- Link OK、Link Active 以及 Sync OK 值是 *No*。
- Cable Length 值是 *50cm*。交换机检测并正确识别了线缆。

交换机 1 端口 2 和交换机 2 端口 1 之间的连接至少有一个连接头引脚不可靠。

其他参考资料

相关文档

相关主题	文档标题

交换机堆栈的连线及供电	Inspur 6650 交换机硬件安装指南 http://www.icntnetworks.com
SGACL 高可用性	*Inspur TrustSec 交换机配置指南*的“Inspur TrustSec SGACL 高可用性”模块。

错误信息解释

描述	**链接**
为帮助管理员搜索并解决该版本中的系统错误信息，管理员可使用错误信息解释工具。	http://www.icntnetworks.com

标准以及 RFC

标准/RFC	**标题**
无	-

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	http://www.icntnetworks.com

配置 Inspur NSF 与 SSO

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件

版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

NSF 与 SSO 的前提条件

以下是配置 NSF 与 SSO 的前提与注意事项：

- 使用路由协议要求有 IP Services 许可证等级。EIGRP-stub 以及被路由访问的 OSPF 在 IP Base 许可证等级中支持。
- NSF 的 BGP 支持要求邻居网络设备是 NSF 感知的，即设备必须有平滑重启的能力，并能够在会话建立期间通过 OPEN 消息通告该能力。如果一台 NSF 支持的路由器发现一个特定的 BGP 邻居没有平滑重启的能力，它就不会与该邻居建立 NSF 支持的会话。所有其他有平滑重启能力的邻居继续与这台 NSF 支持的网络设备使用 NSF 支持的会话。
- NSF 的 OSPF 支持要求所有邻居网络设备是 NSF 感知的。如果一台 NSF 支持的路由器发现特定网段上有一个非 NSF 感知的邻居，它会在该网段上禁用 NSF 能力。其他完全由 NSF 支持或 NSF 感知路由器组成的网段会继续提供 NSF 能力。

NSF 与 SSO 的限制条件

以下是配置 NSF 与 SSO 的限制：

- NSF 功能仅支持 IPv4 路由协议。NSF 功能不支持 IPv6 路由协议。
- NSF 不支持 IP 组播路由，因为它不是 SSO 可知的。
- 如果 INOS 软件运行在 LAN Base 模式中，则不支持 NSF。
- 要进行 NSF 的操作，必须在设备上配置 SSO。
- NSF 与 SSO 仅支持 IP 第 4 版的流量及协议。NSF 与 SSO 不支持 IPv6 流量。
- 所有三层邻居设备必须是 NSH Helper 或支持 NSF 平滑重启能力。
- 对于 IETF，所有邻居设备必须运行 NSF 感知的软件镜像。

关于 NSF 与 SSO 的信息

NSF 与 SSO 概述

交换机支持故障抵御能力，允许在活跃交换机不可用时由备用交换机进行代替。Inspur 不间断转发（nonstop forwarding，NSF）与状态化切换（stateful switchover，SSO）一同工作，能最小化网络不可用的时间。

NSF 提供了以下优势：

- 提升网络可用性——NSF 继续转发网络流量以及应用状态信息，在切换之后可以维持用户会话信息。
- 整体网络稳定性——当网络中的路由器故障并丢失路由表时，可能会造成路由抖动。通

过减少路由抖动的次数，可以提升网络稳定性。

- 邻接路由器不会检测到链路抖动——因为接口在切换期间保持启用，邻接路由器不会检测到链路抖动（链路不会断开并再次启用）。
- 防止路由抖动——因为 SSO 在切换期间持续转发网络流量，路由抖动得以避免。
- 维护在切换之前建立的用户会话。

Keepalive 消息在活跃交换机与备用交换机之间收发。

- 如果备用交换机不回应，会选举新的备用交换机。
- 如果活跃交换机不回应，备用交换机会成为活跃交换机。

此外，所有堆栈成员都收发 hello 消息。

- 如果堆栈成员不回应，该成员被从堆栈移除。
- 如果备用交换机不回应，会选举新的备用交换机。
- 如果活跃交换机不回应，备用交换机会成为活跃交换机。

SSO 操作

当备用交换机运行在 SSO 模式时，启动到完全初始化状态的备用交换机会同步活跃交换机的持久化配置与运行配置。之后它会维护以下列出的协议的状态，且所有支持状态化切换特性的硬件和软件状态变化都会被同步。由此，交换机在冗余活跃交换机的配置中能支持最少的二层会话中断。

如果活跃交换机故障，备用交换机会成为活跃交换机。新的活跃交换机会使用现有的二层交换信息继续转发流量。三层转发会被延迟，直到路由表被重新填充到新的活跃交换机中。

注释： 如果 INOS 软件运行 LAN Base 许可证等级，则不支持 SSO。

活跃交换机和备用交换机会保持以下特性的状态：

- 802.3
- 802.3u
- 802.3x (流量控制)
- 802.3ab (GE)
- 802.3z (吉比特以太网以及 CWDM)
- 802.3ad (LACP)
- 802.1p (二层 QoS)
- 802.1q
- 802.1X (认证)
- 802.1D (生成树协议)
- 802.3af (线上供电)
- PAgP
- VTP
- 动态 ARP 监测
- DHCP 侦听
- IP 源防护
- IGMP 侦听(第 1 版和第 2 版)
- DTP (802.1q 和 ISL)
- MST
- PVST+
- Rapid-PVST

- PortFast/UplinkFast/BackboneFast
- BPDU 防护及过滤
- 语音 VLAN
- 端口安全
- 单播 MAC 过滤
- ACL (VACLS, PACLS, RACLS)
- QOS (DBL)
- 组播风暴控制/广播风暴控制

SSO 与下列特性兼容。然而，这些特性的协议数据库不会在备用交换机与活跃交换机之间同步：

- 802.1Q 隧道与二层协议隧道 (Layer 2 Protocol Tunneling，L2PT)
- 小巨人（Baby Giants）帧
- 巨型（Jumbo）帧
- CDP
- 泛洪阻塞
- UDLD
- SPAN/RSPAN
- NetFlow

如果启用了 SSO，交换机上的所有三层协议都在备用交换机上学习。

NSF 操作

Inspur INOS 不间断转发（NSF）总是与状态化切换（SSO）一同工作，为三层流量提供冗余性。BGP、OSPF 以及 EIGRP 路由协议都支持 NSF，且 NSF 也被 Inspur 快速转发（Inspur Express Forwarding，CEF）支持。路由协议被增加了 NSF 能力及可知性，这意味着运行这些协议的路由器能够检测到切换操作并采取行为，以继续转发网络流量并通过对端设备恢复路由信息。在切换期间，路由协议会重建路由信息库（Routing Information Base，RIB），此时每种协议依赖 CEF 继续转发数据包。在路由协议收敛之后，CEF 会更新 FIB 表并移除过时的路由条目，随后再使用新的 FIB 信息更新硬件。

如果为 BGP、OSPF 或 EIGRP 路由协议配置了活跃交换机（使用 **graceful-restart** 命令），在活跃交换机选举过程中路由更新会自动发送。

IP Services 许可证等级的交换机支持 BGP、OSPF 以及 EIGRP 协议的 NSF 感知及 NSF 支持，IP Base 许可证等级支持 EIGRP-stub 的 NSF 感知。

NSF 有两个主要组成部分：

- NSF 感知

如果运行兼容 NSF 的软件，则网络设备是 NSF 感知的。如果邻居路由器设备检测到了一台 NSF 路由器在交换机选举发生时仍能够转发数据包，这种能力就被称为 NSF 感知。Inspur 增强的三层路由协议（BGP、OSPF 以及 EIGRP）被设计能够避免路由抖动，使得 CEF 路由表不会超时，NSF 路由器不会丢弃路由。NSF 感知的路由器能帮助发送路由协议信息给邻居 NSF 路由器。默认为 EIGRP-stub、EIGRP 以及 OSPF 协议启用 NSF 感知。默认为 BGP 禁用 NSF 感知。

- NSF 支持

如果配置设备支持 NSF，则设备是 NSF 支持的。该设备会通过 NSF 感知或 NSF 支持的邻居重建路由信息。NSF 与 SSO 一起工作，能最小化三层网络不可用的时间，在活跃交换机选举过

程中能继续转发 IP 数据包。三层路由协议（BGP、OSPFv2 以及 EIGRP）的重新收敛对用户是透明的，且自动在后台发生。路由协议通过邻居设备恢复路由信息，并重建 Inspur 快速转发（CEF）表。

注释： NSF 不支持 IPv6，仅支持 IPv4 单播。

Inspur 快速转发

Inspur INOS 不间断转发（NSF）的一个关键元素是数据包转发。在 Inspur 网络设备上，数据包转发由 Inspur 快速转发（CEF）进行处理。CEF 会维护 FIB，并在切换期间使用当前的 FIB 信息继续转发数据包。此特性能减少切换期间的流量中断情况。

在正常的 NSF 操作中，活跃主用交换机上的 CEF 把自己当前的 FIB 以及邻接数据库跟备用交换机上的 FIB 以及邻接数据库进行同步。切换发生时，备用交换机一开始时的 FIB 和邻接数据库就是当前活跃交换机上的镜像。活跃交换机上的 CEF 会把变化信息发送给备用交换机，而备用交换机上的 CEF 由此保持转发引擎与活跃交换机同步。只要接口和数据通路可用，切换期间转发引擎就能继续进行转发。

随着路由协议开始重新按照前缀一个接一个的填充 RIB，CEF 会收到一个接一个的前缀更新，并以此来更新 FIB 以及邻接数据库。现有条目以及新条目会收到新的版本（“阶段”）号，表示它们已经被刷新。转发引擎的转发信息在收敛期间进行更新。交换机会在 RIB 收敛时发出信号。系统会移除阶段号比当前切换阶段老的所有 FIB 以及邻接条目。FIB 现在就代表了最新的路由协议转发信息。

BGP 的操作

当一台 NSF 支持的路由器开始与 BGP 对端的会话时，它会给对端发送一个 OPEN 消息。消息中包含一个声明，表示 NSF 支持的设备有“平滑”重启的能力。平滑重启是 BGP 路由对端在切换期间为避免路由抖动使用的一种机制。如果 BGP 对端收到了此性能的声明，它就知道发送消息的设备是 NSF 支持的。在会话建立时，NSF 支持的路由器及其 BGP 对端都需要在 OPEN 消息中交换平滑重启能力的声明。如果两个对端都不交换平滑重启的能力声明，则此会话不支持平滑重启。

如果活跃交换机切换期间 BGP 会话丢失，NSF 感知的 BGP 对端就会把与 NSF 支持路由器相关的所有路由标记为过时；然而，在设定的一端时间内，它会继续使用这些路由执行转发决策。此功能避免了新活跃交换机等待路由信息收敛期间 BGP 对端的数据包丢失。

在活跃交换机切换发生之后，NSF 支持的路由器会重新与 BGP 对端建立会话。在新会话建立期间，它会发送一个新的平滑重启消息，表示 NSF 支持的路由器已经重新启用了。

此时，两个 BGP 对端之间会交换路由信息。在交换完成之后，NSF 支持的设备使用路由信息更新 RIB，并向 FIB 中更新转发信息。NSF 感知的设备使用网络信息移除 BGP 表中过时的路由，之后 BGP 协议完全收敛。

如果一个 BGP 对端不支持平滑重启的能力，它会忽略 OPEN 消息中的平滑重启能力声明，但也与 NSF 支持的设备建立 BGP 会话。此功能允许与非 NSF 感知的对端（无 NSF 功能）进行互操作，但是与非 NSF 感知 BGP 对端的 BGP 会话不支持平滑重启。

注释： NSF 的 BGP 支持要求邻居网络设备是 NSF 感知的，即设备必须有平滑重启的能力，并能够在会话建立期间通过 OPEN 消息通告该能力。如果一台 NSF 支持的路由器发现一个特定的 BGP 邻居没有平滑重启的能力，它就不会与该邻居建立 NSF 支持的会话。所有其他有

平滑重启能力的邻居继续与这台 NSF 支持的网络设备使用 NSF 支持的会话。

OSPF 的操作

当一台 NSF 支持的 OSPF 路由器执行活跃交换机切换时，为了重新与其 OSPF 邻居同步链路状态数据库，它必须执行以下任务：

- 在不重置邻居关系的情况下重新学习网络上的可用 OSFP 邻居
- 重新获取网络的链路状态数据库内容

在活跃交换机切换后，NSF 支持的路由器会尽快给邻接的 NSF 感知设备发送 OSPF NSF 信号。邻接网络设备能够识别此信号，知道信号表示与此路由器的邻居关系不应该被重置。在 NSF 支持的路由器收到来自网络上其他路由器的信号时，它可以开始重新构建自己的邻居列表。在邻居关系重新建立之后，NSF 支持的路由器开始重新与所有 NSF 感知的邻居同步数据库。此时，路由信息在 OSFP 邻居之间交换。一旦交换完成，NSF 支持的设备会使用这些路由信息来移除过时的路由，更新 RIB，并使用新的转发信息更新 FIB。此后 OSPF 协议完全收敛。

注释： NSF 的 OSPF 支持要求所有邻居网络设备是 NSF 感知的。如果一个 NSF 支持的路由器发现特定网段上有一个非 NSF 感知的邻居，它会禁用该网段的 NSF 功能。对于其他完全由 NSF 支持或 NSF 感知路由器组成的网段，它会继续提供 NSF 功能。

EIGRP 的操作

当一台支持 NSF 的 EIGRP 路由器通过 NSF 重启命令，重新启用时，它没有邻居且拓扑表是空的。备用交换机（现在的活跃交换机）会通知该路由器什么时候启用接口，什么时候获取邻居，以及什么时候重建拓扑表和路由表。重启的路由器必须在不中断发往重启路由器数据流量的情况下完成这些任务。EIGRP 对端路由器会维护通过重启路由器学习到的路由，并在 NSF 重启过程之间继续转发流量。

为了防止邻接关系被邻居重置，重启的路由器会在 EIGRP 数据包报头中使用一个新的重启（RS）位来表明要进行重启。RS 位在 hello 包以及 NSF 重启阶段的初始 INIT 更新包中设置。如果没有看到 RS 位，在收到 INIT 更新或 hello 保持计时器过期时，邻居只能检测到邻接关系被重置。没有 RS 时，邻居不知道邻接关系重置是应该使用 NSF 还是正常的启动方式处理。当邻居通过接收的 hello 包或 INIT 包收到重启的标识时，它能够识别对端列表中的重启对端，并维持与重启路由器的邻接关系。邻居随后会给重启路由器发送自己的拓扑表，并在第一个更新包中设置 RS 位，表明自己是 NSF 感知的，并帮助重启路由器学习路由信息。邻居不会在 hello 包中设置 RS 位，除非它自己也是 NSF 重启的邻居。

注释： NSF 感知的路由器可能不会帮助 NSF 重启的邻居学习路由，因为它在进行冷启动。如果至少有一个对端路由器是 NSF 感知的，重启的路由器就能接收更新并构建数据库。重启路由器必须在此后查询自己是否收敛，以通知路由信息库（RIB）。每个 NSF 感知的路由器都要求在最后一个更新包中发送表结束（end of table，EOT）标记，表示这是表内容的末尾。重启路由器在收到 EOT 标记时知道自己已经收敛。重启路由器可以在此后开始发送更新。

一个 NSF 感知的对端在收到来自重启路由器的 EOT 标识时就知道重启路由器已经收敛了。对端随后会扫描自己的拓扑表，查询以重启的邻居为源的路由。对端会对比路由的时间戳与重启事件的时间戳，以确定路由是否仍然有效。对端随后会进入 active 状态，查询经过重启路由器的不再有效路由的替代路径。

当重启路由器收到了来自邻居的所有 EOT 标识，或者当 NSF 收敛计时器过期时，EIGRP 会通

知 RIB 已经收敛。EIGRP 会等待 RIB 的收敛信号，并在此后把自己的拓扑表泛洪给所有等待的 NSF 感知对端。

如何配置 NSF 与 SSO

配置 SSO

要使用 NSF 处理任何支持的协议，必须配置 SSO。

总步骤

1. **redundancy**
2. **mode sso**
3. **end**
4. **show running-config**
5. **show redundancy states**

具体步骤

	命令或操作	目的
步骤 1	**redundancy** **示例：** `Device(config)# redundancy`	进入冗余配置模式。
步骤 2	**mode sso** **示例：** `Device(config-red)# mode sso`	配置 SSO。输入此命令后，备用交换机会重载并开始在 SSO 模式中工作。
步骤 3	**end** **示例：** `Device(config-red)# end`	返回 EXEC 模式。
步骤 4	**show running-config** **示例：** `Device# show running-config`	验证 SSO 已被启用。
步骤 5	**show redundancy states** **示例：** `Device# show redundancy states`	显示运行的冗余模式。

配置 SSO 示例

此示例展示了如何配置系统使用 SSO 并显示冗余状态：

```
Device(config)# redundancy
Device(config)# mode sso
Device(config)# end
Device# show redundancy states
my state = 13 -ACTIVE
peer state = 8 -STANDBY HOT
```

```
Mode = Duplex
Unit = Primary
Unit ID = 5
Redundancy Mode (Operational) = sso
Redundancy Mode (Configured) = sso
Split Mode = Disabled
Manual Swact = Enabled
Communications = Up
client count = 29
client_notification_TMR = 30000 milliseconds
keep_alive TMR = 9000 milliseconds
keep_alive count = 1
keep_alive threshold = 18
RF debug mask = 0x0
```

验证 CEF NSF

要验证 CEF NSF，使用 **show cef state** 特权 EXEC 命令。

```
Device# show cef state
CEF Status:
RP instance
common CEF enabled
IPv4 CEF Status:
CEF enabled/running
dCEF enabled/running
CEF switching enabled/running
universal per-destination load sharing algorithm, id DEA83012
IPv6 CEF Status:
CEF disabled/not running
dCEF disabled/not running
universal per-destination load sharing algorithm, id DEA83012
RRP state:
I am standby RRP: no
RF Peer Presence: yes
RF PeerComm reached: yes
RF Progression blocked: never
Redundancy mode: rpr(1)
CEF NSF sync: disabled/not running
CEF ISSU Status:
FIBHWIDB broker
No slots are ISSU capable.
FIBIDB broker
No slots are ISSU capable.
FIBHWIDB Subblock broker
```

```
No slots are ISSU capable.
FIBIDB Subblock broker
No slots are ISSU capable.
Adjacency update
No slots are ISSU capable.
IPv4 table broker
No slots are ISSU capable.
CEF push
No slots are ISSU capable.
```

配置 BGP NSF

必须在参与 BGP NSF 的所有对端设备上配置 BGP 平滑重启。

总步骤

1. configure terminal

2. router bgp as-number

3. bgp graceful-restart

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device(config)# configure terminal`	进入全局配置模式。
步骤 2	**router bgp as-number** **示例：** `Device(config)# router bgp 300`	启用 BGP 路由进程，进入交换机配置模式。
步骤 3	**bgp graceful-restart** **示例：** `Device(config)# bgp graceful-restart`	启用 BGP 平滑重启功能，开启 BGP NSF。如果在 BGP 会话建立之后输入此命令，为了使邻居之间能交换此功能的信息，必须重启会话。在重启交换机以及所有对端上使用此命令。

验证 BGP NSF

要验证 NSF，必须检查是否在启用 SSO 的网络设备以及邻居设备上配置了 BGP 平滑重启。按照以下步骤进行验证：

步骤 1 验证启用 SSO 的交换机的 BGP 配置中有“bgp graceful-restart”，输入命令：

示例：

```
Device# show running-config
.
.
.
router bgp 120
.
```

```
.
.
bgp graceful-restart
neighbor 192.0.2.0 remote-as 300
.
.
.
```

步骤 2 在每个 BGP 邻居上重复步骤 1。

步骤 3 在 SSO 设备以及邻居设备上，验证平滑重启功能显示为已通告和已接收，并确定地址族有平滑重启的能力。如果没有列出地址族，BGP NSF 也不会执行：

示例：

```
Device# show ip bgp neighbors
BGP neighbor is 192.0.2.3, remote AS 1, internal link
BGP version 4, remote router ID 192.0.2.4
BGP state = Established, up for 00:02:38
Last read 00:00:38, last write 00:00:35, hold time is 180, keepalive interval is
60seconds
Neighbor capabilities:
Route refresh: advertised and received(new)
Address family IPv4 Unicast: advertised and received
Message statistics:
InQ depth is 0
OutQ depth is 0
Sent Rcvd
Opens: 1 1
Notifications: 0 0
Updates: 0 0
Keepalives: 4 4
Route Refresh: 0 0
Total: 5 5
Default minimum time between advertisement runs is 0 seconds
.....................................................
(其余输出被删节)
```

配置 OSPF NSF

参与 OSPF NSF 的所有设备必须是 OSPF NSF 感知的，在设备上安装 NSF 软件时这会自动设置。

总步骤

1. configure terminal

2. router ospf *processID*

3. nsf

具体步骤

	命令或操作	目的

步骤 1	**configure terminal** **示例：** `Device(config)#` **configure terminal**	进入全局配置模式。
步骤 2	**router ospf** *processID* **示例：** `Device(config)#` **router ospf** ***processID***	启用 OSPF 路由进程，进入路由器配置模式。
步骤 3	**nsf** **示例：** `Device(config)#` **nsf**	为 OSPF 启用 NSF 操作。

验证 OSPF NSF

步骤 1 输入 **show running-config** 命令验证 SSO 设备的 OSPF 配置中有“nsf”：

示例：

```
Device(config)#show running-config
route ospf 120
log-adjacency-changes
nsf
network 192.0.2.0 192.0.2.255 area 0
network 192.0.2.1 192.0.2.255 area 1
network 192.0.2.2 192.0.2.255 area 2
.
.
.
```

步骤 2 输入 **show ip ospf** 命令验证设备启用了 NSF：

示例：

```
Device show ip ospf
Routing Process "ospf 1" with ID 192.0.2.1
Start time: 00:02:07.532, Time elapsed: 00:39:05.052
Supports only single TOS(TOS0) routes
Supports opaque LSA
Supports Link-local Signaling (LLS)
transit capable is 0
External flood list length 0
IETF Non-Stop Forwarding enabled
restart-interval limit: 120 sec
IETF NSF helper support enabled
Inspur NSF helper support
enabled
Reference bandwidth unit is 100 mbps
Area BACKBONE(0)
Number of interfaces in this area is 3 (1 loopback)
Area has no authentication
SPF algorithm last executed 00:08:53.760 ago
```

```
SPF algorithm executed 2 times
Area ranges are
Number of LSA 3. Checksum Sum 0x025BE0
Number of opaque link LSA 0. Checksum Sum 0x000000
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0
```

配置 EIGRP NSF

总步骤

1. configure terminal

2. router eigrp *as-number*

3. nsf

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device` **configure terminal**	进入全局配置模式。
步骤 2	**router eigrp** *as-number* **示例：** `Device(config)#` **router eigrp** *as-number*	启用 EIGRP 路由进程，进入路由器配置模式。
步骤 3	**nsf** **示例：** `Device(config-router)#` **nsf**	启用 EIGRP NSF。 在“重启”交换机以及所有对端上使用此命令。

验证 EIGRP NSF

步骤 1 输入 **showrunning-config** 命令验证 SSO 设备的 EIGRP 配置中有“nsf”：

示例：

```
Device show running-config
..
.
router eigrp 100
auto-summary
nsf
..
```

步骤 2 输入 **show ip protocols** 命令验证设备启用了 NSF：

示例：

```
Device show ip protocols
*** IP Routing is NSF aware ***
Routing Protocol is "ospf 1"
```

```
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Router ID 192.0.2.3
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
Maximum path: 1
Routing for Networks:
Routing on Interfaces Configured Explicitly (Area 0):
Loopback0
GigabitEthernet5/3
TenGigabitEthernet3/1
Routing Information Sources:
Gateway Distance Last Update
192.0.2.1 110 00:01:02
Distance: (default is 110)
Routing Protocol is "bgp 601"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
IGP synchronization is disabled
Automatic route summarization is disabled
Neighbor(s):
Address FiltIn FiltOut DistIn DistOut Weight RouteMap
192.0.2.0
Maximum path: 1
Routing Information Sources:
Gateway Distance Last Update
192.0.2.0 20 00:01:03
Distance: external 20 internal 200 local 200
```

第 15 部分 系统管理

管理交换机

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于管理设备的信息

系统时间及日期管理

可以使用自动配置方式（RTC 以及 NTP）或者手动配置方式来管理设备上的系统时间及日期。
注释： 有关本节使用命令的完整语法以及使用信息，参见 *icntnetworks.com* 上的 *Inspur INOS 配置基本命令参考*。

系统时钟

时间服务的基础是系统时钟。时钟从系统启动就开始运行，跟踪记录着日期和时间。
系统时钟可以通过以下来源设置：

- NTP
- 手动配置

系统时钟可以给以下服务提供时间：

- 用户 **show** 命令
- 日志以及调试信息

系统时钟内部基于世界协调时间（Coordinated Universal Time，UTC）（也称为格林威治标准时间，Greenwich Mean Time，GMT）来记录时间。可以配置本地时区以及夏令时的信息，以正确显示本地时区的时间。

系统时钟会记录时间是否*权威*（即是否由被认为权威的时间源设置）。如果不权威，该时间只用于显示信息，不会被分发其他设备。

网络时间协议

NTP 的目的是对网络中的设备进行时间同步。NTP 在用户数据报协议（User Datagram Protocol，UDP）之上运行，并通过 IP 运行。NTP 在 RFC 1305 中说明。

NTP 网络通常通过权威时间源获取时间，比如连接到时间服务器的无线电时钟或原子钟，之后 NTP 会在网络中分发时间信息。NTP 极其高效，要把两台设备的时间差同步在一毫秒以内，每分钟最多只需要一个数据包。

NTP 层

NTP 使用*层（stratum）*的概念来描述一台设备距离权威时间源有多个 NTP 跳远。第 1 层的时间服务器有直连的无线电时钟或原子钟，第 2 层的时间服务器通过 NTP 接收第 1 层时间服务器的时间，以此类推。运行 NTP 的设备会在与其通过 NTP 通信的时间源中选择层数最低的设备作为自己的时间源。这样的策略能够高效地构建自组的 NTP 通告者树。

NTP 不会从一个未同步的设备上同步时间，进而避免了同步的设备时间不准确的问题。NTP 也会比较几台设备汇报的时间，如果一台设备层数较低，但是它提供的时间与其他设备的时间明显不同，NTP 也不会从该设备同步时间。

NTP 关联

运行 NTP 的设备之间的通信（也称为关联）通常是静态配置的；每台设备都配置了应该建立关联的所有设备的 IP 地址。每对关联设备之间交换 NTP 消息使得可以进行精确的计时。然而，在 LAN 环境中，可以配置 NTP 使用 IP 广播消息。这种方式减少了配置复杂性，因为可以直接配置每台设备发送或接受广播消息。不过此时的信息流仅能是单向的。

NTP 安全

设备上记录的时间是一种关键资源，应该启用 NTP 安全特性，避免意外或被恶意设置了不正确的时间。有两种安全机制可以使用：基于访问列表的限制机制以及加密认证机制。

NTP 实现

Inspur 的 NTP 实现不支持第 1 层服务，设备不能连接无线电时钟或原子钟。建议通过 IP Internet 上的公共 NTP 服务器获取网络时间。

下图展示了一个典型的使用 NTP 的网络。设备 A 是 NTP master，设备 B、C 和 D 都配置为 NTP 服务器模式，与设备 A 进行关联。设备 E 分别是上游和下游设备 B 和设备 F 的 NTP 对端。

图 132：典型的 NTP 网络配置

Switch A	交换机 A

Localworkgroup servers	本地工作组服务器
Workstations	工作站

如果网络与 Internet 隔离，Inspur 的 NTP 实现允许设备在通过其他方式学习时间时，当作是通过 NTP 同步的来进行操作。之后其他设备可以与该设备通过 NTP 进行同步。
当有多个时间源可用时，NTP 总被认为是更权威的。NTP 时间会覆盖由其他方式设置的时间。
一些厂商的主机系统中带有 NTP 软件，UNIX 及众多衍生版本系统也有公开可用的 NTP 软件。这些软件允许主机系统进行时间同步。

NTP 第 4 版
本设备上实现了 NTP 第 4 版。NTPv4 是 NTP 第 3 版的扩展。NTPv4 支持 IPv4 以及 IPv6，并向后兼容 NTPv3。
NTPv4 提供以下功能：

- 支持 IPv6。
- 比 NTPv3 提升了安全性。NTPv4 使用基于公有密钥加密以及标准 X509 证书的安全框架。
- 能够自动计算网络的时间分发层级。使用特定的组播组，NTPv4 能够自动配置服务器层级，以通过最低的带宽开销实现最高的时间精确性。此特性使用 IPv6 站点本地组播地址。

配置 NTPv4 的更多信息参见 *Inspur INOS IPv6 配置指南 12.4T 版*中的*在 IPv6 中部署 NTPv4* 章节。

系统名称及提示符

可以在设备上配置系统名称以进行标识。默认情况下，系统名称及提示符是 Device。
如果没有配置系统提示符，系统名称的前 20 个字符被用于系统提示符，之后附带一个大于号“>”。提示符会在系统名改变时更新。
有关本节使用命令的完整语法以及使用信息，参见 *Inspur INOS 配置基本命令参考 12.4 版*以及 *Inspur INOS IP 命令参考第 2 卷：路由协议 12.4 版*。

堆栈系统名称及提示符
如果通过活跃交换机访问堆栈成员，必须使用特权 EXEC 命令 **session** *stack-member-number*。堆栈成员编号范围是 1 到 9。使用此命令时，堆栈成员编号会被附加在系统提示符之后。例如，Switch-2#是堆栈成员 2 的特权 EXEC 模式提示符，而交换机堆栈的系统提示符是 Switch。

默认系统名称及提示符配置
默认的交换机系统名称及提示符是 *Switch*。

DNS

DNS 协议控制着域名系统（Domain Name System，DNS），这是可以把主机名映射到 IP 地址的分布式数据库。在设备上配置 DNS 时，可以使用主机名替换所有 IP 命令中的 IP 地址，如 **ping**、**telnet**、**connect** 以及相关的 Telnet 支持操作。
IP 地址定义了一种层级化的命名机制，允许通过地址或域名标识设备。域名使用英文句号（.）

作为分隔符拼接在一起。例如，Inspur Systems 是一家商业机构，IP 地址通过 *com* 域名标识，所以其域名是 *icntnetworks.com*。此域名中的一台特定设备，如文件传输协议（File Transfer Protocol，FTP）系统由 *ftp.icntnetworks.com* 标识。
为了记录域名，IP 定义了域名服务器的概念，该服务器会保存名称到 IP 地址映射的缓存（或数据库）。要把域名映射到 IP 地址，必须先标识出主机名，然后指定网络上现有的名称服务器，并且要启用 DNS。

默认 DNS 设置

表 174：默认 DNS 设置

特性	默认设置
DNS 启用状态	启用。
DNS 默认域名	未配置。
DNS 服务器	未配置名称服务器地址。

登录标语

可以配置当日消息（message-of-the-day，MOTD）以及登录标语。MOTD 标语会在所有连接终端登录时显示，可以用来发送会影响所有网络用户的消息（如即将进行的系统关机）。
登录标语也会在所有连接终端上显示。它出现在 MOTD 标语之后，登录提示之前。
注释： 有关本节使用命令的完整语法以及使用信息，参见 *Inspur INOS 配置基本命令参考 12.4 版*。

默认标语配置

MOTD 以及登录标语未被配置。

MAC 地址表

MAC 地址表包含了设备用来在端口之间转发流量使用的地址信息。地址表中的所有 MAC 地址都与一个或者多个端口关联。地址表包含以下几种类型的地址：

- 动态地址——设备学习到的源 MAC 地址，不使用时会被老化。
- 静态地址——手动输入的单播地址，不会老化且设备重置时不会丢弃。

地址表中列出了目的 MAC 地址，关联的 VLAN ID，与地址关联的端口号以及地址类型（静态或动态）。
注释：
注释： 有关本节使用命令的完整语法以及使用信息，参见此版本的命令参考手册。

MAC 地址表创建

所有端口上都支持有多个 MAC 地址，可以把设备端口连接到其他网络设备上。设备拥有动态学习地址的能力，可以学习每个端口收到数据包的源地址，并将学习到的地址以及关联的端口号添加到地址表中。在网中添加或移除设备时，设备会更新地址表，添加新的动态地址，并老化不使用的地址。
老化间隔是全局配置的。然而，设备会为每个 VLAN 维护一张地址表，且 STP 可以加速每个 VLAN 的老化间隔时间。

设备基于收到的数据包的目的地址，在任意端口组合之间发送数据包。通过使用 MAC 地址表，设备可以把数据包只转发到与目的地址关联的端口。如果目的地址在发送数据包的端口上，此数据包会被过滤而不被转发。设备总是会使用存储转发方式：在传输之前会存储完整的数据包并进行错误检查。

MAC 地址以及 VLAN

所有地址都与 VLAN 关联。一个地址可以存在于多个 VLAN 中，且在每个 VLAN 中可以有不同的目的端口。例如，单播地址数据包可以转发到 VLAN1 中的端口 1，以及 VLAN 5 中的端口 9、10 和 1 上。

每个 VLAN 都维护自己的逻辑地址表。一个 VLAN 中的已知地址在另一个 VLAN 中是未知的，除非在另一个 VLAN 中学习到或与某端口进行了静态关联。

MAC 地址以及设备堆栈

所有堆栈成员上的 MAC 地址表都是同步的。任意时刻，每个堆栈成员都有每个 VLAN 地址表的相同拷贝。地址老化后，会被从所有堆栈成员的地址表中移除。设备加入交换机堆栈时，该设备会接收到其他堆栈成员学习的每个 VLAN 的地址信息。当一个堆栈成员离开交换机堆栈时，其余堆栈成员会老化或移除通过之前的堆栈成员学习到的所有地址。

默认的 MAC 地址表设置

下表显示了 MAC 地址表的默认设置。

表 175：MAC 地址默认设置

特性	**默认设置**
老化时间	300 秒
动态地址	自动学习
静态地址	无配置

ARP 表管理

与一台设备进行通信时（如通过以太网通信），软件必须学习到该设备的 48 位 MAC 地址或者本地数据链路地址。这个通过 IP 地址学习本地数据链路地址的过程被称为*地址解析（address resolution）*。

地址解析协议（Address Resolution Protocol，ARP）会把 IP 地址与对应的介质或 MAC 地址以及 VLAN ID 进行关联。使用 IP 地址，ARP 查询关联的 MAC 地址。当找到 MAC 地址时，IP-MAC 地址关联会被存储在 ARP 缓存中，以便快速提取。随后 IP 数据报会被封装在数据链路帧中，并通过网络发送。在以太网以外的 IEEE 802 网络上，IP 数据包封装以及 ARP 请求应答通过子网访问协议（Subnetwork Access Protocol，SNAP）进行了规范。默认情况下，在 IP 接口上启用标准以太网方式的 ARP 封装（由 **arpa** 关键字表示）。

手动添加到表中的 ARP 条目不会过期，且必须被手动移除。

相关 CLI 配置过程，参见 *icntnetworks.com* 上的 Inspur INOS 12.4 版文档。

如何管理设备

手动配置时间及日期

在重启过程之后，系统时间能保持准确，然而也可以在系统重启后手动配置时间以及日期。建议仅在必要时使用手工配置的方式。如果有设备可以同步的外部时间源，则无需手动设置系统时钟。

注释： 如果手动配置了时间，必须在活跃交换机故障且另一个堆栈成员接替了活跃交换机角色之前手动重新配置。

设置系统时钟

如果网络上有能提供时间服务的外部时间源，如 NTP 服务器，则无需手动设置系统时钟。按照以下步骤设置系统时钟：

总步骤

1. enable

2. 使用以下命令之一：

- **clock set** *hh:mm:ss day month year*
- **clock set** *hh:mm:ss month day year*

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	启用特权 EXEC 模式。在提示时输入密码。
步骤 2	使用以下命令之一： • **clock set** *hh:mm:ss day month year* • **clock set** *hh:mm:ss month day year* **示例：** `Device#` **clock set 13:32:00 23 March 2013**	使用以下一种格式设置系统时钟： • *hh:mm:ss* ——按照小时、分钟一集秒的形式指定时间。指定的时间相对于配置的时区。 • *day*——指定一个月中的日期。 • *month*——指定月名称。 • *year*——指定年份（不缩写）。

配置时区

按照以下步骤手动配置时区。

总步骤

1. enable

2. configure terminal

3. clocktimezone*zone hours-offset* [*minutes-offset*]

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**clock timezone***zone hours-offset* [*minutes-offset*] **示例:** `Device(config)#` **clock timezone AST -3 30**	设置时区。 系统内部时间按照世界协调时间（UTC）记录，此命令仅用于在手动设置时间的时候展示时间。 • *zone*——输入标准时间生效时要显示的时区名。默认是 UTC。 • *hours-offset*——输入距 UTC 的偏移小时数。 • （可选）*minutes-offset*——输入距 UTC 的偏移分钟数。此选项在本地时区距 UTC 的时间是一小时的百分比时可用。
步骤 4	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 5	**show running-config** **示例:** `Device#` **show running-config**	验证条目。
步骤 6	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）将条目保存在设备启动配置文件中。

配置夏令时

要在一些地区配置夏令时（日光节约时制）在每年特定某一周的某天起止，需执行以下操作：

总步骤

1. enable

2. configure terminal

3. **clock summer-time** *zone* **date** *date month year hh:mm date month year hh:mm*[*offset*]]
4. **clock summer-time** *zone* **recurring** [*week day month hh:mm week day month hh:mm*[*offset*]]
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**clock summer-time** *zone* **date** *date month year* *hh:mm date month year hh:mm*[*offset*]] **示例:** `Device(config)#` **clock summer-time PDTdate10 March 2013 2:00 3 November 2013 2:00**	配置夏令时在每年特定的日期起止。
步骤 4	**clock summer-time** *zone* **recurring** [*week daymonth hh:mm week day month hh:mm*[*offset*]] **示例:** `Device(config)#` **clock summer-timePDT recurring 10 March 2013 2:00 3November 2013 2:00**	配置夏令时在每年特定的日期起止。所有时间都相对于本地时区。开始时间相对于标准时间。 结束时间相对于夏令时。夏令时默认被禁用。如果使用 **clock summer-time** *zone* **recurring** 且不指定参数，夏令时规则默认为美国规则。 如果开始月份在结束月份之后，系统假定用户位于南半球。 • *zone*——指定夏令时生效时显示的时区名称（如 PDT）。 • （可选）*week*——指定一个月中的一个星期（**1** 到 **4**，**first** 或 **last**）。 • （可选）*day*——指定一周中的一天（Sunday、Monday 等） • （可选）*month*——指定月份（January、February 等） • （可选）*hh:mm*——按小时和分钟指定时间。

		• （可选）*offset*——指定夏令时期间要增加的分钟数。默认值是 60 分钟。
步骤 5	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 6	**show running-config** **示例:** `Device#` **show running-config**	验证条目。
步骤 7	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）将条目保存在设备启动配置文件中。

如果所处地区的夏令时不是循环出现的，按照以下步骤进行配置（配置下一个夏令时的确切日期与时间）：

总步骤

1. enable

2. configure terminal

3. clock summer-time *zone* **date**[*month date year hh:mm month date year hh:mm*[*offset*]]or**clocksummer-time** *zone* **date** [*date month year hh:mm date month year hh:mm*[*offset*]]

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**clock summer-time** *zone* **date**[*month date yearhh:mm month date year hh:mm*[*offset*]]or**clocksummer-time** *zone* **date** [*date month year hh:mmdate month year hh:mm*[*offset*]]	配置夏令时在第一个日期开始，在第二个日期结束。 夏令时默认被禁用。 • 使用 *zone* 指定夏令时生效时显示的时区名（如 PDT）。 • （可选）使用 *week* 指定一个月中的一周（1 到 5 或 last）。 • （可选）使用 *day* 指定一周中

		的一天（Sunday、Monday 等）。 • （可选）使用 *month* 指定月份（January、February 等） • （可选）使用 *hh:mm* 按小时和分钟指定时间。 • （可选）使用 *offset* 指定夏令时期间要增加的分钟数。默认值是 60 分钟。
步骤 4	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 5	**show running-config** **示例:** `Device#` **show running-config**	验证条目。
步骤 6	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）将条目保存在设备启动配置文件中。

配置系统名称

按照以下步骤配置系统名称：

总步骤

1. enable

2. configure terminal

3. hostname*name*

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**hostname** *name* **Example:** `Device(config)#` **hostname**	配置系统名称。设置的系统名也会被用作系统提示符。 默认设置是 Device。

	remote-users	名称必须遵循 ARPANET 主机名规则，必须以字母开始，以字母或数字结束，且之间的字符只能是字母、数字或连字符。名称至多可以有 64 个字符。
步骤 4	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 5	**show running-config** **示例:** `Device#` **show running-config**	验证条目。
步骤 6	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）将条目保存在设备启动配置文件中。

设置 DNS

如果使用设备 IP 地址作为主机名，IP 地址会被使用且不进行 DNS 查询。如果配置的主机名不包含英文句号（.），英文句号以及默认域名会被附加到主机名之后，然后进行 DNS 查询，将名称映射为 IP 地址。默认域名值由全局配置命令 **ip domain-name** 设置。如果域名中有英文句号（.），Inspur INOS 软件会查询 IP 地址且不会给主机名附加默认域名。
按照以下步骤设置交换机使用 DNS：

总步骤

1. **enable**
2. **configure terminal**
3. **ip domain-name** *name*
4. **ip name-server** *server-address1* [*server-address2 ... server-address6*]
5. **ip domain-lookup** [**nsap**| **source-interface** *interface*]
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。

步骤 3	**ip domain-name** *name* **示例:** `Device(config)#` **ipdomainnameIcntnetworks.com**	指定默认域名，系统使用该域名补全不合规的主机名（没有点分十进制域名的名称）。 不要包含分隔不合规名称与域名的句点。 设备启动时，无域名配置，然而如果设备通过 BOOTP 或动态主机配置协议（DHCP）服务器配置，可以使用 BOOTP 或 DHCP 服务器设置默认域名（如果服务器配置了此信息）。
步骤 4	**ip name-server** *server-address1*[*server-address2 ... server-address6*] **示例:** `Device(config)#` **ipname-server 192.168.1.100192.168.1.200 192.168.1.300**	指定名称地址解析使用的一个或多个名称服务器。 可以指定至多六个名称服务器。每个服务器地址之间用空格分开。指定的第一个服务器是主服务器。设备会先给主服务器发送 DNS 查询。如果查询失败，会查询备用服务器。
步骤 5	**ip domain-lookup** [**nsap**| **source-interface***interface*] **示例:** `Device(config)#` **ip domain-lookup**	（可选）在设备上启用基于 DNS 的名称-地址翻译。该特性默认被启用。 如果网络设备要求与不能控制名称分配的网络设备有连通性，可以使用全局 Internet 命名机制（DNS）给设备动态分配唯一标识设备的名称。
步骤 6	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 7	**show running-config** **示例:** `Device#` **show running-config**	验证条目。
步骤 8	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）将条目保存在设备启动配置文件中。

接下来做什么？

配置当日消息登录标语

可以配置单行或多行的消息标语，有人登录设备时会在屏幕上显示。
按照以下步骤配置 MOTD 登录标语：

总步骤

1. enable

2. configure terminal

3. bannermotd*c message c*

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。

步骤 3	**banner motd***c message c*
	示例:
	`Device(config)#` **banner motd #**
	`This is a secure site. Only`
	`authorized users are allowed.`
	`For access, contact technical`
	`support.`
	`#`
		指定当日消息。
		c——输入定界字符，如井号（#），并输入回车键。定界字符标示了标语文本的开始和结束。结束定界符之后的字符会被丢弃。
		message——输入标语消息，至多 255 字符。消息中不能使用定界字符。

步骤 4	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 5	**show running-config** **示例:** `Device#` **show running-config**	验证条目。
步骤 6	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）将条目保存在设备启动配置文件中。

配置登录标语

可以配置在所有连接的终端上显示的登录标语。此标语会在 MOTD 标语之后登录提示符之前显示。

按照以下步骤配置登录标语：

总步骤

1. enable

2. configure terminal

3. banner login *c message c*

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。

步骤 3	**banner login** *c message c*
	示例
	`Device(config)#` **banner login $**
	`Access for authorized users only.`
	`Please enter your username and password.`
	`$`
		指定登录消息。
		c——输入定界字符，如井号（#），并输入回车键。定界字符标示了标语文本的开始和结束。结束定界符之后的字符会被丢弃。
		message——输入标语消息，至多 255 字符。消息中不能使用定界字符。

步骤 4	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 5	**show running-config** **示例:** `Device#` **show running-config**	验证条目。
步骤 6	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）将条目保存在设备启动配置文件中。

管理 MAC 地址表

更改地址老化时间

按照以下步骤配置动态地址表老化时间：

总步骤

1. enable

2. configure terminal

3. mac address-table aging-time [*0* | *10-1000000*] [**routed-mac** | **vlan***vlan-id*]

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**mac address-table aging-time** [*0* | *10-1000000*][**routed-mac** | **vlan***vlan-id*] **示例:** `Device(config)#` **mac address-tableaging-time 500 vlan 2**	设置 MAC 地址表中条目使用或更新之后动态条目保留的时长。 范围从 10 到 1000000 秒。默认值是 300 秒，也可以输入 0 来禁用老化。静态地址条目不会被老化也不会被从表中移除。 *vlan-id*——合法 ID 是 1 到 4094。
步骤 4	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 5	**show running-config** **示例:** `Device#` **show running-config**	验证条目。
步骤 6	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）将条目保存在设备启动配置文件中。

配置 MAC 地址更改通知陷阱

按照以下步骤配置交换机给 NMS 主机发送 MAC 地址更改通知陷阱：

总步骤

1. enable

2. configure terminal

3. snmp-server host *host-addr community-string notification-type* { **informs**| **traps** } {**version** {**1** | **2c** | **3**}}{**vrf***vrf instance name*}

4. snmp-server enable traps mac-notification change

5. mac address-table notification change

6. mac address-table notification change [**interval** *value*] [**history-size** *value*]

7. interface*interface-id*

8. snmp trap mac-notification change {**added** | **removed**}

9. end

10. show running-config

11. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**snmp-server host** *host-addr community-stringnotification-type* { **informs** | **traps** } {**version** {**1**| **2c** | **3**}} {**vrf***vrf instance name*} **示例:** `Device(config)#` **snmp-server host172.20.10.10 traps private mac-notification**	指定陷阱消息的接收者。 • *host-addr*——指定 NMS 的名称或地址。 • **traps**（默认设置）——把 SNMP 陷阱发送给主机。 • **informs**——把 SNMP 通知发送给主机。 • **version**——指定支持的 SNMP 版本。默认是版本 1，该版本不可使用通知。 • *community-string*——指定通知操作发送的字符串。虽然可以使用 **snmp-server host** 命令设置此字符串，建议在使用 **snmp-server host** 命令之前使用 **snmp-server community** 命令定义此字符串。 • *notification-type* ——使用 **mac-notification** 关键字。 • **vrf***vrfinstance name*——指定主机的 VPN 路由/转发实例。
步骤 4	**snmp-server enable traps mac-notificationchange** **示例:** `Device(config)#` **snmp-server enable trapsmac-notification change**	让设备给 NMS 发送 MAC 地址更改通知陷阱。
步骤 5	**mac address-table notification change** **示例:** `Device(config)#` **mac address-tablenotification**	启用 MAC 地址更改通知特性。

	change	
步骤 6	**mac address-table notification change** [**interval***value*] [**history-size** *value*] **示例:** `Device(config)#` **mac address-tablenotification change interval 123** `Device(config)#`**mac address-tablenotification change history-size 100**	输入陷阱间隔时间以及历史表大小。 • （可选）**interval** *value*——指定生成给 NMS 的每组通知陷阱的间隔秒数。范围从 0 到 2147483647 秒，默认是 1 秒。 • （可选）**history-size** *value*——指定 MAC 通知历史表的最大条目数量。范围从 0 到 500，默认值是 1。
步骤 7	**interface** *interface-id* **示例:** `Device(config)#` **interfacegigabitethernet1/0/2**	进入接口配置模式，指定要启用 SNMP MAC 地址通知陷阱的二层接口。
步骤 8	**snmp trap mac-notification change** {**added** | **removed**} **示例:** `Device(config-if)#` **snmp trapmac-notification change added**	在接口上启用 MAC 地址更改通知陷阱。 • 当 MAC 地址 **added**（添加）到接口时启用陷阱。 • 当 MAC 地址从接口 **removed**（移除）时启用陷阱。
步骤 9	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 10	**show running-config** **示例:** `Device#` **show running-config**	验证条目。
步骤 11	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）将条目保存在设备启动配置文件中。

配置 MAC 地址移动通知陷阱

配置 MAC 移动通知后，当一个 MAC 地址从一个端口移动到相同 VLAN 的另一个端口时会生成 SNMP 通知并发往网络管理系统。

按照以下步骤配置设备给 NMS 主机发送 MAC 地址移动通知陷阱：

总步骤

1. enable

2. configure terminal

3. snmp-server host *host-addr*{**traps** | **informs**} {**version** {**1** | **2c** | **3**}} *community-string notification-type*

4. snmp-server enable traps mac-notification move

5. mac address-table notification mac-move

6. end

7. show running-config

8. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**snmp-server host** *host-addr*{**traps** | **informs**} {**version** {**1** | **2c** | **3**}} *community-stringnotification-type* **示例:** `Device(config)#` **snmp-server host 172.20.10.10 traps private mac-notification**	指定陷阱消息的接收者。 • *host-addr*——指定 NMS 的名称或地址。 • **traps**（默认设置）——把 SNMP 陷阱发送给主机。 • **informs**——把 SNMP 通知发送给主机。 • **version**——指定支持的 SNMP 版本。默认是版本 1，该版本不可使用通知。 • *community-string*——指定通知操作发送的字符串。虽然可以使用 **snmp-server host** 命令设置此字符串，建议在使用 **snmp-server host** 命令之前使用 **snmp-server community** 命令定义此字符串。 • *notification-type* ——使用 **mac-notification** 关键字。
步骤 4	**snmp-server enable traps mac-notification move** **示例:** `Device(config)#` **snmp-server enable traps mac-notification move**	让设备给 NMS 发送 MAC 地址移动通知陷阱。
步骤 5	**mac address-table notification mac-move** **示例:** `Device(config)#` **mac address-tablenotification mac-move**	启用 MAC 地址移动通知特性。
步骤 6	**end** **示例:**	返回特权 EXEC 模式。

	`Device(config)#` **end**	
步骤 7	**show running-config** **示例:** `Device#` **show running-config**	验证条目。
步骤 8	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）将条目保存在设备启动配置文件中。

接下来做什么？

要禁用MAC地址移动通知陷阱，需使用全局配置命令**no snmp-server enable traps mac-notificationmove**。要禁用MAC地址移动通知特性，需使用全局配置命令**no macaddress-table notification mac-move**。

可以输入特权EXEC命令**show mac address-table notification mac-move**验证设置。

配置MAC门限值通知陷阱

配置 MAC 门限值通知后，当达到或超过 MAC 地址表门限值时会生成 SNMP 通知并发送给网络管理系统。

按照以下步骤配置交换机给 NMS 主机发送 MAC 地址表门限值通知陷阱：

总步骤

1. enable

2. configure terminal

3. snmp-server host *host-addr*{**traps** | **informs**} {**version** {**1** | **2c** | **3**}} *community-string notification-type*

4. snmp-server enable traps mac-notification threshold

5. mac address-table notification threshold

6. mac address-table notification threshold [**limit** *percentage*] | [**interval** *time*]

7. end

8. show running-config

9. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**snmp-server host** *host-addr*{**traps**	**informs**}{**version** {**1**	**2c**	**3**}} *community-stringnotification-type* **示例:**	指定陷阱消息的接收者。 • *host-addr*——指定 NMS 的名称或地址。 • **traps**（默认设置）——把 SNMP

	Device(config)# **snmp-server host 172.20.10.10 traps privatemac-notification**	陷阱发送给主机。 • **informs**——把 SNMP 通知发送给主机。 • **version**——指定支持的 SNMP 版本。默认是版本 1，该版本不可使用通知。 • *community-string*——指定通知操作发送的字符串。虽然可以使用 **snmp-server host** 命令设置此字符串，建议在使用 **snmp-server host** 命令之前使用 **snmp-server community** 命令定义此字符串。 • *notification-type* ——使用 **mac-notification** 关键字。
步骤 4	**snmp-server enable traps mac-notificationthreshold** **示例:** Device(config)# **snmp-server enable trapsmac-notification threshold**	让设备给 NMS 发送 MAC 门限值通知陷阱。
步骤 5	**mac address-table notification threshold** **示例:** Device(config)# **mac address-table notification threshold**	启用 MAC 地址门限值通知特性。
步骤 6	**mac address-table notification threshold** [**limit***percentage*]	[**interval** *time*] **示例:** Device(config)# **mac address-tablenotification threshold interval 123** Device(config)# **mac address-tablenotification threshold limit 78**	输入 MAC 地址门限值使用监控的门限值。 • （可选）**limit** *percentage*——指定 MAC 地址表使用的百分比，合法值范围从 1 到 100，默认是 50%。 • （可选）**interval** *time*——指定通知间隔，合法值应大于或等于 120 秒。默认值是 120 秒。
步骤 7	**end** **示例:** Device(config)# **end**	返回特权 EXEC 模式。
步骤 8	**show running-config** **示例:** Device# **show running-config**	验证条目。
步骤 9	**copy running-config startup-config** **示例:** Device# **copy running-config startup-config**	（可选）将条目保存在设备启动配置文件中。

接下来做什么？

添加或移除静态地址条目

按照以下步骤添加静态地址：

总步骤

1. enable

2. configure terminal

3. mac address-table static *mac-addr***vlan***vlan-id* **interface** *interface-id*

4. end

5. show running-config

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**mac address-table static** *mac-addr***vlan***vlan-id* **interface** *interface-id* **示例:** `Device(config)#` **mac address-tablestatic c2f3.220a.12f4 vlan 4 interfacegigabitethernet 1/0/1**	向 MAC 地址表添加静态地址。 • *mac-addr*——指定要添加到地址表的目的 MAC 单播地址。在指定 VLAN 中接收的使用该目的地址的数据包会被转发到指定的接口。 • *vlan-id*——指定接收指定 MAC 地址数据包的 VLAN。合法 VLAN ID 从 1 到 4094。 • *interface-id*——指定收到的数据包要被转发到的接口。合法的接口包括物理端口或端口通道。对于静态组播地址，可以输入多个接口 ID。对于静态单播地址，一次仅可以输入一个接口，但是可以多次输入相同 MAC 地址以及 VLAN ID 的命令。
步骤 4	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。此外，也可以按下 **Ctrl-Z** 退出全局配置模式。
步骤 5	**show running-config**	验证条目。

	示例: `Device#` **show running-config**	
步骤 6	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）将条目保存在设备启动配置文件中。

配置单播 MAC 地址过滤

按照以下步骤配置设备丢弃源或目的单播静态地址：

总步骤

1. **enable**
2. **configure terminal**
3. **mac address-table static** *mac-addr***vlan***vlan-id* **drop**
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**mac address-table static** *mac-addr***vlan***vlan-id***drop** **示例:** `Device(config)#` **mac address-tablestatic c2f3.220a.12f4 vlan 4 drop**	启用单播 MAC 地址过滤，并配置设备丢弃使用指定源或目的单播静态地址的数据包。 • *mac-addr*——指定源或目的单播 MAC 地址（48 位）。使用此 MAC 地址的数据包会被丢弃。 • *vlan-id*——指定接收带有指定 MAC 地址数据包的 VLAN。合法 VLAN ID 从 1 到 4094。
步骤 4	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 5	**show running-config** **示例:** `Device#` **show running-config**	验证条目。
步骤 6	**copy running-config startup-config** **示例:**	（可选）将条目保存在设备启动配置文件中。

	Device# **copy running-config startup-config**	

监控并维护设备管理

命令	目的
clear mac address-table dynamic	移除所有动态条目。
clear mac address-table dynamic address*mac-address*	移除指定的 MAC 地址。
clear mac address-table dynamic interface*interface-id*	移除指定物理端口或端口通道上的所有地址。
clear mac address-table dynamic vlan*vlan-id*	移除指定 VLAN 上的所有地址。
show clock [*detail*]	显示时间及日期配置。
show ipigmp snooping groups	显示所有 VLAN 或指定 VLAN 的二层组播条目。
show mac address-table address *mac-address*	显示指定 MAC 地址的 MAC 地址表信息。
show mac address-table aging-time	显示所有 VLAN 或指定 VLAN 中的老化时间。
show mac address-table count	显示所有 VLAN 或指定 VLAN 中的地址数量。
show mac address-table dynamic	仅显示动态 MAC 地址表条目。
show mac address-table interface *interface-name*	显示指定接口的 MAC 地址表信息。
show mac address-table move update	显示 MAC 地址表移动更新信息。
show mac address-table multicast	显示组播 MAC 地址列表。
show mac address-table notification {change |mac-move | threshold}	显示 MAC 通知参数及历史表。
show mac address-table secure	显示安全 MAC 地址。
show mac address-table static	仅显示静态 MAC 地址表条目。
show mac address-table vlan*vlan-id*	显示指定 VLAN 的 MAC 地址表信息。

设备管理的配置示例

示例：设置系统时钟

此示例展示了如何手动设置系统时钟：

```
Device# clock set 13:32:00 23 July 2013
```

示例：配置夏令时

此示例展示了如何指定从 3 月 10 日 02:00 开始并在 11 月 3 日 02:00 结束的夏令时：

```
Device(config)# clock summer-time PDT recurring PST date
10 March 2013 2:00 3 November 2013 2:00
```

This example shows how to set summer time start and end dates:

```
Device(config)#clock summer-time PST date
20 March 2013 2:00 20 November 2013 2:00
```

示例：配置 MOTD 标语

此示例展示了如何配置 MOTD 标语并使用井号（#）作为起始定界字符：

```
Device(config)# banner motd #
This is a secure site. Only authorized users are allowed.
For access, contact technical support.
#
Device(config)#
```

此示例展示了以上配置显示的标语：

```
Unix>telnet 192.0.2.15
Trying 192.0.2.15...
Connected to 192.0.2.15.
Escape character is '^]'.
This is a secure site. Only authorized users are allowed.
For access, contact technical support.
User Access Verification
Password:
```

示例：配置登录标语

此示例展示了如何配置登录标语并使用井号（#）作为起始定界字符：

```
Device(config)# banner login $
Access for authorized users only. Please enter your username and password.
$
Device(config)#
```

示例：配置 MAC 地址更改通知陷阱

此示例展示了如何指定 172.20.10.10 作为 NMS，启用发往 NMS 的 MAC 地址通知陷阱，启用 MAC 地址更改通知特性，设置时间间隔为 123 秒，设置历史大小为 100 个条目，并在 MAC 地址添加到指定端口时启用陷阱：

```
Device(config)# snmp-server host 172.20.10.10 traps private mac-notification
Device(config)# snmp-server enable traps mac-notification change
Device(config)# mac address-table notification change
Device(config)# mac address-table notification change interval 123
Device(config)# mac address-table notification change history-size 100
Device(config)# interface gigabitethernet1/2/1
Device(config-if)# snmp trap mac-notification change added
```

示例：配置 MAC 门限值通知陷阱

此示例展示了如何指定 172.20.10.10 作为 NMS，启用 MAC 地址门限值通知特性，设置时间间隔为 123 秒，并设置限制为 78%：

```
Device(config)# snmp-server host 172.20.10.10 traps private mac-notification
Device(config)# snmp-server enable traps mac-notification threshold
Device(config)# mac address-table notification threshold
Device(config)# mac address-table notification threshold interval 123
Device(config)# mac address-table notification threshold limit 78
```

示例：向 MAC 地址表添加静态地址

此示例展示了如何把静态地址 c2f3.220a.12f4 添加到 MAC 地址表。在 VLAN 4 中收到使用此 MAC 地址作为目的地址的数据包时，数据包会被转发到指定的端口：

```
Device(config)# mac address-table static c2f3.220a.12f4 vlan 4 interface gigabitethernet1/1/1
```

示例：配置单播 MAC 地址过滤

此示例展示了如何启用单播 MAC 地址过滤以及如何配置交换机丢弃源或目的地址为 c2f3.220a.12f4 的数据包。在 VLAN 4 上收到源或目的地址为此 MAC 地址的数据包时，数据包会被丢弃：

```
Device(config)# mac address-table static c2f3.220a.12f4 vlan 4 drop
```

其他参考资料

相关文档

相关主题	文档标题
系统管理命令	*系统管理命令参考（Inspur 6650 交换机）*
网络管理配置	*网络管理配置指南（Inspur 6650 交换机）*
二层配置	*2/三层配置指南（Inspur 6650 交换机）*
VLAN 配置	*VLAN 配置指南（Inspur6650 交换机）*
平台无关的命令参考	*配置基础命令参考，Inspur INOS（Inspur 3850 交换机）*
平台无关的配置信息	*配置基础配置指南，Inspur INOS（Inspur 3850 交换机）* *IP 编址配置指南库，Inspur INOS（Inspur 3850 交换机）*

标准和 RFC

标准/RFC	标题
无	-

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的	http://www.icntnetworks.com

技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	

设备管理的特性历史与信息

版本	修订
Inspur INOS 11.3.1	引入了此特性。

引导完整性可视化

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用 Inspur 特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问 Inspur 特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于引导完整性可视化的信息

引导完整性可视化让 Inspur 的平台身份以及软件完整性信息变得可见且可操作。平台身份提供了平台的制造安装身份，而软件完整性显示了引导完整性度量信息，可以被用来评估平台是否引导了可信的代码。在启动过程中，软件会在引导程序活动的每个阶段创建校验和记录。可以获取此记录并与 Inspur 认证的记录进行比较，判断软件镜像是否真实。如果校验

和值不相同，则用户可能运行了未被 Inspur 认证的软件镜像，或是镜像被未授权的机构修改过。

验证软件镜像及硬件

此任务描述了如何获取交换机启动期间创建的校验和记录。在特权 EXEC 模式中输入以下命令：

注释： 在执行以下命令时，管理员可能会看到 CLI 上显示消息**% Please Try After Few Seconds**。这不表示 CLI 出错，而是表明正在设置所需的底层基础设施来获取所需的输出。建议等待几分钟再重试此命令。

消息**% Error retrieving SUDI certificate** and **% Error retrieving integrity data** 表示真正的 CLI 出错情况。

总步骤

1. **show platform sudi certificate** [**sign** [**nonce** *nonce*]]
2. **show platform integrity** [**sign** [**nonce** *nonce*]]

具体步骤

	命令或操作	目的
步骤 1	**show platform sudi certificate** [**sign** [**nonce** *nonce*]] **示例：** `Device#` **show platform sudi certificate signnonce 123**	显示指定 SUDI 的校验和记录。 • （可选）**sign**——显示签名 • （可选）**nonce**——输入随机值
步骤 2	**show platform integrity** [**sign** [**nonce** *nonce*]] **示例：** `Device#` **show platform integrity sign nonce 123**	显示启动阶段的校验和记录。 • （可选）**sign**——显示签名 • （可选）**nonce**——输入随机值

验证平台身份及软件完整性

验证平台身份

以下示例展示了 PEM 格式的安全唯一设备标识（Secure Unique Device Identity，SUDI）。第一个证书是 Inspur 根 CA 2048，第二个是 Inspur 下属 CA（ACT2 SUDI CA）。两个证书都可以在 http://www.icntnetworks.com 上进行验证。第三个是 SUDI 证书。

```
Device#show platform sudi certificate sign nonce 123
-----BEGIN CERTIFICATE-----
MIIDQzCCAiugAwIBAgIQX/h7KCtU3I1CoxW1aMmt/zANBgkqhkiG9w0BAQUFADA1
MRYwFAYDVQQKEw1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDExJDaXNjbyBSb290IENB
```

```
IDIwNDgwHhcNMDQwNTE0MjAxNzEyWhcNMjkwNTE0MjAyNTQyWjA1MRYwFAYDVQQK
Ew1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDExJDaXNjbyBSb290IENBIDIwNDgwggEg
MA0GCSqGSIb3DQEBAQUAA4IBDQAwggEIAoIBAQCwmrmrp68Kd6ficba0ZmKUeIhH
xmJVhEAyv8CrLqUccda8bnuoqrpu0hWISEWdovyD0My5jOAmaHBKeN8hF570YQXJ
FcjPFto1YYmUQ6iEqDGYeJu5Tm8sUxJszR2tKyS7McQr/4NEb7Y9JHcJ6r8qqB9q
VvYgDxFUl4F1pyXOWWqCZe+36ufijXWLbvLdT6ZeYpzPEApk0E5tzivMW/VgpSdH
jWn0f84bcN5wGyDWbs2mAag8EtKpP6BrXruOIIt6keO1aO6g58QBdKhTCytKmg9l
Eg6CTY5j/e/rmxrbU6YTYK/CfdfHbBcl1HP7R2RQgYCUTOG/rksc35LtLgXfAgED
o1EwTzALBgNVHQ8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUJ/PI
FR5umgIJFq0roIlgX9p7L6owEAYJKwYBBAGCNxUBBAMCAQAwDQYJKoZIhvcNAQEF
BQADggEBAJ2dhISjQal8dwy3U8pORFBi71R803UXHOjgxkhLtv5MOhmBVrBW7hmW
Yqpao2TB9k5UM8Z3/sUcuuVdJcr18JOagxEu5sv4dEX+5wW4q+ffy0vhN4TauYuX
cB7w4ovXsNgOnbFp1iqRe6lJT37mjpXYgyc81WhJDtSd9i7rp77rMKSsH0T8lasz
Bvt9YAretIpjsJyp8qS5UwGH0GikJ3+r/+n6yUA4iGe0OcaEb1fJU9u6ju7AQ7L4
CYNu/2bPPu8Xs1gYJQk0XuPL1hS27PKSb3TkL4Eq1ZKR4OCXPDJoBYVL0fdX4lId
kxpUnwVwwEpxYB5DC2Ae/qPOgRnhCzU=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIEPDCCAySgAwIBAgIKYQlufQAAAAADDANBgkqhkiG9w0BAQUFADA1MRYwFAYD
VQQKEw1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDExJDaXNjbyBSb290IENBIDIwNDgw
HhcNMTEwNjMwMTc1NjU3WhcNMjkwNTE0MjAyNTQyWjAnMQ4wDAYDVQQKEwVDaXNj
bzEVMBMGA1UEAxMMQUNUMiBTVURJIENBMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEA0m5l3THIxA9tN/hS5qR/6UZRpdd+9aE2JbFkNjht6gfHKd477AkS
5XAtUs5oxDYVt/zEbslZq3+LR6qrqKKQVu6JYvH05UYLBqCj38s76NLk53905Wzp
9pRcmRCPuX+a6tHF/qRuOiJ44mdeDYZo3qPCpxzprWJDPclM4iYKHumMQMqmgmg+
xghHIooWS80BOcdiynEbeP5rZ7qRuewKMpl1TiI3WdBNjZjnpfjg66F+P4SaDkGb
BXdGj13oVeF+EyFWLrFjj97fL2+8oauV43Qrvnf3d/GfqXj7ew+z/sXlXtEOjSXJ
URsyMEj53Rdd9tJwHky8neapszS+r+kdVQIDAQABo4IBWjCCAVYwCwYDVR0PBAQD
AgHGMB0GA1UdDgQWBBRI2PHxwnDVW7t8cwmTr7i4MAP4fzAfBgNVHSMEGDAWgBQn
88gVHm6aAgkWrSugiWBf2nsvqjBDBgNVHR8EPDA6MDigNqA0hjJodHRwOi8vd3d3
LmNpc2NvLmNvbS9zZWN1cml0eS9wa2kvY3JsL2NyY2EyMDQ4LmNybDBQBggrBgEF
BQcBAQREMEIwQAYIKwYBBQUHMAKGNGh0dHA6Ly93d3cuY2lzY28uY29tL3NlY3Vy
aXR5L3BraS9jZXJ0cy9jcmNhMjA0OC5jZXIwXAYDVR0gBFUwUzBRBgorBgEEAQkV
AQwAMEMwQQYIKwYBBQUHAgEWNWh0dHA6Ly93d3cuY2lzY28uY29tL3NlY3VyaXR5
L3BraS9wb2xpY2llcy9pbmRleC5odG1sMBIGA1UdEwEB/wQIMAYBAf8CAQAwDQYJ
KoZIhvcNAQEFBQADggEBAGh1qclr9tx4hzWgDERm371yeuEmqcIfi9b9+GbMSJbi
ZHc/CcCl0lJu0a9zTXA9w47H9/t6leduGxb4WeLxcwCiUgvFtCa51Iklt8nNbcKY
/4dw1ex+7amATUQO4QggIE67wVIPu6bgAE3Ja/nRS3xKYSnj8H5TehimBSv6TECi
i5jUhOWryAK4dVo8hCjkjEkzu3ufBTJapnv89g9OE+H3VKM4L+/KdkUO+52djFKn
hyl47d7cZR4DY4LIuFM2P1As8YyjzoNpK/urSRI14WdIlplR1nH7KNDl5618yfVP
0IFJZBGrooCRBjOSwFv8cpWCbmWdPaCQT2nwIjTfY8c=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIDhzCCAm+gAwIBAgIEAJT3DDANBgkqhkiG9w0BAQsFADAnMQ4wDAYDVQQKEwVD
```

```
aXNjbzEVMBMGA1UEAxMMQUNUMiBTVURJIENBMB4XDTE1MTExNDA5MzMzN1oXDTI1
MTExNDA5MzMzN1owczEsMCoGA1UEBRMjUElEOldTLUMzNjUwLTEyWDQ4VVEgU046
RkRPMTk0NkJHMDUxDjAMBgNVBAoTBUNpc2NvMRgwFgYDVQQLEw9BQ1QtMiBMaXRl
IFNVREkxGTAXBgNVBAMTEFdTLUMzNjUwLTEyWDQ4VVEwggEiMA0GCSqGSIb3DQEB
AQUAA4IBDwAwggEKAoIBAQC6SARWyImWrRV/x7XQogAE+02WmzKki+4arMVBvl9o
GgvJfkoJDdaHOROSUkEE3qXtd8N3lfKy3TZ+jtHD85m2aGz6+IRx/e/lLsQzi6dl
WIB+N94pgecFBONPR9wJriox1IGD3B43b0hMLkmro4R5Zrs8XFkDo9k1tBU7F2O7
GEzb/WkO5NLexznef2Niglx9fCDL0HC27BbsR5+03p8jhG0+mvrp8M9du1HKiGin
ZIV4XgTMp1/k/TVaIepEGZuWM3hxdUZjkNGG1c1m+oB8vLX3UlSL76sDBBoiaprD
rjXBgBIozyFW8tTjh50jMDG84hKD5s31ifOe4KpqEcnVAgMBAAGjbzBtMA4GA1Ud
DwEB/wQEAwIF4DAMBgNVHRMBAf8EAjAAME0GA1UdEQRGMESgQgYJKwYBBAEJFQID
oDUTM0NoaXBJRD1VWUpOTlZJMENBUkhVM1Z1SUVSbFl5QXlPQ0F4TXpvek5Ub3lN
U0EwS0NnPTANBgkqhkiG9w0BAQsFAAOCAQEADjtM8vdlf+p1WKSKX1C1qQ4aEnD5
p8T5e4iTer7Y1fbCrHIEEm3mnip+568j299z0H8V7PDp1ljuLHyMFTC+945F9RfA
eAuVWVb5A9dnGL8MssBJe2lVSnZwrWkT1EIdxLYrTiPAQHtll6CN77S4u/f71oYE
tzPE5AGfyGw7ro1MEPVGffaQmYUDAwKFNBH1uI7c2S1qlwk4WWZ6xxci+lhaQnIG
pWzapaiAYL1XrcBz4KwFc1ZZpQT6hHw24jzYaYimvCo+/kSKuA9xNdtSu18ycox0
zKnXQ17s6aChMMt7Y8Nh4iz9BDejoOF6/b3sM0wRi+2/4j+6/GhcMRs0Og==
-----END CERTIFICATE-----
Signature version: 1
Signature:
405C770D802B73947EDBF8DD0D2C8180F10D4B3EF9699444514219C579D2ED52F7D5
83E0F4408133FC4E9F549B2EB1C21725F7CB1C79F98271E47E780E703E674723880F
B52D4963E1D1FB9787B38E28B8E696570A180B7A2F1311B1F174EAA79F55DB4765DF
67386126D899E07EDF6C26E0A81272EAA114437DD03F26992937082756AE1F1BFAFB
BFACD6BE9CF9C84C961FACE9FA0FEE64D85AE4FA0086969D0702C536ABDB8FBFDC47
C14C17D02FEBF4F7F5BB24D2932FA876F56B4C07816270AA0B4195C53D975C85AEAE
3A74F2DBF293F52423ECB7B8539667080A9C57DA3E4B08B2B2CA623B2CBAF7080A0A
EB09B222E5B756970A3AA27E0F1D17C8A243
```

可以对三个证书、签名版本以及用户提供的随机数值进行 RSA 2048 签名。

```
RSA PKCS#1v1.5 Sign {<Nonce (UINT64)> || <Signature Version (UINT32)> || <Inspur Root CA2048
cert (DER)> ||<Inspur subordinate CA (DER)> || <SUDI certificate (DER)> }
```

Inspur 管理解决方案能够解释以上输出。不过也可以通过使用 OpenSSL 命令的简单脚本来显示平台身份并验证签名，进而确认 Inspur 唯一设备身份。

```
[linux-host:~]openssl x509 -in sudicert.pem -subject -noout
subject= /serialNumber=PID:WS-C6650-12X48UQ SN:FDO1946BG05/O=Inspur/OU=ACT-2
Lite SUDI/CN=WS-C6650-12X48UQ
```

验证软件完整性

以下示例展示了启动阶段的校验和值。对于软件成功启动的三个阶段中的每个阶段都会显示一个哈希度量值。这些哈希值可以用来与 Inspur 提供的参考值进行比较。可以选择对输出进行签名，这让验证者可以确认这些输出是真实且未被修改的。可以提供随机数来对抗重放攻击。

```
Device #show platform integrity sign nonce 456
```

```
Platform: WS-C6650-12X48UQ
Boot Loader Version: CAT3K_CAA Boot Loader (CAT3K_CAA-HBOOT-M) Version 4.16, engineering
software (D)
Boot Loader Hash: DB5A686E9F4CE358481DE3AF8B9C762F0A604E3B4764DF2A351F176E3D7
D3C60EB85C02906BD8CF28228C0DFC2AA8960CAFE6675D696E4ABA0CD687C0609E7E2
Boot 0 Version: F01062R15.0508d68fa2015-09-15
Boot 0 Hash: 6EF15CD54D3C66A8B644194A67B7ED57044C8C2E0EECB69736A7FFEC1F6D0EAD
OS Version: 2016-10-18_10.57_mundru
OS Hash: 4C85AECC88DAA49D940BBF65B1F17269F55C8D98DEFB4140F981923AA961140293E1
3B3E6E68CE3F8ED7F596CD858ACDD4BEF6538F59C1E243C351353026E6CD
PCR0: 90214167AAF35C06B2AC97292596E5669EAB72578FCDAD0B91746683BAA7B2B0
PCR8: FC2CE1BAC397F97008936DF372A2218BB16A798222B8FF55A7B6AEDA8018EDF5
Signature version: 1
Signature:
632A724F1AB6ADE134F6B0E8724D2052B3157F45B47E547763EE224A848E807CD737600587FF68
2526A8FE354A116CC9EDEBD9C659B9927336542EE4295084368327D01BD22AB4849BB3C007B6EB
B67708685FD6BC85DD045431E19A389FEB358894D4FBCF7C0FC960AC9133B61099DFD507F316C1
BF82F7F98687C7E7E8F99355DC1A95BD511B0B8DCB0CA909828F9EFBDF18847930392A8E3D072D
F3D90536880BAE9B7D7CF0E301D3F5AF16E7517FC2700E2F75911B836D6559A18E15B4CF452555
91656DF22DFF73392F777AEB796BCF9AC046C581ADEF19CA48A98F620BB58A79B32DA8B3BFB1CF
8399468A096E2F0C54B8B3ECD15EE3FE2C5ABDB5A029
```

可选的 RSA 2048 签名使用 SUDI 的私有密钥生成，并可以通过 SUDI 证书中包含的 SUDI 公有密钥进行验证。系统会显示对 PCR 值、签名版本以及用户提供的随机数进行的签名。

```
RSA PKCS# 1 v1.5 Sign { <Nonce (UINT64)> || <Signature Version (UINT32)> || <PCR0 (32 bytes)>||
<PCR8 (32 bytes)> }
```

Inspur 管理解决方案能够解释以上输出，与发布的 Inspur 值进行比较，并验证签名。

执行设备设置配置

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用 Inspur 特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问 Inspur 特性导航（Inspur Feature Navigator），可以访问

http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于执行设备设置配置的信息

在执行包括 IP 地址分配以及 DHCP 自动配置等初始设备配置任务之前，请查看本模块的内容。

设备引导过程

要启用设备，用户需遵循硬件安装指南中的步骤，安装设备，给设备供电，并设置初始设备配置（IP 地址、子网掩码、默认网关、密码、Telnet 密码等等）。
正常的启动过程涉及到引导加载程序的以下活动：

- 执行低级 CPU 初始化。初始化控制物理内存映射到何处，以及其数量、速度等信息的 CPU 寄存器。
- 对 CPU 子系统执行加电自检（power-on self-test，POST）并测试系统的 DRAM。
- 初始化系统主板上的文件系统。
- 把默认的操作系统软件镜像加载到内存中并起动设备。

引导加载程序提供了在系统加载前访问文件系统的能力。正常情况下，引导加载程序仅被用来加载、解压并启动操作系统。在引导加载程序把 CPU 的控制权交给操作系统之后，直到下一次系统重置或启动之前它都不会活动。
如果操作系统的问题严重了到无法被使用的情况，引导加载程序也提供了隐藏的访问系统方式。隐藏机制提供了足够的系统访问权限，在必要时管理员可以使用 Xmodem 协议重新安装操作系统软件，在丢失或遗忘密码时恢复使用，并最终重启操作系统。
在指定设备信息之前，确保把 PC 或终端连接到了控制台端口，或者把 PC 连接到了以太网管理端口，并确保配置 PC 或终端模拟软件的波特率及字符格式满足如下设备控制台端口的设置：

- 波特率默认为 9600。
- 数据位默认为 8。
  注释： 如果设置数据位选项为 8，请设置奇偶校验和选项为无设置。
- 停止位默认为 2。
- 奇偶校验和默认为无设置。

软件安装程序特性

交换机上支持以下软件安装器特性：

- 在单独的交换机、交换机堆栈或交换机堆栈的交换机子集中安装软件包。如果配置了交换机堆栈，默认在其中所有交换机上进行安装。
- 在交换机堆栈中，Inspur 建议所有交换机都使用安装模式。
- 软件回滚的之前安装的软件包集合。
- 在引导闪存中没有合法的已安装软件包时进行紧急安装。
- 对加入交换机堆栈的使用不兼容软件的交换机进行自动升级。
- 使用一台交换机上的软件包作为源，并安装到交换机堆栈中的另一台交换机上。

注释： 软件安装及回滚必须仅在运行安装模式时进行。可以使用 EXEC 命令 **software expand**

来把软件包启动模式切换为安装模式。

软件引导模式

设备支持两种引导软件包的模式：

- 安装模式
- 捆绑包模式

安装引导模式

可以使用闪存中的软件包规划文件把设备引导为安装模式：

```
device: boot flash:packages.conf
```

规划文件中包含了要引导、挂载与运行的软件包列表。每个安装的软件包中的 ISO 文件系统会直接从闪存中挂载到根文件系统上。

注释： 用于引导安装模式的软件包以及规划文件必须位于闪存中。不支持通过 usbflash0: 或 tftp:引导为安装模式。

捆绑包引导模式

可以使用软件捆绑包（.bin）文件把设备引导为软件包引导模式：

```
switch: boot flash:cat3850-universalk9.SSA.03.08.83.EMD.150-8.83.EMD.bin
```

软件捆绑包中含有的规划文件决定要引导、挂载并运行哪些软件包。软件包会被从捆绑包中提取并拷贝到 RAM 中。每个软件包中的 ISO 文件系统会被挂载到根文件系统上。

与安装引导模式不同，在软件捆绑包模式中进行引导时，会使用与所用捆绑包大小相同的额外内存空间。

与安装引导模式不同，捆绑包引导模式可以在以下几个位置上使用：

- flash:
- usbflash0:
- tftp:

注释： 捆梆包引导模式不支持自动安装以及智能安装功能。

注释： 捆梆包引导模式中不支持 AP 镜像预下载特性。更多关于预下载特性的信息，参见 Inspur WLC 5700 系列的 *预加载镜像到接入点*一章。

交换机堆栈的引导模式

堆栈中的所有交换机都必须运行在安装模式或者捆梆包引导模式中。堆栈不支持混合使用引导模式。如果一台新的交换机尝试加入的堆栈，而其引导模式与活跃交换机不同，新交换机会进入 V-mismatch 的状态。

如果使用混合模式的交换机堆栈同时启动，则除活跃交换机之外的所有交换机会有 V-mismatch 的状态。如果引导模式不支持自动升级，则交换机堆栈成员必须在与活跃交换机相同的引导模式中进行重新引导。

如果堆栈运行在安装模式中，可以使用自动升级特性来升级一台尝试加入交换机堆栈的交换

机。
自动升级特性会把新交换机的引导模式改为安装模式。如果堆栈运行在捆梆包引导模式中，则自动升级特性不可用。此时管理员应使用捆梆包模式引导新的交换机，让它能够加入交换机堆栈。
以下示例展示了引导模式与活跃交换机不匹配时，尝试加入交换机堆栈的交换机状态：

```
Device# show switch
Switch/Stack Mac Address : 6400.f125.1100 - Local Mac Address
Mac persistency wait time: Indefinite
H/W Current
Switch# Role Mac Address Priority Version State
-----------------------------------------------------------
1 Member 6400 f125.1a00 1 0 V-Mismatch
*2 Active 6400.f125.1100 1 V01 Ready
Device
```

指定设备信息

可以通过设备设置程序、DHCP 服务器或通过手动方式指定 IP 地址。
如果希望按照提示输入特定的 IP 信息，可以使用设备设置程序。通过此程序还可以配置主机名以及使能密码，指定 Telnet 密码（为远程管理提供安全性），把交换机配置为集群中的命令交换机或成员交换机，或配置为单独的交换机。
配置 DHCP 服务器后，可以使用 DHCP 服务器集中化控制并自动分配 IP 信息。
注释： 如果使用 DHCP，在设备接收动态分配的 IP 地址并读取配置文件之前，不要回答设置程序中的任何问题。
熟悉设备配置步骤的有经验用户可以手动配置设备。否则，请使用*引导过程*一节中描述的设置程序。

默认的交换机信息

表 176：默认的交换机信息

特性	默认设置
IP 地址以及子网掩码	未定义 IP 地址或子网掩码。
默认网关	未定义默认网关。
使能密码	未定义密码。
主机名	工厂分配的默认主机名是 Device。
Telnet 密码	未定义密码。
集群命令交换机功能	禁用。
集群名称	未定义集群名称。

基于 DHCP 的自动配置概述

DHCP 为 Internet 主机以及网络互连设备提供配置信息。该协议由两个组件组成：一个用于从 DHCP 服务器向设备传送配置参数，另一个用来给设备分配网络地址。DHCP 构建在客户端-服务器模型上，指定的 DHCP 服务器会给动态配置的主机分配网络地址并传送配置参数。

设备可以同时作为 DHCP 客户端和 DHCP 服务器。
在基于 DHCP 的自动配置过程中，设备（DHCP 客户端）在启动过程中会使用 IP 地址信息以及配置文件进行自动配置。
使用基于 DHCP 的自动配置时，设备上的 DHCP 客户端无需进行配置。然而，需要配置 DHCP 服务器使用与 IP 地址相关的多个租用选项。
如果希望使用 DHCP 来中继网络位置上的配置文件，可能还需要配置简单文件传输协议（Trivial File Transfer Protocol，TFTP）服务器以及域名系统（Domain Name System，DNS）服务器。
注释： 建议在交换机堆栈以及 DHCP、DNS 和 TFTP 服务器之间使用冗余连接。这有助于保证在一个连接的堆栈成员从交换机堆栈移除之后仍能访问这些服务器。
设备的 DHCP 服务器可以与设备在同一个 LAN 上，也可以在不同的 LAN 中。如果 DHCP 服务器运行在不同的 LAN 中，应该在设备与 DHCP 服务器之间配置 DHCP 中继设备。中继设备可以在两个直连的 LAN 之间转发广播流量。路由器不会转发广播数据包，只会基于接收包的目的 IP 地址转发数据包。
基于 DHCP 的自动配置会代替设备上的 BOOTP 客户端功能。

DHCP 客户端请求过程

启动设备时，DHCP 客户端会被调用，并在设备上没有配置文件时向 DHCP 服务器请求配置信息。如果存在配置文件，且配置中特定的被路由接口上包含接口配置命令 **ip address dhcp**，则会调用 DHCP 客户端为这些接口请求 IP 地址信息。
以下是 DHCP 客户端与 DHCP 服务器之间交换的消息序列。

图 134：DHCP 客户端及服务器消息交换

Switch A	交换机 A
DHCP server	DHCP 服务器
DHCPDISCOVER（broadcast）	DHCPDISCOVER（广播）
DHCPOFFER（unicast）	DHCPOFFER（单播）
DHCPREQUEST（broadcast）	DHCPREQUEST（广播）
DHCPACK（unicast）	DHCPACK（单播）

客户端交换机 A 会广播发送 DHCPDISCOVER 消息来定位 DHCP 服务器。DHCP 服务器会在 DHCPOFFER 单播消息中给客户端提供配置参数（比如 IP 地址、子网掩码、网关 IP 地址、DNS IP 地址、IP 地址的租期等等）。
在 DHCPREQUEST 广播消息中，客户端为提供的配置信息给 DHCP 服务器返回一个正式的请求。正式请求会广播发送，使所有其他收到这个客户端发送的 DHCPDISCOVER 消息的 DHCP 服务器收回分配给该客户端的 IP 地址。
DHCP 服务器给客户端发送一个 DHCPACK 单播消息，确认 IP 地址已经分配给了该客户端。通过这条消息，客户端和服务器被绑定在一起，客户端会使用从服务器收到的配置信息。设备接收的信息数量取决于如何配置 DHCP 服务器。
如果在 DHCPOFFER 单播消息中发送给客户端的配置参数是非法的（存在配置错误），客户端会给 DHCP 服务器发送 DHCPDECLINE 广播消息。
DHCP 服务器会给客户端发送 DHCPNAK 拒绝广播消息，表示提供的配置参数未被分配，在参数协商过程中发生了错误，或者客户端回应 DHCPOFFER 消息过慢（DHCP 服务器已经把参数分配给了另一个客户端）。

DHCP客户端可能会收到来自多个DHCP或BOOTP服务器的提议，且可以接收任意一个提议，然而，客户端通常接受收到的第一个提议。来自 DHCP 服务器的提议不能保证 IP 地址被分配给了客户端，然而 DHCP 服务器通常会预留该地址，直到客户端有机会正式请求地址。如果设备接受了来自 BOOTP 服务器的应答并且进行了自我配置，它会广播 TFTP 请求来获取设备配置文件。

DHCP 主机名选项允许一组设备通过中心化管理的 DHCP 服务器获取主机名以及标准配置。客户端会在 DHCPDISCOVER 消息中包含一个可选的 12 字段，向 DHCP 服务器请求主机名以及其他配置参数。除了通过 DHCP 获取的主机名，所有客户端的配置文件都相同。

如果客户端有默认的主机名（未配置全局配置命令 **hostname** *name* 或输入全局配置命令 **no hostname** 移除了主机名），输入接口配置命令 **ip address dhcp** 时，DHCP 主机名选项不会包含在数据包中。此时，如果客户端在为接口获取 IP 地址时通过 DHCP 交互接收到了 DHCP 主机名选项，客户端会接受 DHCP 用户名选项，并设置标志位以表示系统配置了主机名。

基于 DHCP 的自动配置及镜像更新

可以使用 DHCP 镜像升级特性，配置 DHCP 服务器给网络中的一台或多台设备下载新镜像以及新配置文件。对网络中的所有交换机同时进行镜像以及配置的升级能够确保加入到网络的每台新设备可以接收到相同的镜像及配置。

有两种类型的 DHCP 镜像升级：DHCP 自动配置以及 DHCP 自动镜像升级。

基于 DHCP 自动配置的限制

- 如果网络中没有处于 up 状态且未分配 IP 地址的三层接口，有保存配置的基于 DHCP 的自动配置过程会停止。
- 除非配置超时，否则有保存配置的基于 DHCP 的自动配置会无限次尝试下载 IP 地址。
- 如果配置文件不能被下载或配置文件损坏，自动安装过程会停止。
- 通过 TFTP 下载的配置文件会与运行配置中的现有配置合并，但是不会保存到 NVRAM 中，除非输入了特权 EXEC 命令 **write memory** 或 **copy running-configuration startup-configuration**。如果下载的配置被保存到启动配置中，在后续系统重启过程中该特性不会被触发。

DHCP 自动配置

DHCP 自动配置会从 DHCP 服务器给网络中的一台或多台设备下载配置文件。下载的配置文件会成为设备的运行配置。它不会覆盖保存在闪存中的启动配置，直到重启设备。

DHCP 自动镜像更新

可以使用 DHCP 自动镜像更新以及 DHCP 自动配置，给网络中的一台或多台设备同时下载配置及镜像。下载新配置以及新镜像的配置可以是空白的（或者只加载了默认工厂配置）。

如果新配置下载到了已经存在配置的交换机上，下载的配置会被附加到交换机存储的配置文件（现有配置不会被下载配置覆盖）。

要在设备上启用 DHCP 自动镜像更新，镜像以及配置所在的 DNS 服务器必须配置了正确的选项 67（配置文件名），选项 66（DHCP 服务器主机名），选项 150（TFTP 服务器地址）以及选项 125（Inspur INOS 镜像文件描述）设置。

把设备安装到网络中之后，自动镜像更新特性开始执行。下载的配置文件会被保存到设备的运行配置中，且新的镜像会下载并在设备上安装。重启设备时，配置会被存储在设备的保存

配置中。

DHCP 服务器配置指南

按照以下指南把设备配置为 DHCP 服务器：

- 应该给 DHCP 服务器配置预留的租用信息，将租用地址与每台设备的硬件地址绑定。
- 如果希望设备接收 IP 地址信息，必须给 DHCP 服务器配置以下租用选项：
  - 客户端 IP 地址（必要）
  - 客户端子网掩码（必要）
  - DNS 服务器 IP 地址（可选）
  - 路由器 IP 地址（设备应使用的默认网关地址）（必要）
- 如果希望设备通过 TFTP 服务器接收配置文件，必须给 DHCP 服务器配置以下租用选项：
  - TFTP 服务器名（必要）
  - 引导文件名（客户端所需的配置文件名称）（建议）
  - 主机名（可选）
- 根据 DHCP 服务器的设置不同，设备可以接收 IP 地址信息或配置文件，也可以同时接收两者。
- 如果不给 DHCP 服务器配置之前上述租用选项，它会只给客户端回应配置了的参数。如果应答中没有 IP 地址以及子网掩码，设备不会被配置。如果未找到路由器 IP 地址或 TFTP 服务器名称，设备可能会发送广播 TFTP 请求。其他租用选项不可用不会影响自动配置过程。

设备可以作为 DHCP 服务器使用。默认情况下，Inspur INOS DHCP 服务器以及中继代理特性在设备上启用但未被配置（这些特性不会工作）。

TFTP 服务器的目的

基于 DHCP 服务器的配置，设备尝试从 TFTP 服务器下载一个或多个配置文件。如果配置 DHCP 服务器给设备回复 IP 连通 TFTP 服务器的所有必要选项，而且为 DHCP 服务器配置了 TFTP 服务器名称、地址以及配置文件，设备会尝试通过指定的 TFTP 服务器下载指定配置文件。如果没有指定配置文件及 TFTP 服务器，或者配置文件不能被下载，设备会尝试使用多种文件名和 TFTP 服务器地址的组合下载配置文件。文件包含指定的配置文件或这些文件：network-config、inspurnet.cfg、hostname.config 或 hostname.cfg，其中 *hostname* 是设备当前的主机名。使用的 TFTP 地址包括指定的 TFTP 服务器地址或者广播地址（255.255.255.255）。要使设备能成功下载配置文件，TFTP 服务器的基本目录中必须包含一个或多个配置文件。这些文件可以有：

- DHCP 应答中说明的配置文件（实际设备的配置文件）。
- network-confg 或 inspurnet.cfg 文件（默认配置文件）。
- router-confg 或 inspurrtr.cfg 文件（这些文件包含所有设备通用的命令。正常情况下，如果正确配置了 DHCP 以及 TFTP 服务器，这些文件不会被访问）。

如果在 DHCP 服务器租用数据库中指定了 TFTP 服务器名称，必须也在 DNS 服务器数据库中配置 TFTP 服务器名称到 IP 地址的映射。

如果使用的 TFTP 服务器与设备在不同的 LAN 中，或者要通过广播地址访问（如果 DHCP 服务器应答不含有之前所述的所有必须信息，会发生此情况），必须配置使用中继把 TFTP 数据包转发到 TFTP 服务器上。首选方案是给 DHCP 服务器配置所有必须的信息。

DNS 服务器的目的

DHCP 服务器使用 DNS 服务器把 TFTP 服务器的名称解析为 IP 地址。必须在 DNS 服务器上配置 TFTP 服务器名称到 IP 地址的映射。TFTP 服务器包含着设备的配置文件。

可以在 DHCP 服务器的租用数据库中配置 DNS 服务器的 IP 地址，DHCP 应答会读取这些信息。在租用数据库中最多可以输入两个 DNS 服务器 IP 地址。

DNS 服务器可以与设备在相同 LAN 上，也可以在不同的 LAN 上。如果在不同 LAN 上，设备必须可以通过路由器访问该服务器。

如何获取配置文件

根据 IP 地址可用性以及 DHCP 预留租用信息中配置文件名的不同，设备会通过以下方式获取配置信息：

- 为设备预留了 IP 地址以及配置文件名，且通过 DHCP 应答提供给设备（读取一个文件的方式）。

  设备从 DHCP 服务器接收 IP 地址、子网掩码、TFTP 服务器地址以及配置文件名。设备给 TFTP 服务器发送单播消息，提取服务器基本目录上对应名称的配置文件，接收完毕后，设备完成启动过程。
- 为设备预留了 IP 地址以及配置文件名，但是 DHCP 应答中没有提供 TFTP 服务器地址（读取一个文件的方式）。

  设备从 DHCP 服务器接收 IP 地址、子网掩码以及配置文件名。设备给 TFTP 服务器发送广播消息，提取服务器基本目录上对应名称的配置文件，接收完毕后，设备完成启动过程。
- 仅为设备预留了 IP 地址并通过 DHCP 应答提供给设备，未提供配置文件名（读取两个文件的方式）。

  设备从 DHCP 服务器接收 IP 地址、子网掩码以及 TFTP 服务器地址。设备给 TFTP 服务器发送单播消息，提取默认配置文件 network-confg 或 inspurnet.cfg（如果无法读取 network-confg，设备会读取 inspurnet.cfg 文件）。

  默认配置文件包含设备主机名到 IP 地址的映射。设备使用文件中的信息填写自己的主机表，并获取其主机名。如果文件中未找到主机名，设备会使用 DHCP 应答中的主机名。如果 DHCP 应答中未指定主机名，设备会使用默认的 *Switch* 作为主机名。

  在通过默认配置文件或者 DHCP 应答获取主机名之后，设备会从 TFTP 服务器上读取与自己主机名相同的配置文件（根据之前读取的是 network-confg 或 inspurnet.cfg，可能读取(*hostname*-confg 或 *hostname*.cfg）。如果读取了 inspurnet.cfg 文件，主机的文件名会被截取为八个字符。

  如果设备不能读取 network-confg、inspurnet.cfg 或 hostname 文件，它会读取 router-confg 文件。如果设备不能读取 router-confg 文件，它会读取 inspurrtr.cfg 文件。

注释： 如果没能从 DHCP 应答中获取 TFTP 服务器，所有通过单播传输读取配置文件的尝试都失败，或者 TFTP 服务器名称不能被解析为 IP 地址，则设备会广播发送 TFTP 服务器请求。

如何控制环境变量

对于正常运行的设备，仅可以通过配置为 9600 bps 的控制台连接来进入引导加载程序模式。拔掉设备的电源线，重连电源线时按住 **Mode** 键。在系统所有琥珀色的 LED 灯都打开并保持

常亮后，可以放开 **Mode** 键。此后会出现引导加载程序设备提示符。
设备引导加载程序提供对非易失性环境变量的支持，可以由此控制引导加载程序或系统运行的任何其他软件的运行方式。引导加载程序的环境变量与可以在 UNIX 或 DOS 系统上设置的环境变量类似。
有值的环境变量会被存储在闪存文件系统之外的闪存内存中。
这些文件的每一行都包含一个环境变量名，一个等号，接着是变量值。未出现的变量没有值；如果变量在文件中列出，就算值是空字符串，该变量也有值。设置为空字符串（如“”）的变量是有值的变量。许多环境变量都有预定义的默认值。
可以通过访问引导加载程序或使用 Inspur INOS 的命令来更改环境变量的设置。在正常情况下，没有必要更改环境变量的设置。

常用的环境变量
下表描述了众多常用环境变量的功能。
表 177：常用环境变量

变量	引导加载程序命令	Inspur INOS 全局配置命令
BOOT	**set BOOT** *filesystem* **:/** *file-url* *…* 设置自动引导时尝试加载并执行的可执行文件列表，用分号隔离。	**boot system** {*filesystem* **:** */file-url …* |**switch** {*number* | **all**}} 指定下一次引导循环期间要加载的 Inspur INOS 镜像以及要加载镜像的堆栈成员。此命令常用来更高 BOOT 环境变量的设置。 软件包规划文件，也称为 *packages.conf* 文件，被系统用来决定在启动过程中要激活哪些软件包。 在安装模式进行引导时，**boot**命令指定的软件包规划文件会被用来决定激活哪些软件包。比如**bootflash:packages.conf**。 在捆绑包模式进行引导时，引导的捆绑包中的软件包规划文件会被用来激活捆绑包中的软件包。比如 **boot flash:image.bin**。
MANUAL_BOOT	**set MANUAL_BOOT yes** 决定交换机是自动引导还是手动引导。 合法的值是 1、yes、0 和 no。如果设置为 0 或 no，引导加载程序会尝试自动引导系统。如果设置为其他值，必须在引导加载程序模式中手动引导交换机。	**boot manual** 在下一次引导循环期间手动引导交换机，并更改环境变量 MANUAL_BOOT 的设置。 下次重启系统时，交换机会在引导加载程序模式中。要引导系统，需使用引导加载程序命令 **boot flash:***filesystem* **:/** *file-url*，并指定可引导镜像的名称。
CONFIG_FILE	**set CONFIG_FILE flash:/***file-url* 更改 Inspur INOS 用来读写系统配置的非易失性拷贝的文件名。	**boot config-file flash:/** *file-url* 指定 Inspur INOS 用来读写系统配置的非易失性拷贝的文件名。此命令会更改 CONFIG_FILE 环境变量。
SWITCH_NUMBER	**set SWITCH_NUMBER***stack-memb*	**switch** *current-stack-member-number***renumber***new-stack*

	er-number 更改堆栈成员的成员编号。	*-member-number* 更改堆栈成员的成员编号。
SWITCH_PRIORITY	**set SWITCH_PRIORITY***stack-member-number* 更改堆栈成员的优先级值。	**switch** *stack-member-number* **priority***priority-number* 更改堆栈成员的优先级值。
BAUD	**set BAUD** *baud-rate*	**line console 0** **speed** *speed-value* 配置波特率。
ENABLE_BREAK	**set ENABLE_BREAK yes/no**	**boot enable-break switch yes/no** 允许打断自动引导循环。用户有 5 秒中的时间来输入 **break** 命令。

TFTP 的环境变量

当交换机通过以太网管理端口连接到 PC 时，可以使用 TFTP 给引导加载程序下载或上传配置文件。确保配置了下表中的环境变量。

表 178：TFTP 的环境变量

变量	描述
MAC_ADDR	指定交换机的 MAC 地址。 **注释：**建议不修改此变量。 然而，如果在引导加载程序启用之后修改此变量，或者变量值与保存的值不同，请在使用 TFTP 之前输入此命令。
IP_ADDR	为交换机指定关联 IP 子网中的 IP 地址及子网掩码。
DEFAULT_ROUTER	指定默认网关的 IP 地址以及子网掩码。

计划重新加载软件镜像

可以计划设备在之后的时间重新加载软件镜像（如在深夜或在设备使用较少的周末），也可以在网络范围内同步进行重新加载（如对网络中的所有设备执行软件更新）。

注释： 计划的重新加载必须在大约 24 天内发生。

有以下重新加载选项：

- 重新加载软件在指定的分钟或小时分钟后生效。重新加载必须在约 24 小时内发生。可以使用至多 255 字符的字符串来指明重新加载的原因。
- 重新加载在指定的时间发生（使用 24 小时制）。如果指定了月份和日期，重新加载会计划在指定的时间和日期发生。如果未指定月份和日期，重新加载会在当天（如果指定的时间比当前时间晚）或下一天（如果指定的时间比当前时间早）指定的时间发生。指定 00:00 会计划在午夜进行重新加载。

reload 命令会让系统停机。如果系统未设置成手动引导，则设备会自行重启。

如果设备配置了手动引导，请不要通过虚拟终端进行重启。此限制避免了设备进入引导加载程序模式后夺取了远程用户的控制权。

如果修改了配置文件，设备会在重新加载之前提示用户保存配置。在保存操作过程中，如果

环境变量 CONFIG_FILE 指向的启动配置文件不存在，系统会请求用户确认是够继续保存。此时如果继续操作，系统会在重新加载时进入设置模式。
要取消之前计划的重新加载，使用特权 EXEC 命令 **reload cancel**。

如何执行设备设置配置

使用 DHCP 给设备下载新镜像以及新配置要求管理员至少要配置两台设备。一台设备作为 DHCP 以及 TFTP 服务器，第二台设备（客户端）要配置下载新配置文件或是同时下载新配置文件以及新镜像文件。

配置 DHCP 自动配置（仅下载配置文件）

总步骤

1. configure terminal
2. ip dhcp pool *poolname*
3. boot *filename*
4. network *network-number mask prefix-length*
5. default-router *address*
6. option 150 *address*
7. exit
8. tftp-server flash:*filename.text*
9. interface *interface-id*
10. no switchport
11. ip address *address mask*
12. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**ip dhcp pool** *poolname* **示例:** `Device(config)#` **ip dhcp pool pool**	创建 DHCP 服务器地址池名称，并进入 DHCP 地址池配置模式。
步骤 3	**boot** *filename* **示例:** `Device(dhcp-config)#` **boot config-boot.text**	指定用作引导镜像的配置文件名称。
步骤 4	**network** *network-number mask prefix-length* **示例:** `Device(dhcp-config)#` **network 10.10.10.0 255.255.255.0**	指定 DHCP 地址池的网络地址以及掩码。 **注释：**前缀长度指定了组成地址前缀的比特数。前缀是指定客户端网络掩码的另一种方式。前缀长度必须在正斜线（/）之后。

步骤 5	**default-router** *address* **示例:** `Device(dhcp-config)#` **default-router 10.10.10.1**	为 DHCP 客户端指定默认路由器的 IP 地址。
步骤 6	**option 150** *address* **示例:** `Device(dhcp-config)#` **option 150 10.10.10.1**	指定 TFTP 服务器的 IP 地址。
步骤 7	**exit** **示例:** `Device(dhcp-config)#` **exit**	返回全局配置模式。
步骤 8	**tftp-server flash**:*filename.text* **示例:** `Device(config)#` **tftp-serverflash:config-boot.text**	指定 TFTP 服务器上的配置文件。
步骤 9	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet1/0/4**	指定要接收配置文件的客户端地址。
步骤 10	**no switchport** **示例:** `Device(config-if)#` **no switchport**	将接口设置为三层模式。
步骤 11	**ip address** *address mask* **示例:** `Device(config-if)#` **ip address 10.10.10.1 255.255.255.0**	指定接口的 IP 地址以及掩码。
步骤 12	**end** **示例:** `Device(config-if)#` **end**	返回特权 EXEC 模式。

配置 DHCP 自动镜像更新（下载配置文件及镜像）

此任务描述了对现有设备进行 TFTP 以及 DHCP 设置的过程，以使用 DHCP 自动配置来进行新交换机的安装设置。

在开始前

必须先创建一个要被上传到设备上的 text 文件（如 autoinstall_dhcp）。在 text 文件中输入希望下载的镜像名（如 c3750e-ipservices-mz.122-44.3.SE.tarc3750x-ipservices-mz.122-53.3.SE2.tar）。此镜像必须是 tar 文件，而不能是 bin 文件。

总步骤

1. **configure terminal**
2. **ip dhcp pool** *poolname*
3. **boot** *filename*

4. **network** *network-number mask prefix-length*
5. **default-router** *address*
6. **option 150** *address*
7. **option 125** *hex*
8. **copy tftp flash** *filename.txt*
9. **copy tftp flash** *imagename.bin*
10. **exit**
11. **tftp-server flash:** *config.text*
12. **tftp-server flash:** *imagename.bin*
13. **tftp-server flash:** *filename.txt*
14. **interface** *interface-id*
15. **no switchport**
16. **ip address** *address mask*
17. **end**
18. **copy running-config startup-config**

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**ip dhcp pool** *poolname* **示例:** `Device(config)#` **ip dhcp pool pool**	创建 DHCP 服务器地址池名称，并进入 DHCP 地址池配置模式。
步骤 3	**boot** *filename* **示例:** `Device(dhcp-config)#` **boot config-boot.text**	指定用作引导镜像的配置文件名称。
步骤 4	**network** *network-number mask prefix-length* **示例:** `Device(dhcp-config)#` **network 10.10.10.0 255.255.255.0**	指定 DHCP 地址池的网络地址以及掩码。 **注释：**前缀长度指定了组成地址前缀的比特数。前缀是指定客户端网络掩码的另一种方式。前缀长度必须在正斜线（/）之后。
步骤 5	**default-router** *address* **示例:** `Device(dhcp-config)#` **default-router 10.10.10.1**	为 DHCP 客户端指定默认路由器的 IP 地址。
步骤 6	**option 150** *address* **示例:** `Device(dhcp-config)#` **option 150 10.10.10.1**	指定 TFTP 服务器的 IP 地址。
步骤 7	**option 125** *hex* **示例:** `Device(dhcp-config)#` **option 125 hex 0000.0009.0a05.08661.7574.6f69.6e73.7461.6c6c.5f64.686370**	指定 test 文件的路径，该文件描述了镜像文件的路径。

步骤 8	**copy tftp flash** *filename.txt* **示例:** `Device(config)#` **copy tftp flash image.bin**	上传 text 文件到设备上。
步骤 9	**copy tftp flash** *imagename.bin* **示例:** `Device(config)#` **copy tftp flash image.bin**	上传新镜像的 tar 文件到设备上。
步 骤 10	**exit** **示例:** `Device(dhcp-config)#` **exit**	返回全局配置模式。
步 骤 11	**tftp-server flash:** *config.text* **示例:** `Device(config)#` **tftp-server flash:config-boot.text**	指定 TFTP 服务器上的 Inspur INOS 配置文件。
步 骤 12	**tftp-server flash:** *imagename.bin* **示例:** `Device(config)#` **tftp-server flash:image.bin**	指定 TFTP 服务器上的镜像名称。
步 骤 13	**tftp-server flash:** *filename.txt* **示例:** `Device(config)#` **tftp-server flash:boot-config.text**	指定包含要下载的镜像文件名称的 text 文件。
步 骤 14	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitEthernet1/0/4**	指定要接收配置文件的客户端地址。
步 骤 15	**no switchport** **示例:** `Device(config-if)#` **no switchport**	把接口设置为三层模式。
步 骤 16	**ip address** *address mask* **示例:** `Device(config-if)#` **ip address 10.10.10.1 255.255.255.0**	为接口指定 IP 地址及掩码。
步 骤 17	**end** **示例:** `Device(config-if)#` **end**	返回特权 EXEC 模式。
步 骤 18	**copy running-config startup-config** **示例:** `Device(config-if)#` **end**	（可选）把配置的条目保存在配置文件中。

配置客户端从 DHCP 服务器下载文件

注释： 管理员应该只配置启用三层接口。不要指定 IP 地址或使用有保存配置的基于 DHCP 的自动配置。

总步骤

1. configure terminal

2. boot host dhcp

3. **boot host retry timeout** *timeout-value*
4. **banner config-save ^C** *warning-message* **^C**
5. **end**
6. **show boot**

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**boot host dhcp** **示例:** `Device(conf)#` **boot host dhcp**	启用有保存配置的自动配置。
步骤 3	**boot host retry timeout** *timeout-value* **示例:** `Device(conf)#` **boot host retry timeout 300**	（可选）设置系统尝试下载配置文件的总时间。 **注释：**如果不设置超时，系统会无限次地尝试从 DHCP 服务器获取 IP 地址。
步骤 4	**banner config-save ^C** *warning-message* **^C** **示例:** `Device(conf)#` **banner config-save ^C Caution -Saving Configuration Fileto NVRAM May Cause You to No longer Automatically Download Configuration Files at Reboot^C**	（可选）创建警告消息，在尝试把配置文件保存到 NVRAM 时显示。
步骤 5	**end** **示例:** `Device(config-if)#` **end**	返回特权 EXEC 模式。
步骤 6	**show boot** **示例:** `Device#` **show boot**	验证配置。

为多个 SVI 手动指定 IP 信息

此任务描述了如何手动给多个交换虚拟接口（SVI）指定 IP 信息。

总步骤

1. **configure terminal**
2. **interface vlan** *vlan-id*
3. **ip address** *ip-address subnet-mask*
4. **exit**
5. **ip default-gateway** *ip-address*
6. **end**

7. show interfaces vlan *vlan-id*

8. show ip redirects

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface vlan** *vlan-id* **示例:** `Device(config)#` **interface vlan 99**	进入接口配置模式，输入分配 IP 信息的 VLAN。范围从 1 到 4094。
步骤 3	**ip address** *ip-address subnet-mask* **示例:** `Device(config-vlan)#` **ip address10.10.10.2 255.255.255.0**	输入 IP 地址以及子网掩码。
步骤 4	**exit** **示例:** `Device(config-vlan)#` **exit**	返回全局配置模式。
步骤 5	**ip default-gateway** *ip-address* **示例:** `Device(config)#` **ip default-gateway10.10.10.1**	输入下一跳路由器接口的 IP 地址，该接口直接连接配置了默认网关的设备。默认网关会接收设备发来的带有未解析目的 IP 地址的 IP 数据包。 配置默认网关之后，设备就有了主机通信所需的远程网络连通性。 **注释：**当配置设备进行 IP 路由时，不需要设置默认网关。 **注释：**设备的 CAPWAP 依靠默认网关配置来支持加入设备的被路由接入点。
步骤 6	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 7	**show interfaces vlan** *vlan-id* **示例:** `Device#` **show interfaces vlan 99**	验证配置的 IP 地址。
步骤 8	**show ip redirects** **示例:** `Device#` **show ip redirects**	验证配置的默认网关。

修改设备设置配置

指定读写系统配置的文件名

默认情况下，Inspur INOS 软件使用 config.text 文件来读写系统配置的非易失性拷贝。然而可

以指定不同的文件名，在下一个引导循环加载使用。

在开始前

使用单独的交换机进行此任务。

总步骤

1. configure terminal

2. boot flash:*/file-url*

3. end

4. show boot

5. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**boot flash**:*/file-url* **示例:** `Device(config)#` **boot flash:config.text**	指定下一个引导循环要加载的配置文件。 *file-url*——路径（目录）以及配置文件名。文件名以及目录名区分大小写。
步骤 3	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 4	**show boot** **示例:** `Device#` **show boot**	验证配置的条目。 全局配置命令 **boot** 会更改环境变量 CONFIG_FILE 的设置。
步骤 5	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）把配置的条目保存到配置文件中。

手动引导交换机

默认时，交换机会自动引导启动，然而也可以配置交换机手动引导启动。

在开始前

使用单独的交换机进行此任务。

总步骤

1. configure terminal

2. boot manual

3. end

4. show boot

5. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**boot manual** **Example:** `Device(config)#` **boot manual**	使交换机在一下个引导循环中手动引导启动。
步骤 3	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 4	**show boot** **示例:** `Device#` **show boot**	验证配置的条目。 全局配置命令 **boot manual** 会更改环境变量 MANUAL_BOOT 的设置。下一次重启系统时，交换机会进入引导加载程序模式，如提示符 *switch:*所示。要引导系统，请使用 **boot***filesystem:/file-url*引导加载程序命令。 • *filesystem*:——使用flash:作为系统主板闪存设备。 `device:` **boot flash:** • *file-url*——指定可引导镜像的路径（目录）及名称。 文件名及目录名区分大小写。
步骤 5	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）把配置的条目保存到配置文件中。

引导安装模式中的设备

总步骤

1. cp *source_file_path destination_file_path*

2. software expand file *source_file_path*

3. reload

4. boot flash:packages.conf

5. show version

具体步骤

	命令或操作	目的
步骤 1	**cp** *source_file_path destination_file_path* **示例：** `Device#` **copy** **tftp://10.0.0.6/cat3k_caa-universalk9.SSA.03.12.**	进入全局配置模式。

	02.EZP.150-12.02.EZP.150-12.02.EZP.bin flash:	
步骤 2	**software expand file** *source_file_path* **示例：** 展开TFTP服务器的bin文件： Switch# **software expand file tftp://10.0.0.2/cat3k_caa-universalk9.SSA.03.09.37.EXP.150-9.37.EXP.bin to flash:** Preparing expand operation ... [1]: Downloading file tftp://10.0.0.2/cat3k_caa-universalk9.SSA.03.09.37.EXP.150-9.37.EXP.bin to active switch 1 [1]: Finished downloading file tftp://10.0.0.2/cat3k_caa-universalk9.SSA.03.09.37.EXP.150-9.37.EXP.bin to active switch 1 [1]: Copying software from active switch 1 to switch 2 [1]: Finished copying software to switch 2 [1 2]: Expanding bundle cat3k_caa-universalk9.SSA.03.09.37.EXP.150-9.37.EXP.bin [1 2]: Copying package files [1 2]: Package files copied [1 2]: Finished expanding bundle cat3k_caa-universalk9.SSA.03.09.37.EXP.150-9.37.EXP.bin 18 -rw- 74387812 Dec 7 2012 05:55:43 +00:00 cat3k_caa-base.SSA.03.09.37.EXP.pkg 19 -rw- 2738868 Dec 7 2012 05:55:44 +00:00 cat3k_caa-drivers.SSA.03.09.37.EXP.pkg 20 -rw- 32465772 Dec 7 2012 05:55:44 +00:00 cat3k_caa-infra.SSA.03.09.37.EXP.pkg 21 -rw- 30389036 Dec 7 2012 05:55:44 +00:00 cat3k_caa-INOSd-universalk9.SSA.150-9.37.EXP.pkg	使交换机在一下个引导循环中手动引导启动。

	22 -rw- 18342624 Dec 7 2012 05:55:44 +00:00 cat3k_caa-platform.SSA.03.09.37.EXP.pkg 23 -rw- 63374028 Dec 7 2012 05:55:44 +00:00 cat3k_caa-wcm.SSA.10.0.10.14.pkg 17 -rw- 1239 Dec 7 2012 05:56:29 +00:00 packages.conf	
步骤 3	**reload** **示例：** Device: **reload**	返回特权 EXEC 模式。
步骤 4	**boot flash:packages.conf** **示例：** switch: **boot flash:packages.conf**	验证配置的条目。 全局配置命令 **boot manual** 会更改环境变量 MANUAL_BOOT 的设置。 下一次重启系统时，交换机会进入引导加载程序模式，如提示符*switch:*所示。要引导系统，请使用**boot***filesystem:/file-url*引导加载程序命令。 • *filesystem*:——使用flash:作为系统主板闪存设备。 device: **boot flash:** • *file-url*——指定可引导镜像的路径（目录）及名称。 文件名及目录名区分大小写。
步骤 5	**show version** **示例：** switch# **show version** Switch Ports Model SW Version SW Image Mode ------ ----- ----- ---------- ---------- ---- 1 6 WS-C3850-6DS-S 03.09.26.EXP ct3850-ipservicesk9 INSTALL	（可选）把配置的条目保存到配置文件中。

在捆梆包模式中引导设备

有几种可以引导设备的方式——可以从 TFTP 服务器拷贝 bin 文件然后引导设备，也可以使用 **boot flash:<image.bin>**或 **boot usbflash0:<image.bin>**命令直接从闪存或 USB 闪存引导设备。

总步骤

1. cp *source_file_path destination_file_path*

2. switch:BOOT=<source path of .bin file>

3. boot

4. show version

具体步骤

	命令或操作	目的
步骤 1	**cp** *source_file_path destination_file_path* **示例：** `Device#` **copy tftp://10.0.0.6/cat3k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin flash:**	（可选）从 FTP 或 TFTP 服务器上把 bin 文件（image.bin）拷贝到闪存或 USB 闪存上。
步骤 2	**switch:BOOT=<source path of .bin file>** **示例：** `Device:` **switch:BOOT=tftp://10.0.0.2/cat3k_caa-universalk9.SSA.03.09.37.EXP.150-9.37.EXP.bin**	设置引导参数。
步骤 3	**boot** **示例：** `switch:` **boot**	引导设备。

步骤 4	**show version**
	示例：
	`switch#` **show version**
	`Switch Ports Model SW Version SW Image Mode`
	`------ ----- ----- ---------- ----------`
	`----`
	`1 6 WS-C3850-6DS-S 03.09.40.EXP`
	`ct3850-ipservicesk9 BUNDLE`
		验证设备在 **BUNDLE** 模式中。

在交换机堆栈上引导特定的软件镜像

默认情况下，交换机会使用 BOOT 环境变量中的信息自动引导启动系统。如果此变量未设置，交换机会对闪存文件系统进行递归的深度优先搜索，加载并执行第一个找到的可执行镜像。在深度优先搜索目录的过程中，继续搜索原始目录之前每个遇到的子目录都会被完全搜索。然而，也可以指定特定的镜像来进行引导启动。

总步骤

1. configure terminal

2. boot system switch {*number* | **all**}

3. end

4. show boot system

5. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：**	进入全局配置模式。

	`Device#` **configure terminal**	
步骤 2	**boot system switch** {*number* | **all**} **示例：** `Switch(config)#` **boot system switch 2 flash:cat3850-universalk9.SSA.03.08.83.EMD.150-8.83.EMD.b**	（可选）对于堆栈中的交换机，需指定下一个引导循环时要加载系统镜像的交换机成员： • 使用 *number* 指定一个堆栈成员（只能指定一个堆栈成员）。 • 使用 **all** 指定所有堆栈成员。 如果进入了一个 Inspur 3750-X 堆栈的 master 或成员，则只为其他 Inspur 3750-X 堆栈成员指定交换机镜像。 如果进入了一个 Inspur 3750-E 堆栈的 master 或成员，则只为其他 Inspur 3750-E 堆栈成员指定交换机镜像。 如果希望为 Inspur 3750 交换机指定镜像，请在 Inspur 3750 堆栈成员上输入此命令。
步骤 3	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 4	**show boot system** **示例：** `Device#` **show boot system**	验证配置的条目。 全局配置命令 **boot system** 会更改 BOOT 环境变量的设置。 在下一个引导循环中，交换机会尝使用 BOOT 环境变量中的信息自动引导启动系统。
步骤 5	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）把配置的条目保存到配置文件中。

配置计划的软件镜像重载

此任务描述了如何配置设备在之后重新加载软件镜像。

总步骤

1. configure terminal

2. copy running-config startup-config

3. reload in [*hh*:]*mm* [*text*]

4. reload at *hh*: *mm* [*month day* | *day month*] [*text*]

5. reload cancel

6. show reload

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：**	进入全局配置模式。

	Device# **configure terminal**	
步骤 2	**copy running-config startup-config** **示例：** **copy running-config startup-config**	在使用 **reload** 命令之前把设备的配置信息保存到启动配置中。
步骤 3	**reload in** [*hh*:]*mm* [*text*] **示例：** Device(config)# **reload in 12** System configuration has been modified. Save? [yes/no]: **y**	计划软件重新加载在指定的分钟或小时分钟以后生效。重新加载必须在约 24 日以内进行。可以使用一个含有至多 255 个字符的字符串说明重新加载的原因。
步骤 4	**reload at** *hh*: *mm* [*month day* | *day month*] [*text*] **示例：** Device(config)# **reload at 14:00**	按照小时分钟的方式指定重新加载的时间。 注释：仅在设置了设备时钟（通过网络时间协议（NTP）、硬件日历或手动设置）的情况下使用 **at** 关键字。此时间相对于设备上配置的时区。要计划多台设备同时进行重载，每台设备的时间必须使用 NTP 进行同步。
步骤 5	**reload cancel** **示例：** Device(config)# **reload cancel**	取消之前计划的重载。
步骤 6	**show reload** **示例：** **show reload**	显示之前计划的重载信息，或者显示设备上是否配置了重载计划。

监控设备设置配置

示例：验证设备的运行配置

```
Device# show running-config
Building configuration...
Current configuration: 1363 bytes
!
version 12.4
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Stack1
!
enable secret 5 $1$ej9.$DMUvAUnZOAmvmgqBEzIxE0
```

```
!
.
<output truncated>
.
interface gigabitethernet6/0/2
mvr type source
<output truncated>
...!
interface VLAN1
ip address 172.20.137.50 255.255.255.0
no ip directed-broadcast
!
ip default-gateway 172.20.137.1 !
!
snmp-server community private RW
snmp-server community public RO
snmp-server community private@es0 RW
snmp-server community public@es0 RO
snmp-server chassis-id 0x12
!
end
```

示例：显示安装模式中的软件引导过程

此示例展示了安装模式中的软件引导过程。

```
switch: boot flash:packages.conf
Getting rest of image
Reading full image into memory....done
Reading full base package into memory...: done = 74596432
Nova Bundle Image
--------------------------------------
Kernel Address : 0x6042f354
Kernel Size : 0x318412/3245074
Initramfs Address : 0x60747768
Initramfs Size : 0xdc08e8/14420200
Compression Format: .mzip
Bootable image at @ ram:0x6042f354
Bootable image segment 0 address range [0x81100000, 0x81b80000] is in range [0x80180000,
0x90000000].
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@boot
_system:
377
Loading Linux kernel with entry point 0x811060f0 ...
```

```
Bootloader: Done loading app on core_mask: 0xf
### Launching Linux Kernel (flags = 0x5)
All packages are Digitally Signed
Starting System Services
Nov 7 09:57:05 %INOSXE-1-PLATFORM: process stack-mgr: %STACKMGR-1-DISC_START: Switch 2 is
starting stack discovery
################################################################################
################################
Nov 7 09:59:07 %INOSXE-1-PLATFORM: process stack-mgr: %STACKMGR-1-DISC_DONE: Switch 2 has
finished stack discovery
Nov 7 09:59:07 %INOSXE-1-PLATFORM: process stack-mgr: %STACKMGR-1-SWITCH_ADDED: Switch 2
has been added to the stack
Nov 7 09:59:14 %INOSXE-1-PLATFORM: process stack-mgr: %STACKMGR-1-ACTIVE_ELECTED: Switch
2
has been elected ACTIVE
Restricted Rights Legend
Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.
inspur Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706
Inspur INOS Software, Inspur L3 Switch Software (CAT3K_CAA-UNIVERSALK9-M),
Version 03.09.12.EMD EARLY DEPLOYMENT ENGINEERING NOVA_WEEKLY BUILD, synced to
DSGS_PI2_POSTPC_FLO_DSBU7_NG3K_1105
Copyright (c) 1986-2012 by Inspur Systems,
Inc. Compiled Sun 04-Nov-12 22:53 by gereddy
License level to INOSd is ipservices
```

此示例显示了捆绑包模式中的软件引导过程：

```
switch: boot flash:cat3k_caa-universalk9.SSA.03.09.12.EMD.150-9.12.EMD.bin
Reading full image into
memory.....................................................................done
Nova Bundle Image
--------------------------------------
Kernel Address : 0x6042ff38
Kernel Size : 0x318412/3245074
Initramfs Address : 0x6074834c
Initramfs Size : 0xdc08e8/14420200
Compression Format: .mzip
Bootable image at @ ram:0x6042ff38
Bootable image segment 0 address range [0x81100000, 0x81b80000] is in range [0x80180000,
```

```
0x90000000].
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
File "flash:cat3k_caa-universalk9.SSA.03.09.12.EMD.150-9.12.EMD.bin" uncompressed and
installed, entry point: 0x811060f0
Loading Linux kernel with entry point 0x811060f0 ...
Bootloader: Done loading app on core_mask: 0xf
### Launching Linux Kernel (flags = 0x5)
All packages are Digitally Signed
Starting System Services
Nov 7 09:45:49 %INOSXE-1-PLATFORM: process stack-mgr: %STACKMGR-1-DISC_START: Switch 2 is
starting stack discovery
######################################################################################
#################################
Nov 7 09:47:50 %INOSXE-1-PLATFORM: process stack-mgr: %STACKMGR-1-DISC_DONE: Switch 2 has
finished stack discovery
Nov 7 09:47:50 %INOSXE-1-PLATFORM: process stack-mgr: %STACKMGR-1-SWITCH_ADDED: Switch 2
has been added to the stack
Nov 7 09:47:58 %INOSXE-1-PLATFORM: process stack-mgr: %STACKMGR-1-ACTIVE_ELECTED: Switch
2
has been elected ACTIVE
Restricted Rights Legend
Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.
inspur Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706
Inspur INOS Software, Inspur L3 Switch Software (CAT3K_CAA-UNIVERSALK9-
M), Version 03.09.12.EMD
EARLY DEPLOYMENT ENGINEERING NOVA_WEEKLY BUILD, synced to
DSGS_PI2_POSTPC_FLO_DSBU7_NG3K_1105
Copyright (c) 1986-2012 by Inspur Systems,
Inc. Compiled Sun 04-Nov-12 22:53 by gereddy
License level to INOSd is ipservices
```

示例：紧急安装

此示例展示了输入 **emergency-install** 引导命令时的示例输出：

switch: **emergency-install**

tftp://192.0.2.47/cat3k/cat3k_caa-universalk9.SSA.03.09.12.EMD.150-9.12.EMD.bin

```
The bootflash will be erased during install operation, continue (y/n)?y
Starting emergency recovery
(tftp://192.0.2.47/cat3k/cat3k_caa-universalk9.SSA.03.09.12.EMD.150-9.12.EMD.bin)...
Reading full image into memory......................done
Nova Bundle Image
--------------------------------------
Kernel Address : 0x6042e5cc
Kernel Size : 0x318261/3244641
Initramfs Address : 0x60746830
Initramfs Size : 0xdb0fb9/14356409
Compression Format: .mzip
Bootable image at @ ram:0x6042e5cc
Bootable image segment 0 address range [0x81100000, 0x81b80000] is in range [0x80180000,
0x90000000].
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
File "sda9:c3850-recovery.bin" uncompressed and installed, entry point: 0x811060f0
Loading Linux kernel with entry point 0x811060f0 ...
Bootloader: Done loading app on core_mask: 0xf
### Launching Linux Kernel (flags = 0x5)

Initiating Emergency Installation of bundle
tftp://172.19.211.47/cstohs/cat3k_caa-universalk9.SSA.03.09.12.EMD.150-9.12.EMD.bin
Downloading bundle
tftp://192.0.2.47/cat3k/cat3k_caa-universalk9.SSA.03.09.12.EMD.150-9.12.EMD.bin...
Validating bundle
tftp://192.0.2.47/cat3k/cat3k_caa-universalk9.SSA.03.09.12.EMD.150-9.12.EMD.bin...
Installing bundle
tftp://192.0.2.47/cat3k/cat3k_caa-universalk9.SSA.03.09.12.EMD.150-9.12.EMD.bin...
Verifying bundle
tftp://192.0.2.47/cat3k/cat3k_caa-universalk9.SSA.03.09.12.EMD.150-9.12.EMD.bin...
Package cat3k_caa-base.SSA.03.09.12.EMD.pkg is Digitally Signed
Package cat3k_caa-drivers.SSA.03.09.12.EMD.pkg is Digitally Signed
Package cat3k_caa-infra.SSA.03.09.12.EMD.pkg is Digitally Signed
Package cat3k_caa-INOSd-universalk9.SSA.150-9.12.EMD.pkg is Digitally Signed
Package cat3k_caa-platform.SSA.03.09.12.EMD.pkg is Digitally Signed
Package cat3k_caa-wcm.SSA.03.09.12.EMD.pkg is Digitally Signed
Preparing flash...
Syncing device...
Emergency Install successful... Rebooting
Restarting system.
Booting...(use DDR clock 667 MHz)Initializing and Testing RAM +++@@@@####...++@@++@@++@@++@
```

执行设备设置的配置示例

示例：把设备配置为 DHCP 服务器

```
Device# configure terminal
Device(config)# ip dhcp pool pool1
Device(dhcp-config)# network 10.10.10.0 255.255.255.0
Device(dhcp-config)# boot config-boot.text
Device(dhcp-config)# default-router 10.10.10.1
Device(dhcp-config)# option 150 10.10.10.1
Device(dhcp-config)# exit
Device(config)# tftp-server flash:config-boot.text
Device(config)# interface gigabitethernet1/0/4
Device(config-if)# no switchport
Device(config-if)# ip address 10.10.10.1 255.255.255.0
Device(config-if)# end
```

示例：配置 DHCP 自动镜像更新

此示例在 VLAN 99 上使用一个三层 SVI 接口，启用了使用保存配置的 DHCP 自动配置。

```
Device# configure terminal
Device(config)# boot host dhcp
Device(config)# boot host retry timeout 300
Device(config)# banner config-save ^C Caution - Saving Configuration File to NVRAM May Cause
You to No longer Automatically Download Configuration Files at Reboot^C
Device(config)# vlan 99
Device(config-vlan)# interface vlan 99
Device(config-if)# no shutdown
Device(config-if)# end
Device# show boot
BOOT path-list:
Config file: flash:/config.text
Private Config file: flash:/private-config.text
Enable Break: no
Manual Boot: no
HELPER path-list:
NVRAM/Config file
buffer size: 32768
Timeout for Config
Download: 300 seconds
Config Download
```

```
via DHCP: enabled (next boot: enabled)
Device#
```

示例：计划软件镜像重载

此示例展示了如何在当前日期的 7:30 pm 重新加载设备上的软件。

```
Device# reload at 19:30
Reload scheduled for 19:30:00 UTC Wed Jun 5 2013 (in 2 hours and 25 minutes)
Proceed with reload? [confirm]
```

此示例展示了如何在未来某个时间重新加载设备上的软件。

```
Device# reload at 02:00 jun 20
Reload scheduled for 02:00:00 UTC Thu Jun 20 2013 (in 344 hours and 53 minutes)
Proceed with reload? [confirm]
```

其他参考资料

相关文档

相关主题	文档标题
设备设置命令 启动引导程序命令	*系统管理命令参考（Inspur 6650 交换机）*
预下载特性	*系统管理配置指南*
INOS DHCP 配置	*IP 编址配置指南库（Inspur 6650 交换机）*
硬件安装	*Inspur 6650 交换机硬件安装指南*
平台无关的命令参考	*配置基础命令参考，Inspur INOS（Inspur 6650 交换机）*
平台无关的配置信息	*配置基础配置指南（Inspur 6650 交换机）* *IP 编址配置指南库（Inspur 6650 交换机）*

标准和 RFC

标准/RFC	标题
	无-

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS	http://www.icntnetworks.com

源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	

配置自治网络

自治网络

自治网络（Autonomic Networking）引入了自我管理的概念，让网络设备拥有智能，简化了网络运维人员对网络的管理操作。

自治网络的前提

- 自治网络基础设施特性仅支持以太网端口以及 IPv6 地址。
- 如果设备没有启动配置，默认所有接口都被启用，以交换邻接（adjacency discovery，AD）发现消息。
- 两台支持自治网络基础设施的邻接设备之间会自动构建一个自治控制层（Autonomic Control Plane，ACP）。两台设备的以太网接口都需要在 UP 状态，设备可以是未配置的（新上线的设备），也可以显式配置自治网络功能。
- 如果两台设备之间有非自治的二层云（如城域以太网服务），自治控制层也可以自动创建。这是通过自治设备上的通道发现（Channel Discovery，CD）协议实现的，该协议会探测使用的 VLAN 封装。
- 为了跨越之间的非自治三层设备来构建 ACP，需要在两个自治设备之间显式配置隧道，并在隧道上启用自动邻接发现。
- 要使自治网络基础设施特性能够工作，必须有自治注册设备（Autonomic Registrar），通常称为*注册设备*。网络中至少有一台设备必须被配置为注册设备，才能把新设备注册到自治域中。在所有需要的设备都注册进自治域的网络中不需要注册设备。
- 每个注册设备都仅支持一个自治域。仅在新自治设备加入域时才需要用到注册设备。
- 要联系注册设备来注册到自治域中，所有新设备必须与至少一台已经注册到域的设备有二层连通性。如果没有二层连通性，用户需要在设备之间配置隧道，并在隧道上配置自动邻接发现。
- 设备只能注册到一个自治域中。注册到不同域的两台设备之间不会创建自治控制层。
- 要进行零接触启动，必须不存在启动配置文件，且配置寄存器必须保持默认值，即

0x2102。

自治网络的限制

- 自治网络仅支持基于唯一设备标识符（unique device identifier，UDI）的设备。
- 自治网络以及零接触规划（Zero Touch Provisioning，ZTP）是不同的零接触方案。建议不要同时测试或使用自治网络与 ZTP。
- 自治网络中的所有设备都必须是连续自治的。如果无连续性，需要手动配置穿过非自治网络的隧道。
- 在 Inspur INOS 中，Inspur 3850 以及 Inspur 6650 交换机仅支持无标记的探针以及通道。
- 启用自治网络功能是，不应该手动禁用 IPv6 单播路由。
- Inspur 3850 以及 Inspur 6650 交换机不支持自治注册设备功能。

关于自治网络的信息

自治网络概述

自治网络的目标是创建自我管理的网络，克服 Internet 以及其他网络中快速增加的复杂性，并让这些网络能进一步增长。在一个自我管理的自治系统中，网络管理人员有了新的角色。管理员无需直接控制独立的网元，而是可以定义网络范围的策略和规则来指导自我管理过程。以下图例展示了自治网络的高层架构。

图 135：自治网络的高层架构

Simple Management Tools	简单的管理工具
Abstract Global Network View	抽象的全局网络试图
Autonomic Process	自治过程
Device OS	设备 OS
Autonomic interaction	自治交互
Traditional interactions(e.g. routing)	传统交互（如路由）

自治网络功能由运行在传统操作系统之上的独立软件实体控制。IP、开放最短路径优先（OSPF）等网络组件都包含在传统操作系统中，这些组件不会被改变，且不会感知到自治进程的存在。自治组件使用传统网络组件暴露的正常接口，与网络中的不同设备进行交互。自治组件之间能够安全地进行协作，这给设备增加了更多的智能，使得自治网络中的设备能在运维人员最少干预下自动进行配置、管理、保护以及自愈。它们也可以安全地整合自己的操作，向运维人员呈现简化且抽象的网络视图。

自治网络基础设施

自治网络基础设施特性简化了网络的引导功能，无需进行任何类型的预先配置，允许设备安全地加入自治域并在此后进行配置。自治网络基础设施特性的目标是让运维人员或网络管理系统能够安全地访问新的未配置的设备。这是通过以下步骤实现的：

1 定义并配置一台设备为注册设备。注册设备是第一个自治域设备。

2 网络管理员会收集要添加到网络的合法设备的标识符列表。此列表控制着添加到自治域的设备。设备由其唯一设备标识符（UDI）进行标识。列表会被编写为一个简单的文本

文件，一个 UDI 一行。此步骤是可选的，因为没有白名单的时候，所有设备都被允许加入域。白名单为允许的实体提供了特定的权限、服务、移动性以及访问或识别能力。列入白名单意味着允许访问。

3 已知设备的白名单会被作为注册设备配置的一部分上传到注册设备上。此步骤是可选的。

4 对于任何新的自治设备，直接连接到注册设备或其他已经注册到域中的设备时会自动从注册设备上接收域证书。

5 自治控制层会自动在自治域上建立，使得新设备可达。

自治网络基础设备的益处如下：

- 通过发现如何连通自治邻居来自动发现二层拓扑及连通性。
- 使用设备名称以及域证书可以安全且零接触地标识新设备身份。
- 虚拟的自治控制层让自治节点之间可以进行通信。

自治行为在新设备上默认被启用。要在现有设备上启用自治行为，使用 **autonomic** 命令。要禁用该行为，使用命令的 **no** 形式。

自治网络有如下组件：

- **注册设备**——特定企业域的注册中心，负责验证域中的新设备，向其提供域范围的凭证，并执行策略决定。基于预先加载的白名单，注册设备可以产生新设备是否可以加入特定域的策略决定。注册设备还有一个数据库，记录加入特定域的设备及设备详情。
- **通道发现**——用来发现跨越非自治二层网络的自治节点之间的可达性。
- **邻接发现**——用来发现自治邻居。邻接发现在三层进行。可以通过预先建立的三层通用被路由封装（Generic RoutedEncapsulation，GRE）隧道发现自治邻居。

新设备加入自治网络

下图说明了新设备是如何加入自治网络的。

图 136：新设备加入自治网络

New Device 1	新设备 1
Autonomic Proxy	自治代理
Registrar	注册设备

1 新设备给邻居发送 hello 消息。此例中，邻居是自治网络域的一部分。

2 hello 消息包含新设备的唯一设备标识符（UDI）。

3 自治设备作为代理，允许新设备加入自治网络域。自治网络设备会把自己的信息以及域信息通告给三层邻居。

4 在从邻居接收到自治网络 hello 消息并检测到 UDI 信息时，新设备被自治注册设备进行了验证。

5 新设备会在给所有邻居发送的 hello 消息中通告自己的域证书。邻居信息每 10 秒交换一次。

注释： 如果邻居消息改变，邻居的条目会被删除，且邻居发现过程重新开始。没有域证书时且设备使用 UDI 工作时，UDI 每 10 秒交换一次。

自治网络中的通道发现

设备启用自治网络功能时，通道发现会在所有接口上自动进行。自治网络特性在设备上默认被启用且无需配置（新设备且假设有自治网络功能），但会处于被动状态。它们不能接收以及应答通道发现（CD）探针。CD 探针是二层数据帧，只有拥有域证书的设备或注册到域中的设备可以在其所有启用的以太网接口上发送，由此，邻居可以动态被发现。探测过程会持

续进行，新加入的邻居可以被发现。

自治网络中的邻接发现

通道建立之后，代理会发送 ND（邻接发现）Hello 消息给新设备。已经注册到域中的设备可以作为新设备加入域的代理。新设备会发送 AN（自治网络）Hello 应答消息给中继。Hello 消息由新设备的 UDI 组成。收到新设备的 AN Hello 消息且检测到 UDI 信息后，AN 代理会把详细信息发送给 ANR（自治网络注册设备），以进行新设备的验证。

自治网络中的服务发现

自治网络使用组播域名系统（multicast Domain Name System，mDNS）基础设备来发现自治网络域中设备所需的多种服务。网络使用 mDNS 基础设施发现的服务有 AAA 服务器、配置服务器、syslog 服务器以及自治网络注册设备等。自治网络会监听域中所有设备的 mDNS 通告。自治网络会从承载服务的设备上发起 mDNS 通告。

自治控制层

当域中的新设备接收到域证书时，它会在 hello 消息中与邻居交换域证书。此行为在相同域中的两台自治设备之间创建了一个自治控制层。根据设备的能力不同，可以创建不同类型的自治控制层。自治控制层使用以下机制创建：

- 配置环回接口。
- 给环回接口动态分配 IPv6 地址。
- 配置自治 VPN 路由转发（VPN routing and forwarding，VRF）。

如何配置自治网络

配置注册设备

总步骤

1. enable

2. configure terminal

3. autonomic

4. autonomic registrar

5. domain-id *domain-name*

6. device-accept *udi*

7. whitelist *filename*

8. no shut

9. exit

10. exit

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。

步骤 3	**autonomic** **示例:** `Device# autonomic`	启用自治网络特性。
步骤 4	**autonomic registrar** **示例:** `Device(config)# autonomic registrar`	让设备成为注册设备，并进入注册设备配置模式。
步骤 5	**domain-id** *domain-name* **示例:** `Device(config-registrar)# domain-id abc.com`	表示向注册设备进行注册的所有设备共同属于的组。
步骤 6	**device-accept** *udi* **示例:** `Device(config-registrar)# device-accept PID:A901-12C-FT-D SN:CAT1902U88Y`	（可选）指定一个隔离设备的唯一设备标识（UDI），接受它加入自治域。 **注释：**配置注册设备无需使用此命令。只有配置注册设备接受之前被隔离的设备时才需要此命令。
步骤 7	**whitelist** *filename* **示例:** `Device(config-registrar)# whitelist flash:whitelist.txt`	（可选）允许加载本地设备上的文件，文件中包含特定域中被接受的设备列表。文件中每行必须包含一个 UDI 条目。 **注释：**如果不配置此命令，所有设备都被接受加入域。
步骤 8	**no shut** **示例:** `Device(config-registrar)# no shut`	启用自治注册设备。
步骤 9	**exit** **示例:** `Device(config-registrar)# exit`	退出注册设备配置模式，并返回全局配置模式。
步骤 10	**exit** **示例:** `Device(config)# exit`	退出全局配置模式，并返回特权 EXEC 模式。

验证并监控自治网络配置

总步骤

1. enable

2. show autonomic device

3. show autonomic neighbors [**detail**]

4. show autonomic control-plane [**detail**]

5. show autonomic l2-channels [**detail**]

6. show autonomic interfaces

7. debug autonomic {**Bootstrap** | **Channel-Discovery** | **Infra** | **Intent** | **Neighbor-Discovery** | **Registrar**| **Services** } {**aaa** | **all** | **ntp** | **events** | **packets**} {**info** | **moderate** | **severe**}

8. clear autonomic {**device** | **neighbor** *UDI* | **registrar accepted-device** *device UDI*}

具体步骤

步骤 1 **enable**

示例：

```
Device> enable
```

启用特权 EXEC 模式，在提示时输入密码。

步骤 2 **show autonomic device**

示例：

```
Device# show autonomic device
Status
Type
UDI
Device ID
Domain ID
Domain Certificate
SN:FOC1847X17A,cn=0021.d8d4.2900-1
Certificate Serial Number
Device Address
Domain Cert is Valid
Enabled
Autonomic Node
PID:WS-C6650-24U SN:FCW1934C05R
0021.d8d4.2900-1
icntnetworks.com
(sub:) ou=icntnetworks.com+serialNumber=PID:WSC6650-
24U-E
0F
FDF6:DBA2:13B6:0:21:D8D4:2900:1
```

显示自治设备的当前状态以及全局详细信息。

步骤 3 **show autonomic neighbors** [**detail**]

示例：

```
Device# show autonomic neighbors detail
UDI: "PID:WS-C6650-24U-E SN:FOC1847X17A"
Device ID
Domain ID
Address
State
Credential
Credential Validation
Last Validated Time
Certificate Expiry Date
Certificate Expire Countdown
Number of Links connected
```

```
Link:
Local Interface:
Remote Interface:
IP Address:
Uptime(Discovered Time):
Last Refreshed time:
0021.d8d4.2900-4
icntnetworks.com
FDF6:DBA2:13B6:0:21:D8D4:2900:4
Nbr inside the Domain
Domain Cert
Passed
2016-06-10 06:07:23 UTC
2017-06-08 14:54:09 UTC
31394668 (secs)
1
ANI2
ANI2
FE80::D66D:50FF:FEAD:2C83
00:30:35 ( 2016-06-10 05:39:06 UTC)
0 seconds ago
```

显示发现的邻居信息。

步骤 4 **show autonomic control-plane** [**detail**]

示例：

```
Device# show autonomic control-plane
VRF Name inspur_autonomic
Device Address FD08:2EEF:C2EE:0:E865:493B:ACFB:7
RPL floating-node, Dag-id = FD08:2EEF:C2EE:0:E865:493B:ACFB:5
Neighbor ACP Channel ACP Security
------------------------------------------------------------------------------
---
PID:WS-C3850-24U SN:FCW1934D05Z Tunnel100002 DIKE
Device# show autonomic control-plane detail
VRF Name inspur_autonomic
Device Address FD08:2EEF:C2EE:0:E865:493B:ACFB:7
RPL grounded-node, Dag-id = FD08:2EEF:C2EE:0:E865:493B:ACFB:1
Neighbor: PID:WS-C3850-24U SN:FCW1934D05Z
Uptime(Created Time): 00:12:16 ( 2016-07-15 05:38:53 UTC)
Supported ACP Channel: IPv6 GRE Tunnel
Negotiated ACP Channel: IPv6 GRE Tunnel
Tunnel Name Tunnel100000
Tunnel Source Interface ANI1
Tunnel Source FE80::5AAC:78FF:FE09:F383
Tunnel Destination FE80::3A20:56FF:FEF3:7158
```

```
Supported ACP Security: IPSec, DIKE
Negotiated ACP Security: DIKE
```

显示关于自治控制层的信息。

步骤 5 **show autonomic l2-channels [detail]**

示例：

```
Device# show autonomic l2-channels
AN L2 Channel Discovery Info :
Nbr UDI Encap Our Intf
State
Retry
------------------------------------------------------------------------------
---
PID:WS-C3850-24U SN:FCW1934D05Z 4018 Gi1/0/3 Active 1
Device# show autonomic l2-channels detail
AN L2 Channel Discovery Info :
------------------------------------------------------------------------------
---
Nbr UDI : PID:WS-C3850-24U SN:FCW1934D05Z
ANI Intf : ANI1
Encap : 0
Nbr Intf : GigabitEthernet1/0/3
Our Intf : GigabitEthernet1/0/3
Keepalives Missed : 0
Channel Status : Active
```

显示通道发现的结果。

步骤 6 **show autonomic interfaces**

示例：

```
Device# show autonomic interfaces
Interface Channel Disc AD Enabled Intf Type
------------------------------------------------------------------------------
---
GigabitEthernet0/0 None No L2 untagged If
GigabitEthernet1/0/1 None No L2 untagged If
GigabitEthernet1/0/2 None No L2 untagged If
GigabitEthernet1/0/3 Probing No L2 untagged If
GigabitEthernet1/0/4 None No L2 untagged If
GigabitEthernet1/0/5 None No L2 untagged If
GigabitEthernet1/0/6 None No L2 untagged If
GigabitEthernet1/0/7 None No L2 untagged If
GigabitEthernet1/0/8 None No L2 untagged If
GigabitEthernet1/0/9 None No L2 untagged If
GigabitEthernet1/0/10 None No L2 untagged If
GigabitEthernet1/0/11 None No L2 untagged If
GigabitEthernet1/0/12 None No L2 untagged If
```

```
GigabitEthernet1/0/13 None No L2 untagged If
GigabitEthernet1/0/14 None No L2 untagged If
GigabitEthernet1/0/15 None No L2 untagged If
GigabitEthernet1/0/16 None No L2 untagged If
GigabitEthernet1/0/17 None No L2 untagged If
GigabitEthernet1/0/18 None No L2 untagged If
GigabitEthernet1/0/19 None No L2 untagged If
GigabitEthernet1/0/20 None No L2 untagged If
GigabitEthernet1/0/21 None No L2 untagged If
GigabitEthernet1/0/22 None No L2 untagged If
GigabitEthernet1/0/23 None No L2 untagged If
GigabitEthernet1/0/24 None No L2 untagged If
GigabitEthernet1/1/1 None No L2 untagged If
GigabitEthernet1/1/2 None No L2 untagged If
TenGigabitEthernet1/1/3 None No L2 untagged If
TenGigabitEthernet1/1/4 None No L2 untagged If
Vlan1
ANI1
Loopback100000
Tunnel100002
None
None
None
None
No
Yes
No
No
Virtual If
Virtual If
Virtual If
Virtual If
```

显示自治域中的接口信息。

步骤7 **debug autonomic** {**Bootstrap** | **Channel-Discovery** | **Infra** | **Intent** | **Neighbor-Discovery** | **Registrar** | **Services**} {**aaa** | **all** | **ntp** | **events** | **packets**} {**info** | **moderate** | **severe**}

开启调试自治网络。

步骤 8 **clear autonomic** {**device** | **neighbor** *UDI* | **registrar accepted-device** *device UDI*}

清空或重置自治信息。

- **clear autonomic device** 命令会清空或重置所有设备特定的 AN 信息，包括在启动或称中获取的信息。
- **clear autonomic neighbor** 命令会清空在邻居发现过程中学习到的邻居相关的信息。
- **clear autonomic registrar accepted-device** 命令会清空注册设备储存的注册设备的公有密钥。

配置使用权许可证

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用 Inspur 特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问 Inspur 特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置使用权许可证的限制

使用及配置使用权许可证的限制如下：

- 可以订购并预先在交换机上激活 AP 计数许可证（AP-count license）。
- 基于镜像的许可证可以被升级。AP 计数许可证可以被取消激活，并可以在交换机和控制器之间移动。
- 要激活永久许可证，必须在配置新的镜像等级之后重启交换机。AP 计数许可证不要求重启就可以激活。
- 过期的基于镜像的评估许可证在重启之后不能被重新激活。
- 交换机堆栈的堆栈成员必须运行相同的许可证等级。
- 交换机堆栈不支持混用许可证。
- 交换机上会预先安装订购的镜像。如果没有预定镜像，则交换机默认使用 LAN Base 镜像引导。
- 附加 AP 计数许可证会在工厂中安装。

关于使用权许可证的信息

使用权许可证

使用权（Right-to-use，RTU）许可允许用户订购并激活特定的许可证类型和级别，并管理交换机上的许可证使用。可以订购的许可证类型有：

- 永久许可证——购买的许可证，包含特定特性集且没有过期日期。
- 评估许可证——交换机上预装的许可证，只有 90 天的使用有效期。

要激活永久许可证或评估许可证，用户需要接受最终用户许可协议（End-User License Agreement，EULA）。对于评估许可证，在 90 天时间到期之前会通知用户购买永久许可证或

者取消激活许可证。
永久许可证可以从一台设备移动到另一台设备上。要激活许可证，必须重启交换机。
评估许可证是交换机上的工厂镜像，不能转移到另一台交换机上。此类型的许可证在重启之后不能被重新激活。

基于镜像的使用权许可证

基于镜像的使用权许可证根据镜像许可不同支持不同的特性集：

- LAN Base——二层特性。
- IP Base——二层及三层特性。
- IP Services——二层、三层以及 IPv6 特性（只适用于交换机，不适用于无线控制器）。

默认的基于镜像的许可证是 LAN Base。

使用权许可证状态

在配置了特定的许可证类型和级别之后，可以监控许可证状态并管理许可证。

表 179：RTU 许可证状态

许可证状态	**描述**
激活，使用中（Active，In Use）	接受 EULA 且设备重启后许可证在使用。
激活，未使用（Active，Not In Use）	接受 EULA 且设备准备就绪，等待启用许可证。
未激活（Not Activated）	未接受 EULA。

监控基于镜像的许可证状态的指南如下：

- 只有在交换机重启之后购买的永久许可证才会被设置为 *Active，In Use* 状态。
- 如果购买了多个许可证，重启会激活特性集最高的许可证。比如，IP Services 许可证会被激活，而 LAN Base 许可证不被激活。
- 交换机重启后，购买的其余许可证会保持在 *Active，Not In Use* 状态。

注释： 对于 AP 计数许可证，要把状态更改为“Active，In Use”，必须首先确保取消激活了评估 AP 计数许可证。

交换机堆栈的许可证激活

交换机堆栈上支持使用权许可。一个交换机堆栈由至多九台支持堆栈的交换机组成，交换机之间通过 StackWise-160 端口连接。堆栈中只能连接一种类型的交换机。堆栈中有一台交换机是活跃交换机，其余交换机是备用交换机。活跃交换机是激活了 RTU 许可证的交换机，通过活跃交换机的控制台可以同时激活堆栈中成员交换机的对应许可证级别。

注释： 交换机堆栈不能混用交换机平台，也不能混用许可证级别。堆栈中的交换机必须是相同平台且使用相同许可证。

移动性控制器模式

交换机只有在移动性控制器（Mobility Controller，MC）模式中才会使用 AP 计数许可证。MC 会跟踪 AP 计数许可证，并决定是否允许接入点加入。
AP 计数许可证可以在移动性控制器模式的 CLI 中进行配置。

使用权 AP 计数许可

使用权许可（RTU）允许用户订购并激活特定的许可证类型，并管理设备上许可证的使用情况。
购买设备时可以选择支持特定数量的附加接入点计数许可证，但购买的许可证总数不应超过 25 个。也可以在收到设备之后再购买附加接入点计数许可证。
如果购买了 25 个新的附加许可证，用户可以把已经购买的这些附加许可证添加到设备上。许可证可以一个接一个依次添加，但是添加到设备上的许可证总数不应该超过 25 个。
用户可以通过配置交换机管理接入点计数许可证，并在 CLI 中查看当前使用的接入点总数。
以下是两种不同类型的接入点许可证：

1 接入点永久许可证
- 附加接入点计数许可证——可以在以后购买附加许可证，增加设备容量。可以把一台设备上的附加接入点计数许可证转移到另一台设备上。

2 接入点评估许可证
- 可以在购买许可证之前激活这些许可证，评估使用更多的接入点。
- 可以评估使用的最大接入点数量是 25 个。
- 接入点许可证的评估时间是 90 天。
- 可以在 CLI 中激活并取消激活评估许可证。

使用权 AP 计数评估许可证

如果考虑升级到支持接入点数量更多的许可证，可以在升级到永久许可证之前试用评估许可证。比如用户正在使用接入点数量为 50 的永久许可证，且希望尝试接入点数量为 100 的评估许可证，则可以试用评估许可证 90 天。评估许可证被激活以后，永久 AP 计数许可证会被忽略。
为了避免中断运行，评估许可证到期后设备不会更换许可证。在到期日的 5 天前开始，每天都会显示到期警告消息。在 90 天之后，评估许可证会过期，并显示警告消息。用户必须禁用评估许可证并购买永久许可证。
评估许可证过期后，设备重启时许可证会默认恢复使用永久许可证。

移植使用权附加 AP 计数许可证

从一台设备上撤销许可证并安装到另一台设备上被称为移植。用户可能希望通过移植许可证来更改设备的使用目的。比如用户可能希望把无线办公（Office Extend）或室内接入点的功能转移到不同的设备上，则可以把附加 AP 计数许可证从一台设备移植到另一台设备。
要移植许可证，用户必须在一台设备上取消激活附加 AP 计数许可证，并在另一台设备上激活同一个许可证。

评估许可证不能被移植。

如何配置使用权许可证

激活基于镜像的许可证

总步骤

1. license right-to-use activate{**ipbase**|**ipservices**| **lanbase**} {**all** | **evaluation all** } [**slot** *slot-number*] [**acceptEULA**]

2. reload[*LINE* | **at** | **cancel** | **in** | **slot** *stack-member-number* | **standby-cpu**]

3. show license right-to-use usage [**slot** *slot-number*]

具体步骤

	命令或操作	目的
步骤 1	**license right-to-use activate**{**ipbase**	**ipservices**	**lanbase**} {**all**	**evaluation all** }[**slot** *slot-number*] [**acceptEULA**] **示例：** `Device# license right-to-use activate ipservices all acceptEULA`	激活一类基于镜像的许可证。可以在所有交换机上进行激活，且可以配置接受 EULA。 **注释：**如果不接受 EULA，修改的配置在重启后不会生效。默认许可证（或没有取消激活的许可证）会在重启后变为活跃状态。

步骤 2	**reload** [*LINE*	**at**	**cancel**	**in**	**slot** *stack-member-number*	**standby-cpu**]
						示例：
						`Device# reload slot 1`
						`Proceed with reload? [confirm] y`
							重新加载特定的堆栈成员，完成 RTU 附加 AP 计数许可证的激活过程。
							注释：如果之前没有接受 EULA，会显示提醒要求接受 EULA。

步骤 3	**show license right-to-use usage** [**slot** *slot-number*]
	示例：
	`Device# show license right-to-use usage`
	`Slot# License Name Type`
	`usage-duration(y:m:d) In-Use EULA`
	`----------`
	`----------`
	`1 ipservices permanent 0 :10 :0 yes yes`
	`1 ipbase permanent 0 :0 :0 no no`
	`1 ipbase evaluation 0 :0 :0 no no`
	`1 lanbase permanent 0 :0 :7 no yes`
	`1 apcount evaluation 0 :0 :0 no no`
	`1 apcount base 0 :0 :0 no no`
	`1 apcount adder 0 :0 :0 no no`
	`Switch#`
		显示详细的使用信息。

激活 AP 计数许可证

总步骤

1. license right-to-use activate{**apcount***ap-number* **slot** *slot-num*} | **evaluation**} [**acceptEULA**]

2. show license right-to-use usage [**slot** *slot-number*]

具体步骤

	命令或操作	目的
步骤 1	**license right-to-use activate**{**apcount***ap-number* **slot** *slot-num*}	**evaluation**} [**acceptEULA**] **示例：** `Device#` **license right to use activate apcount 5 slot 1 acceptEULA**	激活一个或多个附加 AP 计数许可证，并立即接受 EULA。

步骤 2	**show license right-to-use usage** [**slot** *slot-number*]
	示例：
	`Device#` **show license right-to-use usage**
	`Slot# License Name Type`
	`usage-duration(y:m:d) In-Use EULA`
	`----------`
	`----------`
	`1 ipservices permanent 0 :3 :29 yes yes`
	`1 ipservices evaluation 0 :0 :0 no no`
	`1 ipbase permanent 0 :0 :0 no no`
	`1 ipbase evaluation 0 :0 :0 no no`
	`1 lanbase permanent 0 :0 :0 no no`
	`1 apcount evaluation 0 :3 :11 no no`
	`1 apcount base 0 :0 :0 no yes`
	`1 apcount adder 0 :0 :17 yes yes`
	`Switch#`
		显示详细的使用信息。

获取升级或容量附加许可证

可以使用容量附加许可证来增加设备支持的接入点数量。

总步骤

1. license right-to-use {**activate** | **deactivate**} **apcount**{*ap-number* | **evaluation** } **slot** *slot-num*[**acceptEULA**]

具体步骤

	命令或操作	目的
步骤 1	**license right-to-use** {**activate** | **deactivate**} **apcount**{*ap-number* | **evaluation** } **slot** *slot-num*[**acceptEULA**] **示例：** `Device#` **license right to use activate apcount 5 slot 2 acceptEULA**	激活一个或多个附加 AP 计数许可证，并立即接受 EULA。

移植许可证

要移植许可证，需要在一台设备上取消激活许可证并在另一台设备上激活同一个许可证。

总步骤

1. license right-to-use deactivate apcount*ap-number* **slot** *slot-num*[**acceptEULA**]

2. license right-to-use activate apcount*ap-number* **slot** *slot-num*[**acceptEULA**]

具体步骤

	命令或操作	目的
步骤 1	**license right-to-use deactivate apcount***ap-number* **slot** *slot-num*[**acceptEULA**] **示例：** `Device#` **license right to use deactivate apcount 1 slot 1 acceptEULA**	取消激活一台设备上的许可证。
步骤 2	**license right-to-use activate apcount***ap-number* **slot** *slot-num*[**acceptEULA**] **示例：** `Device#` **license right to use activate apcount 2 slot 2 acceptEULA**	在另一台设备上激活许可证。

监控及维护使用权许可证

命令	目的
show license right-to-use default	显示默认的许可证信息。
show license right-to-use detail	显示交换机堆栈中所有许可证的详细信息。
show license right-to-use eula {adder | evaluation| permanent}	显示最终用户许可证协议。
show license right-to-use mismatch	显示不匹配的许可证信息。
show license right-to-use slot *slot-number*	显示交换机堆栈特定插槽的许可证信息。
show license right-to-use summary	显示整个交换机堆栈的许可证信息汇总。
show license right-to-use usage [**slot**	显示交换机堆栈所有许可证使用情况的详

slot-number]	细信息。
show switch	显示交换机堆栈每个成员的详细信息以及许可证状态。

使用权许可证配置示例

示例：激活基于镜像的使用权许可证

此示例展示了如何在特定插槽上激活基于镜像的 IP Services 许可证并接受 EULA。

```
Switch# license right-to-use activate ipservices slot 1 acceptEULA
% switch-1:stack-mgr:Reboot the switch to invoke the highest activated License level
```

此示例展示了如何激活评估许可证：

```
Switch# license right-to-use activate ipservices evaluation acceptEULA
% switch-1:stack-mgr:Reboot the switch to invoke the highest activated License level
```

示例：显示使用权许可证信息

此示例展示了在交换机堆栈的活跃交换机上看到的整合的 RTU 许可证信息。堆栈的所有成员都有相同的许可证级别。激活评估 AP 计数许可证时，附加 AP 计数许可证会被忽略。进行评估时可以使用最大数量的 AP 计数许可证。

```
Switch# show license right-to-use summary
License Name Type Count Period left
--------------------------------------------------------
ipservices permanent 10 Lifetime
apcount evaluation 15 90
--------------------------------------------------------
License Level In Use: ipservices
License Level on Reboot: ipbase
Evaluation AP-Count: Enabled
Total AP Count Licenses: 25
AP Count Licenses In-use: 10
AP Count Licenses Remaining: 15
```

此示例展示了永久许可证以及附加许可证的汇总信息。评估 AP 计数许可证被禁用。示例中显示了交换机堆栈中激活的附加 AP 计数许可证总数。使用中的 AP 计数许可证（AP-count licenses in-use）表示连接的 AP 数量。

```
Switch# show license right-to-use summary
License Name Type Count Period left
------------------------------------------------------------
ipservices permanent N/A Lifetime
apcount base 0
```

```
apcount adder 25 Lifetime
-----------------------------------------------------------
License Level In Use: ipservices
License Level on Reboot: ipserviceseval
Evaluation AP-Count: Disabled
Total AP Count Licenses: 25
AP Count Licenses In-use: 10
AP Count Licenses Remaining: 15
```

此示例显示了默认的 RTU 许可证。默认许可证是预装的，不能被移除或转移。如果没有激活的许可证，交换机在重启后会使用默认许可证。

```
Switch# show license right-to-use default
Slot# License Name Type Count
-----------------------------------------------------
1 ipservices permanent N/A
1apcount base 0
1apcount adder 10
Slot# License Name Type Count
-----------------------------------------------------
2 ipservices permanent N/A
2 apcount base 0
2 apcount adder 10
Slot# License Name Type Count
-----------------------------------------------------
3 ipservices permanent N/A
3 apcount base 0
3 apcount adder 10
```

此示例展示了控制器上汇总的 RTU 许可证信息。激活评估 AP 技术许可证时，基本（base）的以及附加（adder）的 AP 计数许可证会被忽略。进行评估时可以使用最大数量的 AP 计数许可证。

```
controller# show license right-to-use summary
License Name Type Count Period left
-----------------------------------------------
apcount evaluation 25 Expired
-----------------------------------------------
Evaluation AP-Count: Enabled
Total AP Count Licenses: 25
AP Count Licenses In-use: 2
AP Count Licenses Remaining: 23
```

此示例展示了默认的 RTU 许可证。默认许可证是预装的，不能被移除或转移。如果没有激活的许可证，交换机在重启后会使用默认许可证。

```
controller# show license right-to-use default
Slot# License Name Type Count
```

```
-------------------------------------------------
1apcount base 10
```

示例：显示使用权许可证详情

此示例显示了 slot 1 上 RTU 许可证的详细信息。

```
Controller# show license right-to-use detail slot 1
Index 6: License Name: apcount
Period left: Expired
License Type: evaluation
License State: Active, In use
License Count: 1000
License Location: Slot 1
Index 7: License Name: apcount
Period left: Lifetime
License Type: base
License State: Active, Not In use
License Count: 0
License Location: Slot 1
Index 8: License Name: apcount
Period left: Lifetime
License Type: adder
License State: Not Activated
License Count: 0
License Location: Slot 1
```

示例：显示使用权许可证不匹配信息

此示例展示了堆栈中交换机的许可证信息，其中一个成员交换机为不匹配状态。成员许可证必须匹配活跃交换机的许可证。

```
Switch# show switch
Switch/Stack Mac Address : 6400.f125.0c80
H/W Current
Switch# Role Mac Address Priority Version State
-----------------------------------------------------------------------------
1 Standby 6400.f125.1b00 1 0 Ready
*2 Active 6400.f125.0c80 1 V01 Ready
3 Member 6400.f125.1780 1 0 Lic-Mismatch
```

注释： 要解决许可证不匹配问题，请首先检查 RTU 许可证汇总信息：

```
Switch# show switch right-to-use summary
```

随后更改不匹配交换机的许可证级别，使其与活跃交换机相同。此示例展示了为成员交换机激活 IP Base 许可证，以匹配活跃交换机。

示例：显示使用权许可证使用情况

此示例展示了控制器上详细的许可证使用情况：

```
Controller# show license right-to-use usage
Slot# License Name Type usage-duration(y:m:d) In-Use EULA
-------------------------------------------------------------------------
1 apcount evaluation 0 :3 :3 yes yes
1 apcount base 0 :0 :0 no yes
1 apcount adder 0 :0 :0 no no
```

其他参考资料

相关文档

相关主题	文档标题
RTU 命令	*系统管理命令参考（Inspur 6650 交换机）*
RTU AP 镜像预加载特性	*系统管理命令参考（Inspur WLC 5700 系列）*

标准和 RFC

标准/RFC	标题
无	-

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	http://www.icntnetworks.com

使用权许可证的特性历史与信息

版本	特性信息
Inspur INOS 11.3.1	引入了此特性。

配置管理员用户名及密码

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用 Inspur 特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问 Inspur 特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于配置管理员用户名及密码的信息

可以配置管理员用户名及密码，防止未授权的用户重新配置设备以及查看配置信息。本节提供了初始配置及密码恢复的说明。

用户也可以设置管理员用户名及密码来管理并配置与设备关联的接入点。

强密码

管理员在管理接入点时，可以使用强管理员密码，如通过 ASCII 密钥加密的密码。

按照以下说明创建强密码：

- 密码应包含以下几类字符中至少三类——小写字母、大写字母、数字以及特殊字符。
  注释： 特殊字符不支持在 GUI 登录的用户名或密码中使用。
- 新密码不能与关联的用户名相同，且密码中不应该颠倒使用用户名。
- 密码中的字符不能连续重复三次以上。
- 密码不能是 **inspur**、**rupsni**、**admin**、**nimda**，也不能是通过更改其中字母的大小写，把“i”替换成“1”、“|”或“!”，把“o”替换成“0”或把“s”替换成“$”等方式得到的变体。
- 用户名和密码可接受的最大字符数量是 32 个。

加密密码

可以为密码设置三种类型的密钥：

- 随机生成的密钥——此密钥是随机生成的，这是最安全的选项。要把配置文件从一个系统导出到另一个系统时，此密钥也应该被导出。
- 静态密钥——最简单的选项是使用固定（静态）加密密钥。使用固定加密密钥不需要进行密钥管理，但是如果密钥被别人发现了，数据就可能被知道密钥的人解密。此选项不安全，且在 CLI 中被成为混淆（obfuscation）方式。
- 用户定义的密钥——用户可以自己定义密钥。要把配置文件从一个系统导出到另一个系统时，两个系统应该都配置了相同的密钥。

配置管理员用户名及密码

总步骤

1. configure terminal

2. username *admin-username* **password {0** *unencrypted_password* **| 7** *hidden_password* **|** *unencrypted_text***}**

3. username *admin-username* **secret {0** *unencrypted_secret_text* **| 4** *SHA256 encrypted_secret_text* **| 5***MD5 encrypted_secret_text* **| LINE}**

4. ap mgmtuser username *username* **password {0** *unencrypted password* **| 8** *AES encrypted password***}secret {0** *unencrypted password* **| 8** *AES encrypted password* **}**

5. ap dot1x username *username* **password {0** *unencrypted password* **| 8** *AES encrypted password* **}**

6. end

7. ap name apname mgmtuser username *username***password** *password* **secret** *secret _text*

8. ap name apname dot1x-user *username* **password** *password*

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**username** *admin-username* **password {0***unencrypted_password* **| 7** *hidden_password* **|***unencrypted_text***}** **示例：** `Device(config)#` **username adminuser1 password 0QZsek239@**	指定管理员的用户名以及密码。 管理员可以配置设备并查看配置的信息。
步骤 3	**username** *admin-username* **secret {0***unencrypted_secret_text* **| 4** *SHA256 encrypted_secret_text***| 5** *MD5 encrypted_secret_text* **| LINE}** **示例：** `Device(config)#` **username adminuser1 secret 0QZsek239@**	指定管理员口令。
步骤 4	**ap mgmtuser username** *username* **password {0***unencrypted password* **| 8** *AES encrypted password* **}secret{0** *unencrypted password* **| 8** *AES encrypted password* **}** **示例：** `Device(config)#` **ap mgmtuser username inspurpassword 0 Qwci12@ secret 0 Qwci14@!**	指定管理员用于管理设备上配置的所有接入点的用户名及密码。 也可以包含口令来执行特权接入点管理。 **注释：**如果密码不足以满足强密码策略，密码会被拒绝并显示验证错误消息。比如，以下密码会被拒绝，因为它不是强密码。 `Device#` **ap mgmtuser username inspur password 0 abcd secret 0**

		1234
步骤 5	**ap dot1x username** *username* **password {0** *unencryptedpassword* **| 8** *AES encrypted password* **}** **示例：** `Device(config)#` **ap dot1x username inspur password** **0 Qwci12@**	指定用于管理设备配置的所有接入点的 802.1X 用户名及密码。
步骤 6	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。此外，可以按下 **Ctrl-Z** 退出全局配置模式。
步骤 7	**ap name apnamemgmtuser username** *username***password***password* **secret** *secret_text* **示例：** `Device#` **ap name APf0f7.55c7.7b23 mgmtuser username** **inspur password Qne35! secret Nzep592$**	配置管理员用户名、密码以及口令，管理设备上配置的指定接入点。
步骤 8	**ap name apname dot1x-user** *username* **password** *password* **示例：** `Device#` **ap name APf0f7.55c7.7b23 dot1x-user username inspur password Qne35!**	配置指定接入点的 802.1X 用户名及密码。

其他参考资料

相关文档

相关主题	**文档标题**
系统管理命令	系统管理命令参考指南

标准和 RFC

标准/RFC	**标题**
无	-

技术助手

描述	**链接**
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS	http://www.icntnetworks.com

源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	

执行管理员用户名及密码配置的特性历史与信息

版本	特性信息
Inspur INOS 11.3.1	引入了此特性。

配置应用程序可见性和控制

应用程序可见性和控制（Application Visibility and Control，AVC）解决方案为 Inspur 网络设备提供应用级别的分类、监控以及流量控制功能，用来提升关键业务性能，促进容量管理及规划，并减少网络运维开销。Inspur AVC 解决方案在分支机构路由器、聚合服务路由器以及 Inspur 交换机中提供。
关于 Inspur 交换机上的 AVC 信息，参见在*有线网络中配置应用程序可见性和控制*。

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看自己使用的平台及软件版本的发布注释。
用户可以使用 Inspur 特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问 Inspur 特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于有线网络中应用程序可见性和控制的信息

Inspur 致力于把分支机构和园区的解决方案从严格基于数据包和连接的方式改进为应用感知和应用智能的，而应用程序可见性和控制（AVC）是努力中的关键部分。应用程序可见性和控制方案能够使用深度包检测技术，利用基于网络的应用识别（Network-Based Application Recognition，NBAR2）引擎来进行应用分类。从 Inspur INOS 11.3.1 开始，单独交换机以及交

换机堆栈的有线接入端口已经支持了 AVC。NBAR2 可以通过在接口上启用协议发现功能显式地激活，也可以通过配置包含 **match protocol** 分类器的 QoS 策略隐式地激活。有线接入端口支持基本的有线 AVC FNF。有线 AVC FNF 提供了每个端口的客户端、服务器以及应用程序统计数据。记录的结果与 ezPM**application-statistics** 和 **application-perforrnance** 中的 **application-client-server-stats** 流量监控效果相似。

支持的 AVC 类映射及策略映射格式

支持的 AVC 类映射格式

类映射格式	类映射示例	方向
match protocol *protocol name*	**class-map match-any NBAR-VOICE** **match protocol ms-lync-audio**	入向和出向
组合过滤器	**class-map match-any NBAR-VOICE** **match protocol ms-lync-audio** **match dscp 45**	仅出向

支持的 AVC 策略格式

策略格式	QoS 行为
基于匹配协议过滤器的出向策略	标记并限速
基于匹配协议过滤器的入向策略	标记并限速

下表描述了详细的 AVC 策略格式及其示例：

AVC 策略格式	AVC 策略示例	方向
基本设置	**policy-map MARKING-IN** **class NBAR-MM_CONFERENCING** **set dscp af41**	入向和出向
基本限速	**policy-map POLICING-IN** **class NBAR-MM_CONFERENCING** **police cir 600000** **set dscp af41**	入向和出向
基本设置及限速	**policy-map webex-policy** **class webex-class** **set dscp ef** **cos police 5000000**	入向和出向
多项设置以及限速，包含默认策略	**policy-map webex-policy** **class webex-class** **set dscp af31** **cos police 4000000** **class class-webex-category** **set dscp ef** **cos police 6000000** **class class-default** **set dscp <>**	入向和出向

层级化限速	**policy-map webex-policy** **class webex-class** **police 5000000** **service-policy client-in-police-only** **policy-map client-in-police-only** **class webex-class** **police 100000** **class class-webex-category** **set dscp ef** **cos police 200000**	入向和出向
层级化设置及限速	**policy-map webex-policy** **class class-default** **police 1500000** **service policy client-up-child** **policy-map webex-policy** **class webex-class** **police 100000** **set dscp ef** **class class-webex-category** **police 200000** **set dscp af31**	

有线网络应用程序可见性和控制的限制

- 基于 NBAR 的 QoS 策略配置只能在有线物理端口上进行。策略配置不支持在虚拟接口上进行，如 VLAN、端口通道以及其他逻辑接口。
- 基于 NBAR2 的匹配条件 **match protocol** 只允许使用标记和限速行为。
- NBAR2 匹配条件不支持在配置了队列特性的策略中使用。
- “Match Protocol”：所有策略中同时只能支持最多 255 个不同的协议（8 位硬件限制）。
- 不支持使用基于 QoS 的 NBAR2 属性（**match protocol** 属性）。
- 管理端口（Gig 0/0）上不支持 AVC。
- 不支持 IPv6 数据包分类。
- 仅支持 IPv4 单播（TCP/UDP）。
- Web UI：可以配置应用可见性并通过 Web UI 执行应用监控。应用控制只能使用 CLI 进行，不支持通过 Web UI 进行。
- 不能在一台交换机上同时配置 NBAR 和 ACL 记录。
- 协议发现、基于应用的 QoS 以及有线 AVC FNF 不能同时与不基于应用的 FNF 配置在相同的接口上。然而，这些有线 AVC 特性可以同时配置。比如，协议发现、基于应用的 QoS 以及有线 AVC FNF 可以同时配置在相同接口上。
- 有线 AVC FNF 支持单个预定义的记录。
- 策略配置只能在二层物理端口（接入/中继）或三层端口上进行。上行链路也可以配置策略，只要链路是单个上行链路而不属于端口通道。
- 性能：每个交换机成员可以处理 500 连接每秒（connections per second，CPS），且 CPU

利用率低于 50%。

- 规模：每 48 个接入端口可以处理至多 10000 个双向数据流，每 24 个接入端口 5000 个双向数据流（约 200 数据流每接入端口）。

如何配置应用程序可见性和控制

在有线网络中配置应用程序可见性和控制

按照以下步骤在有线端口上配置应用程序可见性和控制：

配置可见性：

- 在接口配置模式中使用**ip nbarprotocol-discovery**命令，启用协议发现，激活NBAR2引擎。参见在接口上启用应用识别。

配置控制：基于应用配置 QoS 策略

1 创建 AVC QoS 策略。参见创建 AVC QoS 策略。

2 把 AVC QoS 策略应用到接口。参见应用 QoS 策略到交换机端口。

配置基于应用的灵活 Netflow（FNF）：

- 指定流的关键字段和非关键字段，创建数据流记录。参见创建流记录。
- 创建流导出器以导出流记录。参见创建流导出器。
- 创建基于流记录以及流导出器的流监控器。参见创建流监控器。
- 把流监控器部署到接口上。参见配置接口应用流监控器。

协议发现、基于应用的 QoS 以及基于应用的 FNF 都是独立的特性。它们可以独立配置，也可以同时配置在相同接口上。

在接口上启用应用识别

按照以下步骤在接口上启用应用识别。

总步骤

1. configure terminal

2. interface *interface-id*

3. ip nbar protocol-discovery

4. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例：** `Device(config)#` **interface gigabitethernet 1/0/1**	指定要启用协议发现的接口，并进入接口配置模式。
步骤 3	**ip nbar protocol-discovery** **示例：** `Device(config-if)#` **ip nbar protocol-discovery**	在接口上启用应用识别，激活 NBAR2 引擎。
步骤 4	**end** **示例：**	返回特权 EXEC 模式。

	`Device(config-if)#` **end**	

创建 AVC QoS 策略

执行以下步骤创建 AVC QoS 策略：

1 创建类映射，使用匹配协议过滤器。

2 创建策略映射。

3 把策略映射应用到接口。

创建类映射

在配置匹配协议过滤器之前需要创建类映射。可以向流量应用标记及限速等 QoS 行为。AVC 匹配协议过滤器会被应用到有线接入端口上。有关受支持协议的更多信息，参见 http://www.icntnetworks.com

总步骤

1. configure terminal

2. class-map *class-map-name*

3. match protocol *application-name*

4. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**class-map** *class-map-name* **示例：** `Device(config)#` **class-map webex-class**	创建类映射。
步骤 3	**match protocol** *application-name* **示例：** `Device(config)#` **class-map webex-class** `Device(config-cmap)#` **match protocol webex-media**	指定匹配应用名称。
步骤 4	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。此外，也可以按下 **Ctrl-Z** 退出全局配置模式。

创建策略映射

总步骤

1. configure terminal

2. policy-map *policy-map-name*

3. class [*class-map-name* | **class-default**]

4. police *rate-bps burst-byte*

5. set {**dscp** *new-dscp* | **cos** *cos-value*}

6. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**policy-map***policy-map-name* **示例：** `Device(config)#` **policy-map webex-policy**	输入策略映射名称，创建策略映射，并进入策略映射配置模式。 默认情况下，没有定义的策略映射。 策略映射的默认行为是在数据包是 IP 数据包时把 DSCP 设置为 0，数据包有标签时把 CoS 设置为 0。默认不执行限速。 **注释：**要删除现有的策略映射，使用全局配置命令**no policy-map***policy-map-name*。
步骤 3	**class** [*class-map-name* | **class-default**] **示例：** `Device(config-pmap)#` **class webex-class**	定义流量分类，进入策略映射类配置模式。 默认情况下，没有定义的策略映射与类映射。 如果已经使用全局配置命令 **class-map** 定义了流量类，请在 *class-map-name* 字段中指出流量类名称。 流量类 **class-default** 是预定义的，可以添加到任意策略中。该类总是被放置在策略映射的末尾。**class-default** 类中隐含 **match any**，所有没有匹配其他流量类的数据包都会匹配 **class-default**。 **注释：**要删除现有的流量类，使用 **no class** *class-map-name* 策略映射配置命令。
步骤 4	**police** *rate-bps burst-byte* **示例：** `Device(config-pmap-c)#` **police100000 80000**	定义分类流量的限速器。 默认情况下，无定义的限速器。 • 使用 *rate-bps* 指定平均流量速率，单位比特每秒（b/s）。范围从 8000 到 10000000000。 • 使用 *burst-byte* 指定正常突发大小，单位是字节。范围从 8000 到 1000000。
步骤 5	**set** {**dscp** *new-dscp* | **cos** *cos-value*} **示例：** `Device(config-pmap-c)#` **set dscp45**	分类 IP 流量，在数据包中设置新值： • **dscp** *new-dscp*,——输入要指定给分类后流量的新 DSCP 值，范围从 0 到 64。
步骤 6	**end** **示例：**	返回特权 EXEC 模式。此外，也可以按下 **Ctrl-Z** 退出全局配置模式。

	Device(config)# **end**	

应用 QoS 策略到交换机端口

总步骤

1. configure terminal

2. interface *interface-id*

3. service-policy input *policymapname*

4. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例：** Device(config)# **interface Gigabitethernet 1/0/1**	进入接口配置模式。
步骤 3	**service-policy input** *policymapname* **示例：** Device(config-if)# **service-policy input MARKING_IN**	应用本地策略到接口。
步骤 4	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式。此外，也可以按下 **Ctrl-Z** 退出全局配置模式。

配置有线 AVC 灵活 Netflow

创建流记录

可以配置单个流记录并将其与流监控器关联。

总步骤

1. configure terminal

2. flow record *flow_record_name*

3. description *description*

4. match ipv4 version

5. match ipv4 protocol

6. match application name

7. match connection client ipv4 address

8. match connection server ipv4 address

9. match connection server transport port

10. match flow observation point

11. collect flow direction

12. collect connection initiator

13. collect connection client counter packets long

14. collect connection client counter bytes network long

15. collect connection server counter packets long

16. collect connection server counter bytes network long

17. collect timestamp absolute first

18. collect timestamp absolute last

19. collect connection new-connections

20. end

21. show flow record

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** 示例： `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**flow record** *flow_record_name* 示例： `Device(config)#` **flow record** `flow-record-1`	进入流记录配置模式。
步骤 3	**description** *description* 示例： `Device(config-flow-record)#` **description**`flow-record-1`	（可选）创建流记录描述。
步骤 4	**match ipv4 version** 示例： `Device (config-flow-record)#` **match ipv4version**	指定匹配 IPv4 报头中的 IP 版本。
步骤 5	**match ipv4 protocol** 示例： `Device (config-flow-record)#` **match ipv4protocol**	指定匹配 IPv4 协议。
步骤 6	**match application name** 示例： `Device (config-flow-record)#` **match applicationname**	指定匹配应用名称。 注释：要配置 AVC 支持，此步骤是强制进行的。这让流可以匹配应用。
步骤 7	**match connection client ipv4 address** 示例： `Device (config-flow-record)#` **match connectionclient ipv4 address**	指定匹配客户端的 IPv4 地址（流发起者）。
步骤 8	**match connection server ipv4 address** 示例： `Device (config-flow-record)#` **match connectionserver ipv4 address**	指定匹配服务器的 IPv4 地址（流响应者）。
步骤 9	**match connection server transport port** 示例： `Device (config-flow-record)#` **match**	指定匹配服务器的传输端口。

	connectionserver transport port	
步骤 10	**match flow observation point** **示例：** `Device (config-flow-record)#` **match flowobservation point**	指定匹配流观察点的观察点 ID。
步骤 11	**collect flow direction** **示例：** `Device (config-flow-record)#` **collect flowdirection**	对于下一步中 **collect connection initiator** 命令 **initiator** 关键字指定的双向数据流，指定收集某一端（发起者或响应者）的某一方向（入向或出向）流量。 根据 **initiator** 关键字指定的值，**flow direction** 关键字可以采用以下值： • 0x01=入向数据流 • 0x02 =出向数据流 **initiator** 关键字设置为 initiator 时，流方向指定为流的发起者一端。**initiator** 关键字设置为 responder 时，流方向指定为流的响应者一端。对于有线 AVC，**initiator** 关键字总是被设置为 initiator。
步骤 12	**collect connection initiator** **示例：** `Device (config-flow-record)#` **collectconnection initiator**	指定收集 **collect flow direction** 命令指定的流方向的某一端（发起者或响应者）流量。关键字 **initiator** 提供了关于流方向的以下信息： • 0x01=发起者——数据流的源是连接的发起者 对于有线 AVC，**initiator** 关键字总是被设置为 initiator。
步骤 13	**collect connection client counter packets long** **示例：** `Device (config-flow-record)#` **collectconnection client counter packets long**	指定收集客户端发送的数据包数量。
步骤 14	**collect connection client counter bytes network long** **示例：** `Device (config-flow-record)#` **collectconnection client counter bytes network long**	指定收集客户端传输的总字节数。
步骤 15	**collect connection server counter packets long** **示例：** `Device (config-flow-record)#` **collectconnection server counter packets long**	指定收集服务器发送的数据包数量。
步骤 16	**collect connectionserver**	指定收集服务器传输的总字节

	counterbytesnetworklong **示例：** `Device (config-flow-record)#` **collectconnection server counter bytes network long**	数。
步骤 17	**collect timestamp absolute first** **示例：** `Device (config-flow-record)#` **collect timestampabsolute first**	以毫秒形式收集收到第一个包的时间。
步骤 18	**collect timestamp absolute last** **示例：** `Device (config-flow-record)#` **collect timestampabsolute last**	以毫秒形式收集收到最新包的时间。
步骤 19	**collect connection new-connections** **示例：** `Device (config-flow-record)#` **collectconnection new-connections**	收集发现的发起连接的数量。
步骤 20	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。此外，也可以按下 **Ctrl-Z** 退出全局配置模式。
步骤 21	**show flow record** **示例：** `Device #` **show flow record**	显示所有流记录的信息。

创建流导出器

可以创建流导出器，导出流的参数。

总步骤

1. **configure terminal**
2. **flow exporter** *flow_exporter_name*
3. **description** *description*
4. **destination** { *hostname* | *ipv4-address* | *ipv6-address* }
5. **option application-table** [**timeout** *seconds*]
6. **end**
7. **show flow exporter**
8. **show flow exporter statistics**

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**flow exporter** *flow_exporter_name* **示例：**	进入流导出器配置模式。

	Device(config)# **flow exporter** flow-exporter-1	
步骤 3	**description** *description* **示例：** Device(config-flow-exporter)# **description** flow-exporter-1	（可选）为流导出器创建描述。
步骤 4	**destination** { *hostname* | *ipv4-address* | *ipv6-address* } **示例：** Device (config-flow-exporter)# **destination 10.10.1.1**	指定导出器要发送数据的主机名以及系统的 IPv4 或 IPv6 地址。
步骤 5	**option application-table** [**timeout** *seconds*] **示例：** Device (config-flow-exporter)# **optionapplication-table timeout 500**	（可选）为流导出器配置应用表选项。**timeout** 选项配置流导出器的重发时间，单位是秒。合法的范围从 1 到 86400 秒。
步骤 6	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式。此外，也可以按下 **Ctrl-Z** 退出全局配置模式。
步骤 7	**show flow exporter** **示例：** Device # **show flow exporter**	现有所有流导出器的信息。
步骤 8	**show flow exporter statistics** **示例：** Device # **show flow exporter statistics**	显示流导出器的统计信息。

创建流监控器

可以创建流监控器，并与流记录关联。

总步骤

1. **configure terminal**
2. **flow monitor** *monitor-name*
3. **description** *description*
4. **record** *record-name*
5. **exporter** *exporter-name*
6. **cache type normal** { **timeout** {**active** | **inactive**} | **type normal** }
7. **end**
8. **show flow monitor**

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式。
步骤 2	**flow monitor** *monitor-name*	创建流监控器，并进入流监控器配置

	示例： `Device (config)# flow monitor flow-monitor-1`	模式。
步骤 3	**description** *description* **示例：** `Device (config-flow-monitor)# description flow-monitor-1`	（可选）为流监控器创建描述。
步骤 4	**record** *record-name* **示例：** `Device (config-flow-monitor)# record flow-record-1`	指定之前创建的流记录的名称。
步骤 5	**exporter** *exporter-name* **示例：** `Device (config-flow-monitor)# exporter flow-exporter-1`	指定之前创建的流导出器的名称。

步骤 6	**cache type normal** { **timeout** {**active** | **inactive**} | **type normal** }
	示例：
	`Device (config-flow-monitor)# cache timeout active 1800`
	示例：
	`Device (config-flow-monitor)# cache timeout inactive 200`
	示例：
	`Device (config-flow-monitor)# cache type normal`
		（可选）配置流缓存参数。
		注释：仅支持正常的缓存类型。不支持配置缓存大小。缓存恒定大小预定义为 10000。

步骤 7	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式。此外，也可以按下 **Ctrl-Z** 退出全局配置模式。
步骤 8	**show flow monitor** **示例：** `Device # show flow monitor`	显示所有流监控器的信息。 注释：有线AVC不支持**show flow monitor** *flow-monitor-name* **statistics**和**show flow monitor** *flow-monitor-name* **cache**命令。这些命令不会显示任何针对于有线AVC的信息。**show flow exporter statistics**命令可以作为**show flowmonitor** *flow-monitor-name* **cache** 命令的替代命令使用，显示流监控器缓存的统计信息。

配置接口应用流监控器

总步骤

1. configure terminal

2. interface *interface-id*

3. ip flow monitor *monitor-name* { **input** | **output** }

4. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例：** `Device(config)#` **interface Gigabitethernet 1/0/1**	进入接口配置模式。
步骤 3	**ip flow monitor** *monitor-name* { **input**	**output** } **示例：** `Device (config-if) #` **ip flow monitorflow-monitor-1 input**	把流监控器与接口的输入或输出数据包关联。
步骤 4	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。此外，也可以按下 **Ctrl-Z** 退出全局配置模式。

NBAR2 自定义应用

NBAR2 支持使用自定义协议来识别自定义应用。自定义协议功能支持 NBAR2 当前不支持的协议和应用。

每个部署场景都包含没有被 Inspur 提供的 NBAR2 协议包覆盖的本地特定的应用程序。本地应用程序主要有两类：

- 特定于组织机构的应用程序
- 特定于地理位置的应用程序

NBAR2 提供了手动定义这样的本地应用的方式。可以在全局配置模式中使用 **ip nbar custom** *myappname* 命令手动定义应用程序。自定义的应用程序优先于内置的协议。对于每个自定义协议，用户可以定义一个用来进行汇报的选择器 ID（Selector ID）。

有多种类型的应用程序自定义：

通用协议自定义

- HTTP
- SSL
- DNS

组合自定义：基于多种底层协议的自定义——**server-name**

三层/4 层自定义

- IPv4 地址
- DSCP 值

- TCP/UDP 端口
- 数据流源或目的方向

字节偏移：基于有效载荷中特定字节值的自定义

HTTP 自定义

HTTP 自定义可以基于以下 HTTP 字段的组合进行：

- **cookie**——HTTP Cookie
- **host**——拥有资源的源服务器主机名
- **method**——HTTP 方式
- **referrer**——获取资源请求的地址
- **url**——统一资源定位符路径
- **user-agent**——发送请求客户端使用的软件
- **version**——HTTP 版本
- **via**——HTTP via 字段

HTTP 自定义

自定义一个名为 MYHTTP 的应用，HTTPhost 字段使用“*mydomain.com”，设置 Selector ID 为 10。

```
Device# configure terminal
Device(config)# ip nbar custom MYHTTP http host *mydomain.com id 10
```

SSL 自定义

可以使用从 SSL 服务器名称标识（Server NameIndication，SNI）或通用名称（Common Name，CN）中提取出的信息来进行 SSL 加密流量的自定义。

SSL 自定义

自定义一个名为 MYSSL 的应用，SSL 唯一名称使用“*mydomain.com”，设置 Selector ID 为 11。

```
Device# configure terminal
Device(config)#ip nbar custom MYSSL ssl unique-name *mydomain.com id 11
```

DNS 自定义

NBAR2 会检查 DNS 请求以及应答流量，并且可以把 DNS 应答与应用程序关联。DNS 应答中返回的 IP 地址会被缓存，给之后与该应用关联的数据包流使用。

使用**ip nbar custom** *application-name* **dns** *domain-name* **id** *application-id*命令进行DNS自定义。要扩展现有应用，使用**ip nbar custom** *application-name* **dnsdomain-name** *domain-name* **extends** *existing-application*命令。

更多关于基于DNS进行自定义的信息，参见http://www.icntnetworks.com

DNS 自定义

自定义一个名为 MYDNS 的应用，DNS 域名使用“*mydomain.com”，设置 Selector ID 为 12。

```
Device# configure terminal
Device(config)# ip nbar custom MYDNS dns domain-name *mydomain.com id 12
```

组合自定义

NBAR2 提供了一种基于 HTTP、SSL 或 DNS 中出现的域名来自定义应用的方式。

组合自定义

自定义一个名为 MYDOMAIN 的应用，HTTP、SSL 或 DNS 域名使用“*mydomain.com”，设置 Selector ID 为 13。

```
Device# configure terminal
```

```
Device(config)# ip nbar custom MYDOMAIN composite server-name *mydomain.com id 13
```

三层/4 层自定义

三层/4 层自定义基于数据包元组进行，且总是匹配数据流的第一个数据包。

三层/4 层自定义

自定义一个名为 LAYER4CUSTOM 的应用，匹配 IP 地址 10.56.1.10 和 10.56.1.11，设置 TCP 与 DSCP ef，且 Selector ID 为 14。

```
Device# configure terminal
Device(config)# ip nbar custom LAYER4CUSTOM transport tcp id 14
Device(config-custom)# ip address 10.56.1.10 10.56.1.11
Device(config-custom)# dscp ef
```

示例：监控自定义的应用

监控自定义应用的 show 命令

show ip nbar protocol-id | inc Custom

```
Device# show ip nbar protocol-id | inc Custom
LAYER4CUSTOM 14 Custom
MYDNS 12 Custom
MYDOMAIN 13 Custom
MYHTTP 10 Custom
MYSSL 11 Custom
```

show ip nbar protocol-discovery protocol *CUSTOM_APP*

```
WSW-157# show ip nbar protocol-id MYSSL
Protocol Name id type
-----------------------------------------------
MYSSL 11 Custom
```

NBAR2 动态无中断协议包升级

协议包是软件包，能够升级设备上的 NBAR2 协议，而无需替换设备上的 Inspur 软件。一个协议包中包含 NBAR2 官方支持的应用程序信息，这些信息被汇集并打包到一起。对于每种应用，协议包中都有应用签名以及应用属性的信息。每个系统软件版本中都内置了协议包。协议包提供了以下特性：

- 简单且加载速度快。
- 容易升级到高版本，也容易退回到低版本。
- 不要求交换机重新加载。

用户可以在 Inspur 软件中心下载 NBAR2 协议包：

https://software.icntnetworks.com/download/navigator.html

NBAR2 协议包前提

在加载新的协议包之前，必须把协议包拷贝到所有交换机成员的闪存中。要加载协议包，参见示例：加载 NBAR2 协议包。

加载 NBAR2 协议包

总步骤

1. enable

2. configure terminal

3. ip nbar protocol-pack *protocol-pack* [**force**]

4. exit

5. show ip nbar protocol-pack {protocol-pack | **active**} [**detail**]

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。

步骤 3	**ip nbar protocol-pack** *protocol-pack*[**force**]
	示例：
	`Device(config)# ip nbar protocol-pack flash:defProtoPack`
	示例：
	`Device(config)# default ip nbarprotocol-pack`
		加载协议包。
		• 使用 **force** 关键字指定加载低版本协议包。这也会移除当前交换机协议包不支持的配置。
		要回退使用内置的协议包，使用以下命令：
		default ip nbar protocol-pack

步骤 4	**exit** **示例：** `Device(config)# exit`	返回特权 EXEC 模式。
步骤 5	**show ip nbar protocol-pack** {protocol-pack |**active**} [**detail**] **示例：** `Device# show ip nbar protocol-pack active`	显示协议包信息。 • 使用此命令验证加载的协议包版本、发布者以及其他详情。 • 使用 *protocol-pack* 参数显示特定协议包的信息。 • 使用 **active** 关键字显示活跃的协议包信息。 • 使用 **detail** 关键字显示协议包详细信息。

示例：加载 NBAR2 协议包

以下示例展示了如何加载新的协议包：

```
Device>enable
Device# configure terminal
Device(config)# ip nbar protocol-pack flash:newDefProtoPack
Device(config)# exit
```

以下示例展示了如何使用 **force** 关键字加载低版本协议包：

```
Device>enable
Device# configure terminal
Device(config)# ip nbar protocol-pack flash:OldDefProtoPack force
Device(config)# exit
```

以下示例展示了如何回退到内置协议包：

```
Device>enable
```

```
Device# configure terminal
Device(config)# default ip nbar protocol-pack
Device(config)# exit
```

监控应用程序可见性和控制

监控应用程序可见性和控制（CLI）

本节描述了应用程序可见性的新命令。
以下命令可以用来在接入端口上监控应用程序可见性。

表 188：在接入端口上监控应用程序可见性

命令	目的
show ip nbar protocol-discovery [**interface***interface-type interface-number*] [**stats**{**byte-count**| **bit-rate** | **packet-count** | **max-bit-rate**}] [**protocol***protocol-name* | **top-n** *number*]	显示 NBAR 协议发现特性收集的统计数据。 • （可选）输入关键字和参数来微调显示的统计信息。关于每个关键字的更多信息，参见 *Inspur INOS 服务质量解决方案命令参考*中的 **showip nbarprotocol-discovery** 命令。
show policy-map interface *interface-typeinterface-number*	显示应用到接口上的策略映射的信息。
show platform software fed switch *switch id***wdavc flows**	显示特定交换机上所有数据流的统计信息。

示例：应用程序可见性和控制

示例：应用程序可见性和控制配置

此示例展示了如何创建类映射，并对应用名称使用匹配协议过滤器：

```
Device# configure terminal
Device(config)# class-map match-any NBAR-VOICE
Device(config-cmap)# match protocol ms-lync-audio
Device(config-cmap)#end
```

此示例展示了如何创建策略映射，并为出向 QoS 指定现有的类映射：

```
Device# configure terminal
Device(config)# policy-map test-avc-up
Device(config-pmap)# class cat-browsing
Device(config-pmap-c)# police 150000
Device(config-pmap-c)# set dscp 12
Device(config-pmap-c)#end
```

此示例展示了如何创建策略映射，并为入向 QoS 指定现有的类映射：

```
Device# configure terminal
Device(config)# policy-map test-avc-down
```

```
Device(config-pmap)# class cat-browsing
Device(config-pmap-c)# police 200000
Device(config-pmap-c)# set dscp 10
Device(config-pmap-c)#end
```

此示例展示了如何把策略映射应用到交换机端口上：

```
Device# configure terminal
Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# switchport mode access
Device(config-if)# switchport access vlan 20
Device(config-if)# service-policy type control subscriber POLICING_IN
Device(config-if)#end
```

查看配置的 show 命令

show ip nbar protocol-discovery

显示每个接口的协议发现统计数据报告。

以下是每个接口统计信息的示例输出：

```
Deviceqos-cat3k-reg2-r1# show ip nbar protocol-discovery int GigabitEthernet1/0/1
GigabitEthernet1/0/1
Last clearing of "show ip nbar protocol-discovery" counters 00:03:16
Input
Output
------
-----
Protocol Packet Count
Packet Count
Byte Count
30sec Bit Rate (bps)
Byte Count
30sec Bit Rate (bps)
30sec Max Bit Rate (bps)
30sec Max Bit Rate (bps)
---------------------- ------------------------
--------------------------------------------------
ms-lync 60580
55911
31174777
28774864
3613000
93000
3613000
3437000
Total 60580
55911
31174777
```

28774864
3613000
93000
3613000
3437000

show policy-map interface

显示所有接口上的 QoS 统计数据以及配置的策略映射。
以下是所有接口上配置的策略映射示例输出：

```
Deviceqos-cat3k-reg2-r1# show policy-map int
GigabitEthernet1/0/1
Service-policy input: MARKING-IN
Class-map: NBAR-VOICE (match-any)
718 packets
Match: protocol ms-lync-audio
0 packets, 0 bytes
30 second rate 0 bps
QoS Set
dscp ef
Class-map: NBAR-MM_CONFERENCING (match-any)
6451 packets
Match: protocol ms-lync
0 packets, 0 bytes
30 second rate 0 bps
Match: protocol ms-lync-video
0 packets, 0 bytes
30 second rate 0 bps
QoS Set
dscp af41
Class-map: class-default (match-any)
34 packets
Match: any
```

基本故障排除问答

按照以下基本故障排除问答来解决有线应用程序可见性和控制的问题：

1 **问题：**我的 IPv6 流量没有被分类。
回答：当前仅支持 IPv4 流量。

2 **问题：**我的组播流量没有被分类。
回答：当前仅支持单播流量。

3 **问题：**我发送了 ping，但是 ping 包没有被分类
回答：仅支持 TCP/UDP 协议。

4 **问题：**为什么不能把 NBAR 配置在 SVI 上？
回答：NBAR 仅支持在物理接口上使用。

5 **问题：**在协议发现过程中，我只在一端看到了流量，而且有许多未知的流量。

回答：这通常表示 NBAR 发现了非对称的流量：流量的一端在一个交换机成员上进行了分类，而另一端在不同的交换机成员上分类。建议仅在能接收流量两端的接入端口上配置 NBAR。如果存在多个上行链路，因为存在这样的问题，所以不能在链路上配置 NBAR。在端口通道中的接口上配置 NBAR 也会出现类似的问题。

6 **问题：**使用协议发现，我能看到所有应用的汇聚视图。怎样查看一段时间内的流量分布？
回答：Web UI 可以查看过去 48 小时的流量。

7 **问题：**我不能使用 **match protocol** *protocol-name* 命令配置基于队列的出向策略。
回答：在使用基于 NBAR2 分类器的策略中，仅支持 **shape** 和 **set DSCP**。通常的方式是在入向设置 DSCP，并基于 DSCP 在出向执行流量整型。

8 **问题：**我没有在接口上配置 NBAR2，但是仍能看到 NBAR2 被激活。
回答：如果使用带有 **match protocol** *protocol-name* 的类映射，NBAR 会在堆栈上全局激活，但是流量不会进行 NBAR 分类。这是预期行为，不会消耗任何资源。

9 **问题：**我看见一些流量在默认 QoS 队列下。这是为什么？
回答：对于每个新的数据流，需要通过几个数据包来对其进行分类并把结果安装在硬件中。在此期间，分类是“未知的”，且流量会进入默认队列。

其他参考资料

相关文档

相关主题	文档标题
QoS	*NBAR 配置指南，Inspur INOS*
NBAR2 协议包无中断升级	*NBAR 配置指南，Inspur INOS*

有线网络中应用程序可见性和控制的特性历史与信息

版本	特性信息
Inspur INOS 11.3.1	此特性被引入。

园区交换矩阵

- 园区交换矩阵

园区交换矩阵

园区交换矩阵（Campus Fabric）为使用基于策略的分段结构建造虚拟网提供了基础设施。本节描述了如何在 Inspur 交换机上配置园区交换矩阵。

园区交换矩阵概述

园区交换矩阵（Campus Fabric）为使用基于策略的分段结构建造虚拟网提供了基础设施。交换矩阵 Overlay 提供的服务有主机移动性以及增强安全性等，这些是正常的交换与路由能力的补充。

园区交换矩阵 Overlay 规划由三个主要元素组成：

- 控制层
- 数据层
- 策略层

理解交换矩阵域元素

下图展示了组成交换矩阵域的元素。

User/Group Repository	用户/组仓库
Fabric Domain(Overlay)	交换矩阵域（Overlay）
Fabric Edge Nodes	交换矩阵边缘节点
ISE/AD	ISE/AD
Host DB	主机 DB
Control-Plane Nodes	控制层节点
Fabric Border Nodes	交换矩阵边界节点

- 交换矩阵边缘设备（Fabric Edge Devices）——为连接到交换矩阵域的用户和设备提供连通性。交换矩阵边缘设备能够识别并认证终端，并在交换矩阵主机追踪数据库中注册终端 ID 信息。这些设备在入向封装并在出向解封装，并为连接到交换矩阵域的终端转发流量。
- 交换矩阵控制层设备（Fabric Control-Plane Devices）——在主机追踪数据库中提供 overlay 可达性信息以及终端到路由定位器的映射。控制层设备会接收来自交换矩阵边界交换机的本地终端注册信息，并解析边界设备的请求以定位远程终端。可以配置至多 3 个控制层设备，为网络提供冗余性。
- 交换矩阵边界设备（Fabric Border Devices）——连接传统三层网络或不同的交换矩阵域到本地域，并翻译域间的可达性以及策略信息，如 VRF 以及 SGT 信息。
- 虚拟环境（Virtual Contexts）——提供设备级别的虚拟化，使用虚拟路由转发（virtual routing and forwarding，VRF）来创建多个三层路由表实例。虚拟环境或 VRF 提供了跨越 IP 地址的分段，允许使用重叠的地址空间，并能实现流量隔离。可以在交换矩阵域中配置至多 32 个虚拟环境。
- 主机池（Host-Pools）——把交换矩阵域中的终端分组到 IP 池中，并通过 VLAN ID 以及 IP 子网识别主机。

园区交换矩阵配置指南

按照指南配置园区交换矩阵元素时请考虑以下限制：

- 每个交换矩阵域中不要配置多于 3 个控制层设备。
- 每个交换矩阵边缘设备最多支持 2000 个终端。

- 每个控制层设备最多支持 5000 个交换矩阵边缘设备注册。
- 每个交换矩阵域中不要配置多于 32 个虚拟环境。

如何配置交换矩阵 Overlay

配置交换矩阵边缘设备

按照以下步骤配置交换矩阵边缘设备。

在开始前

为每个边缘设备配置 loopback0 的 IP 地址，确保设备可达。

总步骤

1. enable
2. configure terminal
3. fabric auto
4. domain {**default** | **name** *fabric domain name*}
5. control-plane *ipv4 address* **auth_key** *key*
6. border *ipv4 address*
7. context name *name* **id** *ID*
8. host-pool name *name*
9. vlan *ID*
10. gateway *IP address/ mask*
11. context name *name*
12. use-dhcp *IP address*
13. exit
14. show fabric domain

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 3	**fabric auto** **示例：** `Device(config)#`**fabric auto**	启用自动交换矩阵规划，并进入自动交换矩阵配置模式。
步骤 4	**domain** {**default** | **name** *fabric domain name*} **示例：** `Device(config-fabric-auto)#`**domain default** `Device(config-fabric-auto)#`**domain name *exampledomain***	配置默认的交换矩阵域并进入域配置模式。**name** 关键字允许用户添加新的交换矩阵域。使用命令的 **no** 形式来删除交换矩阵域。

步骤 5	**control-plane** *ipv4 address* **auth_key** *key* **示例：** `Device(config-fabric-auto-domain)#`**control-plane*198.51.100.2*auth_key *examplekey123***	配置控制层设备 IP 地址以及认证密钥，允许交换矩阵边缘设备与控制层设备通信。命令的 **no** 形式会从交换矩阵域中删除控制层设备。
步骤 6	**border** *ipv4 address* **示例：** `Device(config-fabric-auto-domain)#`**border *198.51.100.4***	配置交换矩阵边界设备的 IP 地址，允许交换矩阵边缘设备与交换矩阵边界设备通信。
步骤 7	**context name** *name* **id** *ID* **示例：** `Device(config-fabric-auto-domain)#`**context name*example-context*id *10***	在交换矩阵域中创建新的虚拟环境并指定 ID。
步骤 8	**host-pool name** *name* **示例：** `Device(config-fabric-auto-domain)#`**host-pool name*VOICE_DOMAIN***	创建 IP 池来对交换矩阵中的终端进行分组，并进入主机池配置模式。
步骤 9	**vlan** *ID* **示例：** `Device(config-fabric-auto-domain-host-pool)#`**vlan *10***	配置 VLAN ID，与主机池关联。
步骤 10	**gateway** *IP address/ mask* **示例：** `Device(config-fabric-auto-domain-host-pool)#`**gateway*192.168.1.254/24***	配置主机池的路由网关 IP 地址以及子网掩码。地址和掩码会被用来映射终端标识符（EID）到 RLOC。
步骤 11	**context name** *name* **示例：** `Device(config-fabric-auto-domain-host-pool)#`**contextname*example-context***	把场景或 VRF 与主机池进行关联。可以在交换矩阵域中配置至多 32 个场景。
步骤 12	**use-dhcp** *IP address* **示例：** `Device(config-fabric-auto-domain-host-pool)#`**use-dhcp *172.10.1.1***	为主机池配置DHCP服务器地址。可以我主机池配置多个DHCP地址。要删除DHCP服务器地址，使用**nouse-dhcp** *IP address*命令。**no use-dhcp**命令会删除所有的DHCP地址。
步骤 13	**exit** **示例：** `Device(config-fabric-auto-domain)#` **exit**	
步骤 14	**show fabric domain** **示例：** `Device#` **show fabric domain**	显示交换矩阵域配置。作为配置的一部分，其他的 CLI 命令会自动生成。更多信息参见*交换矩阵边缘设备上自动配置的命令*。

配置控制层设备

要配置控制层设备，使用以下 LISP 命令：

在开始前

为控制层设备配置环回 IP 地址，确保设备可达性。

总步骤

1. enable

2. configure terminal

3. router lisp

4. site *site-name*

5. authentication-key *key*

6. eid-prefix [**instance-id** *instance-id*] *eid-prefix* **accept-more-specifics**

7. exit

8. 重复进行步骤4到步骤7，创建另一个LISP站点。

9. ipv4 map-server

10. ipv4 map-resolver

11. end

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 3	**router lisp** **示例：** `Device(config)# router lisp`	进入定位器 ID/分离协议（Locator ID/Separation Protocol，LISP）配置模式。
步骤 4	**site** *site-name* **示例：** `Device(config-router-lisp)# siteFD_Default`	在控制层设备上配置 LISP 站点，并进入 LISP 站点配置模式。
步骤 5	**authentication-key** *key* **示例：** `Device(config-router-lisp-site)#authentication-key examplekey`	配置用于创建基于哈希的消息认证码（Hash-based MessageAuthentication Code，HMAC）安全哈希算法（Secure Hash Algorithm，SHA-1）哈希值的密码。此密码会对边缘设备与控制层设备注册时发送的映射注册消息使用。

步骤 6	**eid-prefix** [**instance-id** *instance-id*] *eid-prefix***accept-more-specifics**
	示例：
	`Device(config-router-lisp-site)#eid-prefix 10.1.0.0/16accept-more-specifics`
	`Device(config-router-lisp-site)#eid-prefix instance-id 10 10.1.0.0/16accept-more-specifics`
		配置主机池或允许的终端标识符（EID）前缀列表，对于边缘设备与控制层设备注册时发送的映射注册消息使用。设备会接收并追踪比配置的 EID 前缀更具体的 EID 前缀。使用 **instance-id** 关键字把特定的实例 ID（包含在主机池中的场景使用的实例 ID）配置在主机池中，与控制层设备注册时可以使用此 ID。

步骤 7	**exit** **示例：** `Device(config-router-lisp-site)# exit`	退出 LISP 站点配置模式，并返回 LISP 配置模式。
步骤 8	重复进行步骤4到步骤7，创建另一个LISP站点。	-
步骤 9	**ipv4map-server** **示例：** `Device(config-router-lisp)# ipv4map-server`	配置设备作为 IPv4 控制层设备工作。
步骤 10	**ipv4 map-resolver** **示例：** `Device(config-router-lisp)# ipv4map-resolver`	在交换矩阵域中，控制层设备作为映射服务器（map-server）以及映射解析器（map-resolver）使用。启用控制层设备的 IPv4 LISP 映射协议功能。
步骤 11	**end** **示例：** `Device(config-router-lisp)# end`	退出 LISP 配置模式，并返回特权 EXEC 模式。

配置边界设备

使用以下 LISP 命令配置边界设备。

在开始前

为边界设备配置环回 IP 地址，确保可达性。

总步骤

1. enable

2. configure terminal

3. router lisp

4. encapsulation vxlan

5. eid-table default instance-id *instance-id*

6. map-cache *eid-prefix ipv4 address/subnet mask* **map-request**

7. ipv4 sgt

8. ipv4 proxy-etr

9. ipv4 proxy-itr *ipv4 address*

10. exit

11. ip route *ipv4-prefix next-hop*

12. exit

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。

步骤 3	**router lisp** **示例：** `Device(config)# router lisp`	进入 LISP 配置模式。
步骤 4	**encapsulation vxlan** **示例：** `Device(config-router-lisp)# encapsulationvxlan`	指定使用基于 VXLAN 的封装。
步骤 5	**eid-table default instance-id** *instance-id* **示例：** `Device(config-router-lisp)# eid-tabledefault instance-id 0`	把默认的 EID 表与指定的实例 ID 关联。 控制层设备消息会包含此实例 ID 以及关联的 EID 前缀。
步骤 6	**map-cache** *eid-prefix ipv4 address/subnet mask***map-request** **示例：** `Device(config-router-lisp)# map-cache 10.1.1.0/24 map-request`	配置静态 IPv4 EID 到 RLOC 的映射关系，添加 map-cache，对指定的动态 EID 或主机池使用 send-map-request 行为。
步骤 7	**ipv4 sgt** **示例：** `Device(config-router-lisp)# ipv4 sgt`	在交换矩阵中启用安全组标签（Security Group Tags，SGT）的传输。关于 SGT 的更多信息，参见 Inspur TrustSec 配置指南。
步骤 8	**ipv4 proxy-etr** **示例：** `Device(config-router-lisp)# ipv4 proxy-etr`	在交换矩阵域中启用边界设备服务。
步骤 9	**ipv4 proxy-itr** *ipv4 address* **示例：** `Device(config-router-lisp)# ipv4 proxy-itr 10.1.1.1`	配置设备作为 IPv4 中继入向隧道路由器（proxy ingress tunnelrouter，PITR）使用，并配置接口 IP 地址作为封装数据包的源地址使用。IPv4 定位器的地址会被用作数据包或映射请求（map-request）消息的源地址。
步骤 10	**exit** **示例：** `Device(config-router-lisp)# exit`	退出 LISP 配置模式，并进入全局配置模式。
步骤 11	**ip route** *ipv4-prefix next-hop* **示例：** `Device(config)# ip route 0.0.0.0 0.0.0.0 10.10.10.1`	配置 IPv4 静态路由。
步骤 12	**exit** **示例：** `Device(config)# exit`	退出全局配置模式并返回特权 EXEC 模式。

交换矩阵边缘设备上自动配置的命令

作为交换矩阵 Overlay 规划的一部分，一些基于 LISP 的配置，SGT（安全组标签）配置以及 EID 到 RLOC 映射配置会自动生成，并且在运行配置中出现。

例如，考虑以下边缘设备配置场景（环回接口地址 2.1.1.1/32）：

```
device(config)#fabric auto
device(config-fabric-auto)#domain default
device(config-fabric-auto-domain)#control-plane 192.168.1.4 auth-key example-key1
device(config-fabric-auto-domain)#control-plane 192.168.1.5 auth-key example-key2
device(config-fabric-auto-domain)#border 192.168.1.6
device(config-fabric-auto-domain)#context name example-context ID 10
device(config-fabric-auto-domain)#host-pool name VOICE_DOMAIN
device(config-fabric-auto-domain-host-pool)#vlan 10
device(config-fabric-auto-domain-host-pool)#context example-context
device(config-fabric-auto-domain-host-pool)#gateway 192.168.1.254/24
device(config-fabric-auto-domain-host-pool)#use-dhcp 209.165.201.6
```

以下是交换矩阵边缘配置的示例输出：

```
device#show running-config
router lisp
encapsulation vxlan
locator-set default.RLOC
IPv4-interface Loopback0 priority 10 weight 10
exit
!
eid-table default instance-id 0
exit
!
eid-table vrf example-context instance-id 10
dynamic-eid example-context.EID.VOICE_DOMAIN
database-mapping 192.168.1.0/24 locator-set default.RLOC
exit
!
exit
!
loc-reach-algorithm lsb-reports ignore
disable-ttl-propagate
ipv4 sgt
ipv4 use-petr 192.168.1.6 priority 10 weight 10
ipv4 itr map-resolver 192.168.1.4
ipv4 itr map-resolver 192.168.1.5
ipv4 itr
ipv4 etr map-server 192.168.1.4 key example-key1
ipv4 etr map-server 192.168.1.5 key example-key2
ipv4 etr
```

```
exit
!
```

更多关于 LISP 元素的信息及配置命令，参见*配置 LISP*。

示例：配置交换矩阵边缘、边界以及控制层设备

```
device#show running-config
!
ip vrf example-context
description Auto-provisioned vrf for neighborhood example-context
!
ip dhcp relay information option vpn
ip dhcp relay information option
!
ip dhcp snooping vlan 10
ip dhcp snooping
!
!
fabric auto
!
domain default
control-plane 192.168.1.4 auth-key example-key1
control-plane 192.168.1.5 auth-key example-key2
border 192.168.1.6
context name example-context id 10
!
host-pool name VOICE_DOMAIN
context example-context
vlan 10
gateway 192.168.1.254/24
use-dhcp 209.65.201.6
exit
exit
exit
!
vlan 10
name VOICE_DOMAIN
!
interface Vlan10
ip vrf forwarding example-context
ip dhcp relay source-interface Loopback0
ip address 192.168.1.254 255.255.255.0
ip helper-address global 209.65.201.6
no ip redirects
```

```
ip local-proxy-arp
ip route-cache same-interface
no lisp mobility liveness test
lisp mobility example-context.EID.VOICE_DOMAIN
!
router lisp
encapsulation vxlan
locator-set default.RLOC
IPv4-interface Loopback0 priority 10 weight 10
exit
!
eid-table default instance-id 0
exit
!
eid-table vrf example-context instance-id 10
dynamic-eid example-context.EID.VOICE_DOMAIN
database-mapping 192.168.1.0/24 locator-set default.RLOC
exit
!
exit
!
loc-reach-algorithm lsb-reports ignore
disable-ttl-propagate
ipv4 sgt
ipv4 use-petr 192.168.1.6 priority 10 weight 10
ipv4 itr map-resolver 192.168.1.4
ipv4 itr map-resolver 192.168.1.5
ipv4 itr
ipv4 etr map-server 192.168.1.4 key example-key1
ipv4 etr map-server 192.168.1.5 key example-key2
ipv4 etr
exit
!
```

配置 SDM 模板

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用 Inspur 特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问 Inspur 特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于配置 SDM 模板的信息

SDM 模板

可以使用 SDM 模板配置系统资源，并根据网络中的设备使用情况来优化对特定特性的支持。可以选用模板来为一些功能提供最大的系统使用情况。

设备上支持的模板如下：

- 高级模板——此版本的所有支持镜像中均可用。对于 netflow、组播组、安全 ACE、QoS ACE 等特性，模板能最大化系统资源使用。
- VLAN 模板——VLAN 模板仅对于 LAN Base 许可证可用。VLAN 模板禁用路由，支持最大数量的单播 MAC 地址。该模板通常对二层设备选用。

更改模板并重启系统之后，可以使用特权 EXEC 命令 **show sdm prefer** 来验证新的模板配置。如果在输入 **reload** 特权 EXEC 命令之前输入 **show sdm prefer** 命令，输出会显示当前在用的模板以及重启后会激活的模板。

默认使用高级模板，

表 198：模板允许的特性资源大致数量

资源	**高级**	**VLAN**
VLAN 数量	4094	4094
单播 MAC 地址	32K	32K
溢出的单播 MAC 地址	512	512
IGMP 组以及组播路由	4K	4K
溢出的 IGMP 组以及组播路由	512	512
直连路由	32K	32K
非直连 IP 主机	8K	8K
基于策略路由 ACE	1024	0
QoS 分类 ACE	3K	3K
安全 ACE	1.5K	1.5K
Netflow ACE	1024	1024
输入 Microflow 限速器 ACE	256K	256K

输出 Microflow 限速器 ACE	256K	256K
FSPAN ACE	256	256
隧道	256	0
控制层条目	512	512
输入 Netflow 流	8K	8K
输出 Netflow 流	16K	16K
SGT/DGT 条目	4K	4K
SGT/DGT 溢出条目	0	512

注释： 当交换机被用作无线移动性代理（Wireless Mobility Agent）时，仅允许使用高级模板。

表中展示了选用模板时的大致硬件限制。如果某部分硬件资源占满，所有在处理的过载都会被送往 CPU，这会严重影响交换机性能。

SDM 模板与交换机堆栈

在交换机堆栈中，所有堆栈成员都必须使用存储在活跃交换机上的相同 SDM 模板。当新交换机被添加到堆栈时，存储在活跃交换机上的 SDM 配置会覆盖独立交换机上配置的模板。可以使用特权 EXEC 命令 **show switch** 查看堆栈成员是否在 SDM 不匹配模式中。

如何配置 SDM 模板

配置 SDM 模板

配置交换机 SDM 模板

设置 SDM 模板

按照以下步骤使用 SDM 模板最大化特性使用情况：

总步骤

1. enable

2. configure terminal

3. sdm prefer { **advanced** | **vlan** }

4. end

5. reload

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**sdm prefer** { **advanced** | **vlan** } **示例:** `Device(config)#` **sdm prefer advanced**	指定要在交换机上使用的 SDM 模板。关键字含义如下： • **advanced**——支持高级特性，

		如 Netflow。 • **vlan**——最大化交换机上的 VLAN 配置，在硬件上不支持路由。 **注释：no sdm prefer** 命令以及默认模板不被支持。
步骤 4	**end** **示例:** `Device(config)# end`	返回特权 EXEC 模式。
步骤 5	**reload** **示例:** `Device# reload`	重新加载操作系统。

监控并维护 SDM 模板

命令	目的
show sdm prefer	显示使用的 SDM 模板。
reload	重新加载交换机，激活新配置的 SDM 模板。
no sdm prefer	设置默认 SDM 模板。

SDM 模板的配置示例

示例：配置 SDM 模板

示例：显示 SDM 模板

此示例输出显示了高级模板信息：

```
Device# show sdm prefer
Showing SDM Template Info
This is the Advanced template.
Number of VLANs: 4094
Unicast MAC addresses: 32768
Overflow Unicast MAC addresses: 512
IGMP and Multicast groups: 8192
Overflow IGMP and Multicast groups: 512
Directly connected routes: 32768
Indirect routes: 8192
Security Access Control Entries: 3072
QoS Access Control Entries: 2816
Policy Based Routing ACEs: 1024
Netflow ACEs: 1024
Input Microflow policer ACEs: 256
```

```
Output Microflow policer ACEs: 256
Flow SPAN ACEs: 256
Tunnels: 256
Control Plane Entries: 512
Input Netflow flows: 8192
Output Netflow flows: 16384
These numbers are typical for L2 and IPv4 features.
Some features such as IPv6, use up double the entry size;
so only half as many entries can be created.
Device#
```

此示例输出显示了 VLAN 模板信息：

```
Device# show sdm prefer vlan
Showing SDM Template Info
This is the VLAN template for a typical Layer 2 network.
Number of VLANs: 4094
Unicast MAC addresses: 32768
Overflow Unicast MAC addresses: 512
IGMP and Multicast groups: 8192
Overflow IGMP and Multicast groups: 512
Directly connected routes: 32768
Indirect routes: 8192
Security Access Control Entries: 3072
QoS Access Control Entries: 3072
Policy Based Routing ACEs: 0
Netflow ACEs: 1024
Input Microflow policer ACEs: 0
Output Microflow policer ACEs: 0
Flow SPAN ACEs: 256
Tunnels: 0
Control Plane Entries: 512
Input Netflow flows: 16384
Output Netflow flows: 8192
These numbers are typical for L2 and IPv4 features.
Some features such as IPv6, use up double the entry size;
so only half as many entries can be created.
Device#
```

配置 SDM 模板的特性历史与信息

版本	修改
Inspur INOS 11.3.1	此特性被引入。

配置系统消息日志

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用 Inspur 特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问 Inspur 特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于配置系统消息日志的信息

系统消息日志

默认情况下，交换机会把系统消息以及特权 EXEC 命令 **debug** 的输出发送给日志进程。堆栈成员可以触发系统消息。生成系统消息的堆栈成员会把按照“hostname-n”的形式（n 是交换机编号）把自己的主机名附加到消息中，并把输出重定向给活跃交换机的日志进程。虽然活跃交换机是堆栈成员，但是它不会把自己的主机名附加到系统消息上。日志进程控制着日志消息的分发，根据配置可以把消息发送到众多目的，如日志缓冲区、终端线路或 UNIXsyslog 服务器。此进程也可以把消息发送到控制台。

日志进程被禁用时，消息只会被发往控制台。消息生成时就会被发出，所以日志消息、调试输出以及提示或其他命令的输出会穿插在一起。生成消息的进程完成后消息会出现在活跃的控制台上。

可以设置消息的严重性等级，控制在控制台上展示以及发送给每个目的的消息类型。可以给日志消息加时间戳，或者设置 syslog 源地址，来增强实时调试与管理能力。有关可能出现的消息的信息，参见此版本的系统消息指南。

可以使用交换机的命令行界面（CLI）来访问记录的系统消息，也可以把消息保存到配置好的 syslog 服务器上。在单独的交换机中，交换机软件会把 syslog 消息保存到内部缓冲区，而在交换机堆栈中，消息会保存在活跃交换机上。如果单独交换机或堆栈 master 故障，除非保存到了闪存中，否则日志会丢失。

可以通过查看 syslog 服务器上的日志来远程监控系统消息，也可以通过 Telnet、控制台端口或以太网管理端口访问交换机。在交换机堆栈上，所有堆栈成员的控制台都有相同的控制台输出。

注释： syslog 的格式与 4.3 BSD UNIX 兼容。

系统日志消息格式

系统日志消息可以包含至多 80 个字符以及一个百分号（%），之前是可选配置的序号或时间戳信息。根据交换机不同，消息会按以下格式显示：

- seq no:timestamp: %facility-severity-MNEMONIC:description (hostname-n)
- seq no:timestamp: %facility-severity-MNEMONIC:description

消息百分号之前的内容取决于以下全局配置命令的设置：

- **service sequence-numbers**
- **service timestamps log datetime**
- **service timestamps log datetime** [**localtime**] [**msec**] [**show-timezone**]
- **service timestamps log uptime**

表 199：系统日志消息元素

元素	**描述**
seq no:	如果配置了全局配置命令 **service sequence-numbers**，会给消息标记上序号。
timestamp formats: *mm/dd h h:mm:ss* 或 *hh:mm:ss* (短形式启用时间) 或 *d h* (长形式启用时间)	消息或事件的日期与时间。此信息只在配置了全局配置命令**service timestampslog** [**datetime** | **log**]时出现。
facility	消息涉及的设备（如 SNMP、SYS 等）。
severity	表示消息严重性的数字代码，从 0 到 7。
MNEMONIC	唯一描述消息的文本串。
description	包含报告事件详细信息的文本串。
hostname-n	堆栈成员的主机名及其在堆栈中的交换机编号。虽然活跃交换机是堆栈成员，但它*不会*把自己的主机名附加到系统消息中。

默认系统消息日志设置

表 200：默认系统消息日志设置

特性	默认设置
系统消息日志输出到控制台	启用。
控制台严重级别	调试（Debugging）。
日志文件配置	未指定文件名。
日志缓冲区大小	4096 字节。
日志历史大小	1 个消息。
时间戳	禁用。
同步记录	禁用。
日志服务器	禁用。
syslog 服务器 IP 地址	未配置。
服务器设备	Local7
服务器严重级别	信息（Informational）。

Syslog 消息限制

如果使用全局配置命令 **snmp-server enable trap** 配置给 SNMP 网络管理工作站发送 syslog 消息陷阱，可以更改发送消息的等级以及储存在交换机历史表中的消息等级。也可以更改储存在历史表中的消息数量。

因为 SNMP 陷阱不保证能送达目的，所以消息会被保存在历史表中。默认情况下，即使没有启用 syslog 陷阱，历史表中也会储存一个 **warning** 级别以及数值更低级别的消息。

当历史表存满时（表中存储了全局配置命令**logginghistory size**指定的最大数量消息条目），最老的消息条目会被从表中删除，以允许存储新消息。

历史表中列出了消息的级别关键字以及安全性等级。对于SNMP，安全性等级值以1递增。例如，*emergencies*等于1，*critical*等于3。

如何配置系统消息日志

设置显示消息的目的设备

如果启用消息日志功能，可以把消息发送给控制台以外的其他位置。此任务是可选的。

总步骤

1. **configure terminal**
2. **logging buffered** [*size*]
3. **logging** *host*
4. **logging file flash:** *filename* [*max-file-size* [*min-file-size*]] [*severity-level-number* | *type*]
5. **end**
6. **terminal monitor**

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**logging buffered** [*size*] **示例：** `Device(config)#` **logging buffered8192**	对于单独交换机，日志消息记录在交换机的内部缓冲区；对于交换机堆栈，日志消息记录在活跃交换机的内部缓冲区。范围从 4096 到 2147483647 字节。默认缓冲区大小是 4096 字节。 如果单独交换机或者活跃交换机故障，日志文件会丢失，除非之前把它保存在闪存内存中，见步骤 4。 **注释：**不要把缓冲区大小设置的太大，因为交换机可能耗尽其他任务所需的内存。使用特权 EXEC 命令

		show memory 查看交换机处理器的空闲内存空间。不过显示的值是最大可用值，所以缓冲区大小不应该设置为此值。
步骤 3	**logging** *host* **示例：** `Device(config)#` **logging125.1.1.100**	把日志消息发送到 UNIX syslog 服务器主机。 *host* 指定了用作 syslog 服务器主机的名称或 IP 地址。要创建接收日志消息的 syslog 服务器列表，请多次输入此命令。
步骤 4	**logging file flash:** *filename*[*max-file-size* [*min-file-size*]][*severity-level-number* | *type*] **示例：** `Device(config)#` **logging file flash:log_msg.txt 40960 4096 3**	对于单独交换机，日志消息记录在交换机的内存文件中；对于交换机堆栈，日志消息记录在活跃交换机的内存文件中。 • *filename*——输入日志消息文件名。 • （可选）**max-file-size**——指定最大日志文件大小。范围从 4096 到 2147483647 字节，默认值是 4096 字节。 • （可选）*min-file-size*——指定最小日志文件大小。范围从 1024 到 2147483647 字节，默认值是 2048 字节。 • （可选）*severity-level-number* | *type*——指定日志安全性等级或日志类型。安全性等级范围从到 7。
步骤 5	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 6	**terminal monitor** **示例：** `Device#` **terminal monitor**	在当前会话期间把日志消息发送到非控制台的终端。 终端参数设置命令在本地进行设置，且会话结束后失效。要查看调试消息，必须为每个会话执行此步骤。

同步日志消息

对于特定的控制台端口线路或虚拟终端线路，可以对非请求消息、特权 EXEC 命令 **debug** 输出以及请求设备消息、提示进行同步。可以基于安全性等级，指定要异步输出的消息类型。也可以为终端配置存储异步消息的最大缓冲区数量，超过此数量后消息会被丢弃。

对非请求的日志消息以及 **debug** 命令输出进行同步控制时，控制台上的非请求设备输出会在请求设备输出之后出现。控制台上的非请求消息以及 **debug** 命令输出会出现在用户输入提示返回之后。因此，非请求消息以及 **debug** 命令输出就不会与请求命令输出及提示穿插在一起。在非请求消息显示之后，控制台会再次显示用户提示符。
此任务是可选的。

总步骤

1. configure terminal

2. line [**console** | **vty**] *line-number* [*ending-line-number*]

3. logging synchronous [**level** [*severity-level* | **all**] | **limit** *number-of-buffers*]

4. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**line** [**console**	**vty**] *line-number*[*ending-line-number*] **示例:** `Device(config)#` **line console**	指定要配置的同步输出日志消息的线路。 • **console**——指定消息通过交换机控制台端口或以太网管理端口费阿松。 • **line vty** *line-number*——指定哪个 vty 线路要启用同步日志控制。可以对 Telnet 会话进行的配置使用 vty 连接。线路编号范围从 0 到 15。输入以下命令可以一次更所全部 16 个 vty 线路的设置： **line vty 0 15** 也可以更改当前连接使用的 vty 线路设置。比如，要更改 vty 线路 2 的设置，输入： **line vty 2** 输入此命令时，配置模式更改为线路配置模式。
步骤 3	**logging synchronous** [**level**[*severity-level*	**all**]	**limit***number-of-buffers*] **示例:** `Device(config)#` **loggingsynchronous level 3 limit 1000**	启用日志消息同步控制。 • （可选）**level** *severity-level*——指定消息安全性等级。安全性等级等于或大于此值的消息会异步打印。数字越小安全性等级越高。默认值是 2。 • （可选）**level all**——指定所有消息都异步输出，无论安全性等级如何。 • （可选）**limit** *number-of-buffers*——

		指定终端的缓冲区数量，超过此数量的消息会被丢弃。范围从 0 到 2147483647，默认值是 20。
步骤 4	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。

禁用消息日志

消息日志默认被启用。要把消息发送给控制台以外的目的地，消息日志必须被启用。启用时，日志消息会被发送给日志进程，该进程会把消息异步发送给指定的位置。
禁用日志进程可能会减慢交换机速度，因为进程必须等待消息写到控制台之后才能继续运行。
禁用日志进程后，消息产生后会立刻显示在控制台上，通常会出现在命令输出之间。
全局配置命令 **logging synchronous** 也会影响控制台的消息显示。启用个该命令时，消息仅在用户输入**回车**之后显示。
要在禁用消息日志后重新启用，需使用全局配置命令 **logging on**。此任务是可选的。

总步骤

1. configure terminal
2. no logging console
3. end

具体步骤

	命令或操作	**目的**
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**no logging console** **示例：** `Device(config)#` **no logging console**	禁用消息日志。
步骤 3	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。

启用及禁用日志消息的时间戳

默认情况下，日志消息不加时间戳。
此任务是可选的。

总步骤

1. configure terminal
2. 使用以下命令之一：

- **service timestamps log uptime**
- **service timestamps log datetime**[**msec** | **localtime** | **show-timezone**]

3. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	使用以下命令之一： **• service timestamps log uptime** **• service timestamps log datetime**[**msec** |**localtime** | **show-timezone**] **示例：** `Device(config)#` **service timestamps log uptime** 或 `Device(config)#` **service timestamps log datetime**	启用日志时间戳。 • **log uptime**——在日志消息上加时间戳，显示系统重启后经过的时间。 • **log datetime**——在日志消息上加时间戳。根据选择选项不同，时间戳可以包含日期、相对于本地时区的毫秒形式时间以及时区名。
步骤 3	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。

启用及禁用日志消息序号

如果有多个时间戳相同的消息，可以显示带有序号的消息。默认情况下日志消息的序号不被显示。
此任务是可选的。

总步骤

1. configure terminal

2. service sequence-numbers

3. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**service sequence-numbers** **示例：** `Device(config)#` **service sequence-numbers**	启用序号。
步骤 3	**end**	返回特权 EXEC 模式。

	示例： `Device(config)#` **end**	

定义消息安全性等级

可以通过指定消息的等级来限制显示到所选设备的消息。
此任务是可选的。

总步骤

1. **configure terminal**
2. **logging console** *level*
3. **logging monitor** *level*
4. **logging trap** *level*
5. **end**

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**logging console** *level* **示例：** `Device(config)#` **logging console 3**	限制显示到控制台的日志消息。 默认时，控制台接收 debugging 消息以及更低数值等级的消息。
步骤 3	**logging monitor** *level* **示例：** `Device(config)#` **logging monitor 3**	限制输出到终端线路的日志消息。 默认时，终端线路接收 debugging 消息以及更低数值等级的消息。
步骤 4	**logging trap** *level* **示例：** `Device(config)#` **logging trap 3**	限制输出到 syslog 服务器的日志消息。 默认时，syslog 服务器接受 informational 消息以及更低数值等级的消息。
步骤 5	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。

限制发送到历史表以及 SNMP 的 Syslog 消息

此任务展示了如何限制发送到历史表以及 SNMP 的 syslog 消息。

总步骤

1. **configure terminal**
2. **logging history** *level*

3. logging history size *number*

4. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**logging history** *level* **示例：** `Device(config)#` **logging history 3**	更改存储到历史表以及发送到 SNMP 服务器的 syslog 消息等级。默认发送 **warnings**、**errors**、**critical**、**alerts** 及 **emergencies** 的消息。
步骤 3	**logging history size** *number* **示例：** `Device(config)#` **logging history size 200**	指定存储在历史表中的消息数量。默认存储 1 条消息，配置范围从 0 到 500。
步骤 4	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。

发送日志消息给 UNIX Syslog 守护进程

此任务是可选的。

注释： 一些新版本的 UNIX syslog 守护进程不再接受来自网络的默认 syslog 数据包。如果使用的系统是这种情况，请使用 **man syslogd** UNIX 命令来决定为了启用远程 syslog 消息记录需要添加或删除哪些 syslog 命令。

在开始前

- 登录为 root。
- 在给 UNIX syslog 服务器发送系统日志消息之前，必须在 UNIX 服务器上配置 syslog 守护进程。

总步骤

1. 在/etc/syslog.conf 文件中添加一行命令。
2. 在 UNIX shell 中输入相关命令。
3. 确保 syslog 守护进程能读到新的配置变化。

具体步骤

	命令或操作	目的
步骤 1	在/etc/syslog.conf 文件中添加一行命令。 **示例：** **local7.debug /usr/adm/logs/*inspur.log***	• **local7**——指定日志记录设置。 • **debug**——指定 syslog 等级。此文件必须已经存在，且 syslog 守护进程必须有权限进行写文件。

步骤 2	在 UNIX shell 中输入相关命令。 **示例：** $ **touch /var/log/*inspur.log*** $ **chmod 666 /var/log/*inspur.log***	创建日志文件。syslog 守护进程会把此等级或更严重等级的消息发送到此文件中。
步骤 3	确保 syslog 守护进程能读到新的配置变化。 **示例：** $ **kill -HUP `cat /etc/syslog.pid`**	更多信息参见UNIX系统的**man syslog.conf**以及**mansyslogd**命令。

监控并维护系统消息日志

监控存档日志的配置

命令	目的
show archive log config {**all** | **number**[*end-number*] | **user** *username* [**session***number*] *number* [*end-number*] | **statistics**}[**provisioning**]	显示整个配置 log 或参数指定的日志。

系统消息日志配置示例

示例：堆栈系统消息

此示例展示了活跃交换机以及一个堆栈成员（*Switch-2*）的部分交换机系统消息：

```
00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/2, changed state to up
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/1,
changedstate to down 2
*Mar 1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
18:47:02: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
*Mar 1 18:48:50.483 UTC: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up (Switch-2)
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet2/0/1, changed state to up (Switch-2)
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet2/0/2, changed state to up (Switch-2)
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
(Switch-2)
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet2/0/1,
changedstate to down 2 (Switch-2)
```

示例：交换机系统消息

此示例展示了交换机上的部分交换机系统消息：

```
00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed
stateto down 2
*Mar 1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
18:47:02: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
*Mar 1 18:48:50.483 UTC: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
```

其他参考资料

相关文档

相关主题	文档标题
系统管理命令	*系统管理命令参考（Inspur 6650 交换机）*
平台无关的命令参考	*配置基础命令参考，Inspur INOS（Inspur 6650 交换机）*
平台无关的配置信息	*IP 编址配置指南库，Inspur INOS（Inspur 6650 交换机）* *配置基础配置指南（Inspur 6650 交换机）*

标准和 RFC

标准/RFC	标题
无	-

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	http://www.icntnetworks.com

系统消息日志的特性历史与信息

版本	特性信息
Inspur INOS 11.3.1	引入了此特性。

配置在线诊断

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用 Inspur 特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问 Inspur 特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于配置在线诊断的信息

在线诊断

在设备连接到现有网络时，可以通过在线诊断功能测试并验证设备的硬件功能。
在线诊断包含数据包交换测试，能检查不同的硬件组件并验证数据通路以及控制信令。
在线诊断能检测以下方面的内容：

- 硬件组件
- 接口（以太网端口等）
- 焊接接头

在线诊断分为按需诊断、计划诊断以及健康监测三类。按需诊断通过 CLI 运行；计划诊断在设备连接到现有网络时按照用户指定的间隔运行或在特定的时间运行；健康监测在后台按照用户定义的间隔运行。默认时，健康监测测试每 30 秒运行一次。
配置了在线诊断之后，可以手动开始诊断测试或显示测试结果。也可以看到已经为设备或交换机堆栈配置了哪些测试，以及已经运行过哪些诊断测试。

如何配置在线诊断

开始在线诊断测试

在设备上配置运行诊断测试之后，可以使用特权 EXEC 命令 **diagnostic start** 执行诊断测试。测试开始后，用户不能停止测试进程。
使用特权 EXEC 命令手动开始在线诊断测试：

总步骤

1. diagnostic start switch *number* **test** {*name* | *test-id* | *test-id-range* | **all** | **basic** | **complete** | **minimal** |**non-disruptive** | **per-port**}

具体步骤

	命令或操作	目的
步骤 1	**diagnostic start switch** *number* **test** {*name*| *test-id* | *test-id-range* | **all** | **basic** | **complete**| **minimal** | **non-disruptive** | **per-port**} **示例：** `Device# diagnostic start switch 2test basic`	开始诊断测试。 **switch** *number* 关键字仅在堆栈设备上支持。范围从 1 到 4。 可以使用以下选项之一指定测试参数： • *name*——输入测试名称。 • *test-id*——输入测试 ID 编号。 • *test-id-range*——输入测试 ID 范围，由逗号及连字符分隔的整数表示。 • **all**——开始所有测试。 • **basic**——开始基本测试集。 • **complete**——开始完整测试集。 • **minimal**——开始最小启动测试集。 • **non-disruptive**——开始无干扰测试集。 • **per-port**——开始基于端口的测试集。

配置在线诊断

在启用诊断监控之前，必须配置失败门限值以及测试间隔。

计划在线诊断

可以计划设备在线诊断的运行时间，在一天中指定的时间运行测试，或每天、每周或每月运行一次。使用命令的 **no** 形式移除计划。

总步骤

1. configure terminal

2. diagnostic schedule switch *number* **test** {*name* | *test-id* | *test-id-range* | **all** | **basic** | **complete** | **minimal**| **non-disruptive** | **per-port**} {**daily** | **on** *mm dd yyyy hh:mm* | **port** *inter-port-number port-number-list* |**weekly** *day-of-week hh:mm*}

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**diagnostic schedule switch** *number* **test**{*name* | *test-id* | *test-id-range* | **all** | **basic** |**complete** | **minimal** | **non-disruptive** |**per-port**} {**daily** | **on** *mm dd yyyy hh:mm* |**port** *inter-port-number port-number-list* |**weekly** *day-of-week hh:mm*} **示例：** `Device(config)#` **diagnostic scheduleswitch 3 test 1-5 on July 3 2013 23:10**	计划在指定的日期和时间执行按需诊断测试。 **switch** *number* 关键字仅在堆栈设备上支持。范围从 1 到 4。 可以使用以下选项之一指定测试计划参数： • *name*——命令 **show diagnostic content** 输出显示的测试名称。 • *test-id*——命令 **show diagnostic content** 输出显示的测试 ID 编号。 • *test-id-range* ——命令 **show diagnostic content** 输出显示的测试 ID 编号。 • **all**——所有测试 ID。 • **basic**——开始基本按需诊断测试。 • **complete**——开始完整测试集。 • **minimal**——开始最小启动测试集。 • **non-disruptive**——开始无干扰测试集。 • **per-port**——开始基于端口的测试集。 可以计划测试执行时间： • 每天——使用 **daily** *hh:mm* 参数。 • 指定日志与时间——使用 **on** *mm dd yyyy hh:mm* 参数。 • 每周——使用 **weekly** *day-of-week hh:mm* 参数。

配置健康监测诊断

可以在设备连接到现有网络时配置设备进行健康监测诊断测试。可以配置每个健康监测测试的执行间隔，让设备产生测试失败 syslog 消息，并启用特定的测试。
使用此命令的 **no** 形式来禁用测试。
默认情况下，健康监测被禁用，但是设备会在测试失败时产生 syslog 消息。
按照以下步骤配置并启用健康监测诊断测试：

总步骤

1. enable

2. configure terminal

3. diagnostic monitor interval switch *number* **test** {*name* | *test-id* | *test-id-range* | **all**} *hh:mm:ss milliseconds day*

4. diagnostic monitor syslog

5. diagnostic monitor threshold switch *number* **test** {*name* | *test-id* | *test-id-range* | **all**} **failure count** *count*

6. diagnostic monitor switch *number* **test** {*name* | *test-id* | *test-id-range* | **all**}

7. end

8. show running-config

9. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 3	**diagnosticmonitor interval switch** *number***test** {*name* | *test-id* | *test-id-range* | **all**}*hh:mm:ss milliseconds day* **示例：** `Device(config)#` **diagnostic monitorinterval switch 2 test 1 12:30:00 7505**	配置指定健康监测测试的执行间隔。 **switch** *number* 关键字仅在堆栈设备上支持。范围从 1 到 4。 可以使用以下选项之一指定测试计划参数： • *name*——命令 **show diagnostic content** 输出显示的测试名称。 • *test-id*——命令 **show diagnostic content** 输出显示的测试 ID 编号。 • *test-id-range* ——命令 **show diagnostic content** 输出显示的测试 ID 编号。 • **all**——所有诊断测试。 指定间隔请设置以下参数： • *hh:mm:ss*——按照小时、分钟、秒的形式设置监控间隔。*hh* 的

		范围从 0 到 24，*mm* 和 *ss* 的范围从 0 到 60。 • *milliseconds*——毫秒格式（ms）的监控间隔。范围从 0 到 999。 • *day*——监控间隔天数。范围从 0 到 20。
步骤 4	**diagnostic monitor syslog** **示例：** `Device(config)#` **diagnostic monitorsyslog**	（可选）配置交换机在健康监测测试失败时生成 syslog 消息。
步骤 5	**diagnostic monitor threshold switch***number* **test** {*name* | *test-id* | *test-id-range* |**all**} **failure count** *count* **示例：** `Device(config)#` **diagnostic monitorthreshold switch 2 test 1 failurecount 20**	（可选）设置健康监测测试的故障门限值。 **switch** *number* 关键字仅在堆栈交换机上支持。范围从 1 到 9。 可以使用以下选项之一指定测试计划参数： • *name*——命令 **show diagnostic content** 输出显示的测试名称。 • *test-id*——命令 **show diagnostic content** 输出显示的测试 ID 编号。 • *test-id-range* ——命令 **show diagnostic content** 输出显示的测试 ID 编号。 • **all**——所有诊断测试。 失败门限值 *count* 范围从 0 到 99。
步骤 6	**diagnostic monitor switch** *number* **test**{*name* | *test-id* | *test-id-range* | **all**} **示例：** `Device(config)#` **diagnostic monitorswitch 2 test 1**	启用指定的健康监测测试。 **switch** *number* 关键字仅在堆栈交换机上支持。范围从 1 到 9。 可以使用以下选项之一指定测试计划参数： • *name*——命令 **show diagnostic content** 输出显示的测试名称。 • *test-id*——命令 **show diagnostic content** 输出显示的测试 ID 编号。 • *test-id-range* ——命令 **show diagnostic content** 输出显示的测试 ID 编号。 • **all**——所有诊断测试。
步骤 7	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 8	**show running-config** **示例：**	验证配置的条目。

	Device# **show running-config**	
步骤 9	**copy running-config startup-config** **示例：** Device# **copy running-configstartup-config**	（可选）把配置的条目保存到配置文件中。

接下来做什么？

使用全局配置命令**no diagnostic monitor interval test***test-id* | *test-id-range* }把间隔更改为默认值或者0。使用**no diagnostic monitor syslog**命令禁止在健康监测测试失败时产生syslog消息。使用**diagnostic monitor threshold test***test-id* |*test-id-range* }**failure count**命令移除失败门限值。

监控及维护在线诊断

显示在线诊断测试及测试结果

可以显示为设备或设备堆栈配置的在线诊断测试，并可以使用下表中的特权 EXEC **show** 命令检查测试结果：

表 201：诊断测试配置及结果的命令

命令	目的
show diagnostic content switch [*number* | **all**]	显示为交换机配置的在线诊断测试。 **switch** [*number* | **all**]参数仅支持在堆栈交换机上使用。
show diagnostic status	显示当前运行的诊断测试。
show diagnostic result switch [*number* | **all**] [**detail**| **test** {*name* | *test-id* | *test-id-range* | **all**} [**detail**]]	显示在线诊断测试结果。 **switch** [*number* | **all**]参数仅支持在堆栈交换机上使用。
show diagnostic switch [*number* | **all**] [**detail**]	显示在线诊断测试结果。 **switch** [*number* | **all**]参数仅支持在堆栈交换机上使用。
show diagnostic schedule switch [*number* | **all**]	显示计划在线诊断测试结果。 **switch** [*number* | **all**]参数仅支持在堆栈交换机上使用。
show diagnostic post	显示 POST 结果（输出与 **show post** 命令输出相同）。

在线诊断测试配置示例

示例：开始诊断测试

此示例展示了如何通过测试名称开始诊断测试：

```
Device# diagnostic start switch 2 test TestInlinePwrCtlr
```

此示例展示了如何开始所有基本诊断测试：

```
Device# diagnostic start switch 1 test all
```

示例：配置健康监测测试

此示例展示了如何配置健康监测测试：

```
Device(config)# diagnostic monitor threshold switch 1 test 1 failure count 50
Device(config)# diagnostic monitor interval switch 1 test TestPortAsicStackPortLoopback
```

示例：计划诊断测试

此示例展示了如何在特定交换机上配置计划诊断测试在指定的日期与时间运行：

```
Device(config)# diagnostic schedule test DiagThermalTest on June 3 2013 22:25
```

此示例展示了如何在特定交换机上配置计划诊断测试每周在特定时间运行：

```
Device(config)# diagnostic schedule switch 1 test 1,2,4-6 weekly saturday 10:30
```

示例：显示在线诊断

此示例展示了如何展示按需诊断设置：

```
Device# show diagnostic ondemand settings
Test iterations = 1
Action on test failure = continue
```

此示例展示了如何显示诊断事件错误：

```
Device# show diagnostic events event-type error
Diagnostic events (storage for 500 events, 0 events recorded)
Number of events matching above criteria = 0
No diagnostic log entry exists.
```

此示例展示了如何显示诊断测试的描述：

```
Device# show diagnostic description switch 1 test all
DiagGoldPktTest :
The GOLD packet Loopback test verifies the MAC level loopback
functionality. In this test, a GOLD packet, for which doppler
provides the support in hardware, is sent. The packet loops back
at MAC level and is matched against the stored packet. It is a non
-disruptive test.
DiagThermalTest :
This test verifies the temperature reading from the sensor is below the yellow
temperature threshold. It is a non-disruptive test and can be run as a health
monitoring test.
DiagFanTest :
This test verifies all fan modules have been inserted and working properly on the
board
It is a non-disruptive test and can be run as a health monitoring test.
DiagPhyLoopbackTest :
The PHY Loopback test verifies the PHY level loopback
functionality. In this test, a packet is sent which loops back
```

```
at PHY level and is matched against the stored packet. It is a
disruptive test and cannot be run as a health monitoring test.
DiagScratchRegisterTest :
The Scratch Register test monitors the health of application-specific
integrated circuits (ASICs) by writing values into registers and reading
back the values from these registers. It is a non-disruptive test and can
be run as a health monitoring test.
DiagPoETest :
This test checks the PoE controller functionality. This is a disruptive test
and should not be performed during normal switch operation.
DiagStackCableTest :
This test verifies the stack ring loopback functionality
in the stacking environment. It is a disruptive test and
cannot be run as a health monitoring test.
DiagMemoryTest :
This test runs the exhaustive ASIC memory test during normal switch operation
NG3K utilizes mbist for this test. Memory test is very disruptive
in nature and requires switch reboot after the test.
Device#
```

此示例展示了如何显示启动等级：

```
Device# show diagnostic bootup level
Current bootup diagnostic level: minimal
Device#
```

其他参考资料

相关文档

相关主题	文档标题
系统管理命令	*系统管理命令参考（Inspur 6650 交换机）*
平台无关的命令参考	*配置基础命令参考，Inspur INOS（Inspur 6650 交换机）*
平台无关的配置信息	*IP 编址配置指南库，Inspur INOS（Inspur 6650 交换机）* *配置基础配置指南（Inspur 6650 交换机）*

标准和 RFC

标准/RFC	标题
无	-

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档	http://www.icntnetworks.com

及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	

配置在线诊断的特性历史与信息

版本	特性信息
Inspur INOS 11.3.1	引入了此特性。

管理配置文件

管理配置文件的前提

- 用户至少应该基本熟悉 Inspur INOS 环境以及命令行界面。
- 用户设备上至少应该运行着最少配置。可以使用 **setup** 命令创建基本的配置文件。

管理配置文件的限制

- 此文档中描述的许多 Inspur INOS 命令仅在设备的特定配置模式中可用。
- 一些 Inspur INOS 配置命令仅在特定的设备平台上可用，且命令语法在不同平台上可能有所不同。

关于管理配置文件的信息

配置文件中包含用于定义 Inspur 设备功能的 Inspur 软件命令。系统启动（从 startup-config 文件读取）或当用户在 CLI 的配置模式中输入命令时，命令会被 Inspur INOS 软件解析（翻译并执行）。

启动配置文件（startup-config）在系统启动时被用来配置系统软件。运行配置文件（running-config）包含系统的当前配置。这两个配置文件可以不同。比如，如果希望在短期内更改配置，用户可以使用 EXEC 命令 **configure terminal** 更改运行配置，但不使用 EXEC 命令

copy running-config startup-config 保存配置。
要更改运行配置，请使用 **configure terminal** 命令，如*修改配置文件（CLI）*一节所述。使用 Inspur INOS 配置模式时，命令通常会立即执行，且会在输入命令后立即保存到运行配置文件，或在退出配置模式时保存。
要更改启动配置文件，用户可以使用 EXEC 命令 **copy running-config startup-config** 把运行配置文件保存到启动配置中，也可以把配置文件从文件服务器拷贝到启动配置中（参见*把配置文件从 TFTP 服务器拷贝到设备（CLI）*）。

配置模式及选择配置源

要在设备上进入配置模式，请在特权 EXEC 提示符后输入 **configure** 命令。Inspur INOS 软件会返回以下提示，请求用户指定使用终端（terminal）、内存（memory）或网络服务器上储存的文件（network）作为配置命令来源：

```
Configuring from terminal, memory, or network [terminal]?
```

通过终端进行配置允许用户在命令行中输入配置命令，如下一节所述。更多信息参见*重新执行启动配置文件中的配置命令（CLI）*。
通过网络进行配置允许用户通过网络加载并执行配置命令。更多信息参见*把配置文件从 TFTP 服务器拷贝到设备（CLI）*。

使用 CLI 更改配置文件

Inspur INOS软件支持每行使用一条配置命令。可以按需输入任意数量的命令。可以向配置文件中添加输入命令的注释描述，注释前使用一个感叹号（!）。因为注释不会保存到NVRAM或配置文件的活跃拷贝中，在用户使用EXEC命令**show running-config**或**more system:running-config**时，注释不会出现。使用**show startup-config**或**morenvram:startup-config** EXEC命令列出启动配置时，注释也不会被显示。在配置文件加载到设备上时，注释会被去除。然而，可以查看储存到文件传输协议（File Transfer Protocol，FTP）服务器，远程拷贝协议（RemoteCopy Protocol，RCP）服务器或简单文件传输协议（Trivial File Transfer Protocol，TFTP）服务器上的配置文件注释。使用CLI配置系统软件时，命令会在输入后执行。

配置文件的位置

配置文件会被保存到以下位置：

- 运行配置被保存到 RAM 中。
- 在除了 A 类闪存文件系统平台以外的所有平台上，启动配置会被储存到非易失性随机存取存储器（非易失性随机存取存储器，NVRAM）中。
- 在 A 类闪存文件系统平台上，启动配置会被保存到 CONFIG_FILE 环境变量指定的位置（参见*指定 A 类闪存文件系统的 CONFIG_FILE 环境变量（CLI）*）。CONFIG_FILE 变量默认使用 NVRAM，也可以使用以下文件系统中的文件：
  - **nvram:**（NVRAM）
  - **bootflash:**（内部闪存内存）
  - **usbflash0:**（闪存文件系统）

把配置文件从网络服务器拷贝到设备

可以把 TFTP、RCP 或 FTP 服务器上的配置文件拷贝到设备的运行配置或启动配置中。执行此操作的原因可能有：

- 恢复备份的配置文件。
- 使用另一台设备的配置文件。例如，向网络中添加了新设备，并希望其配置与原始设备类似。通过把文件拷贝到新设备上，管理员可以仅更改相关的部分，而不用重新创建整个配置文件。
- 把相同的配置命令加载到网络的所有设备上，让所有设备有相似的配置。

EXEC 命令 **copy{ftp: | rcp: | tftp:system:running-config}**会把配置文件加载到设备上，就好像在命令行输入命令一样。在添加命令前设备不会擦除现有的运行配置。如果拷贝的配置文件命令替换了现有配置文件中的命令，现有命令会被擦除。比如，如果拷贝的配置文件中有某条命令的 IP 地址与现有配置不同，拷贝配置的 IP 地址会被使用。然而，现有配置中的一些命令可能不会被代替。此时，最终生成的配置文件会混合使用现有配置以及拷贝的配置，且优先使用拷贝的配置。

要把配置文件完全恢复为服务器上保存文件的拷贝，用户需要直接把配置文件拷贝到启动文件中（使用 **copy ftp:| rcp:| tftp:} nvram:startup-config** 命令）并重启设备。

要把服务器上的配置文件拷贝到设备上，请执行下一节所述的任务。

使用哪种协议取决于使用哪种类型的服务器。FTP 以及 RCP 传输机制能提供比 TFTP 更快的性能及更可靠的传输能力。这些性能提升的原因是 FTP 以及 RCP 传输机制是基于 TCP/IP 协议栈构建的，是面向连接的。

把配置文件从设备拷贝到 TFTP 服务器

在一些 TFTP 实现中，用户在拷贝文件之前必须在 TFTP 服务器上创建一个空文件，并授予读写以及执行权限。更多信息参见使用的 TFTP 文档。

把配置文件从设备拷贝到 RCP 服务器

用户可以把配置文件从设备拷贝到 RCP 服务器上。

UNIX 社区把网络作为资源使用的初次尝试，推动了远程 Shell 协议的设计与实现，其中就包括远程 Shell（remote shell，rsh）以及远程拷贝（remote copy，rcp）功能。rsh 和 rcp 让用户可以在网络上远程执行命令，并在本地与远程主机或服务器的文件系统之间拷贝文件。Inspur 的 rsh 与 rcp 实现可以与标准的实现进行互操作。

rcp 的 **copy** 命令依赖于远程系统上的 rsh 服务器（或守护进程）。要使用 rcp 拷贝文件，用户需创建一个用于文件分发的服务器，这与对 TFTP 服务器的操作相同。用户仅需要对支持远程 Shell（rsh）的服务器有访问权限（多数 UNIX 系统都支持 rsh）。因为要把文件从一个地方拷贝到另一个地方，用户必须对源文件有读权限，对目的文件有写权限。如果目的文件不存在，rcp 或创建该文件。

虽然 Inspur 的 rcp 实现模仿了 UNIX rcp 实现的功能——在网络上的系统之间拷贝文件——但是 Inspurrcp 命令语法与 UNIX rcp 命令语法不同。

Inspur 的 rcp 支持使用 rcp 作为一组 **copy** 命令的传输机制。这些 rcp 的 **copy** 命令风格与 Inspur TFTP **copy** 命令类似，但是能提供更快的性能以及更可靠的数据传输服务。这些性能提升的原因是 RCP 传输机制是基于 TCP/IP 协议栈构建的，是面向连接的。可以使用 rcp 命令在设备和网络服务器之间拷贝系统镜像以及配置文件。

也可以启用 rcp 支持，允许远程系统上的用户在设备上拷贝文件。

要配置Inspur INOS软件允许远程用户在设备上来回拷贝文件，请使用全局配置命令**iprcmd rcp-enable**。

RCP 用户名的要求

RCP 协议要求客户端在每个 RCP 请求中向服务器发送远程用户名。使用 RCP 把配置文件从设备拷贝到服务器时，Inspur INOS 软件会发送按照以下顺序遇到的第一个合法用户名：

1 **copy** EXEC 命令中指定的用户名。

2 全局配置命令 **ip rcmd remote-username** 设置的用户名。

3 与当前 tty（terminal）进程关联的远程用户名。比如，如果用户通过 Telnet 连接设备，并通过命令 **username** 进行了验证，设备软件会把 Telnet 用户名当作远程用户名发送。

4 设置主机名。

要使 RCP 拷贝请求能成功执行，网络服务器上必须为远程用户名定义了一个账户。如果服务器上有对应的目录结构，配置文件或镜像会到拷贝到服务器上与远程用户名关联的目录中，或从目录中拷贝出。例如，如果系统镜像位于服务器上用户的 home 目录下，可以把改用户名指定为远程用户名。

更多信息参见使用的 RCP 服务器文档。

使用命令 **ip rcmd remote-username** 指定所有拷贝使用的用户名（rcmd 是一个 UNIX 程序，在超级用户级别使用，可以使用基于预留端口号的认证机制在远程机器上执行命令，rcmd 表示“remotecommand”）。如果仅希望给特定的拷贝操作指定用户名，请在 **copy** 命令中包含用户名。

如果要向服务器写文件，必须正确配置 RCP 服务器接收来自设备用户的 RCP 写请求。对于 UNIX 系统，必须在 RCP 服务器上的.rhosts 文件中为远程用户添加一个条目。例如，假设设备有以下配置：

```
hostname Device1
ip rcmd remote-username User0
```

如果设备的 IP 能翻译成 device1.example.com，则 RCP 服务器上 User0 的.rhosts 文件应包含以下配置：

```
Device1.example.com Device1
```

把配置文件从设备拷贝到 FTP 服务器

可以把配置文件从设备拷贝到 FTP 服务器上。

理解 FTP 用户名及密码

FTP 协议要求客户端在每个 FTP 请求中向服务器发送远程用户名及密码。使用 FTP 把配置文件从设备拷贝到服务器时，Inspur INOS 软件会发送按照以下顺序遇到的第一个合法用户名：

1 **copy** EXEC 命令中指定的用户名。

2 全局配置命令 **ip ftp username** 设置的用户名。

3 匿名。

设备会发送按照以下顺序遇到的第一个合法密码：

1 **copy** 命令中指定的密码。

2 命令 **ip ftp password** 设置的密码。

3 设备产生的密码 *username* @*devicename*.*domain*。*username* 变量是与当前会话关联的用户名，*devicename* 是配置的主机名，*domain* 是设备的域名。

用户名与密码必须与 FTP 服务器上的账户关联。如果向服务器写文件，必须正确配置 FTP 服务器接受来自设备用户的 FTP 写请求。如果服务器上有对应的目录结构，配置文件或镜像会到拷贝到服务器上与远程用户名关联的目录中，或从目录中拷贝出。例如，如果系统镜像位于服务器上用户的 home 目录下，可以把改用户名指定为远程用户名。

更多信息参见使用的 FTP 服务器文档。

使用 **ip ftp username** 和 **ip ftp password** 全局配置命令为所有拷贝命令指定用户名和密码。如果仅希望给特定的拷贝操作指定用户名，请在 **copy**EXEC 命令中包含用户名。

通过 VRF 拷贝文件

可以通过 **copy** 命令中指定的 VRF 接口拷贝文件。在 **copy** 命令中指定 VRF 更简单且更高效，因为用户无需请求更改配置就可以直接更改源接口。

以下示例展示了如何使用 **copy** 命令通过 VRF 拷贝文件：

```
Device# copy scp: flash-1: vrf test-vrf
Address or name of remote host [10.1.2.3]?
Source username [ScpUser]?
Source filename [/auto/tftp-server/ScpUser/vrf_test.txt]?
Destination filename [vrf_test.txt]?
Getting the vrf name as test-vrf
Password:
Sending file modes: C0644 10 vrf_test.txt
!
223 bytes copied in 22.740 secs (10 bytes/sec)
```

把配置文件从一台交换机拷贝到另一台交换机

可以把配置从一台交换机拷贝到另一台交换机。这是一个 2 步的过程——把配置文件从交换机拷贝到 TFTP 服务器，然后从 TFTP 拷贝到另一台交换机上。

要从交换机上拷贝当前的配置，运行 **copy startup-config tftp:**命令。执行命令会把配置拷贝到 TFTP 服务器上。

接着，登录到另一台交换机上并运行 **copy tftp: startup-config** 命令。配置此时会被拷贝到另一台交换机上。

在配置拷贝完成后，使用 **write memory** 命令保存配置，然后重启交换机或者运行 **copy startup-config running-config** 命令。

更多信息参见 *Inspur INOS 配置基础命令参考，Inspur INOS（Inspur 3850 交换机）*。

拷贝比 NVRAM 空间大的文件

要维护超出 NVRAM 大小的配置文件，请查看以下内容。

压缩配置文件

全局配置命令 **service compress-config** 会指定在 NVRAM 中压缩存储配置文件。配置文件被压缩后，设备就能正常工作。系统重启时会识别配置文件被压缩，系统会扩展该文件并正常执行操作。EXEC 命令 **more nvram:startup-config** 会在显示配置之前进行扩展。

压缩配置文件之前，请查阅对应的硬件安装及维护说明，验证系统 ROM 是否支持文件压缩。如果不支持，可以安装支持文件压缩的新 ROM。

配置的大小不能超过 NVRAM 大小的三倍。对于 128KB 大小的 NVRAM，最大展开配置文件的大小是 384 KB。

只有 Inspur INOS 10.0 以及之后版本的引导 ROM 支持全局配置命令 **service compress-config**。安装新 ROM 仅需在 ROM 中没有 Inspur INOS 10.0 版时执行一次。如果引导 ROM 没有识别压缩的配置，会显示以下信息：

```
Boot ROMs do not support NVRAM compression Config NOT written to NVRAM
```

在 A 类闪存文件系统中把配置保存到闪存内存

在 A 类闪存文件系统的设备上，可以把启动配置保存在闪存内存中。设置环境变量 CONFIG_FILE 可以把启动配置保存在内部闪存内存或 PCMCIA 插槽上的闪存内存。

更多信息参见*指定 A 类闪存文件系统的 CONFIG_FILE 环境变量（CLI）*。

编辑或更改大型配置文件时要小心。每次输入 EXEC 命令 **copy system:running-config nvram:startup-config** 时闪存空间都会被使用。因为闪存的文件管理（如优化空闲空间）不会自动进行，必须小心注意可用的闪存空间。使用 **squeeze** 命令来回收已使用的空间。建议使用至少 20 MB 的大容量闪存卡。

通过网络加载配置命令

可以把大型配置文件保存在 FTP、RCP 或 TFTP 服务器上，并在系统启动时进行下载。要使用网络服务器保存大型配置，更多信息参见*把配置文件从设备拷贝到 TFTP 服务器（CLI）*以及*配置设备下载配置文件*。

配置设备下载配置文件

可以配置设备在系统启动时加载一个或两个配置文件。配置文件会被加载到内存中并被读入，就好像在命令行中输入的命令一样。此时，设备的配置文件会混用原始启动配置以及一个或两个下载的配置文件。

网络配置文件与主机配置文件

出于历史原因，设备下载的第一个文件被称为网络配置文件。设备下载的第二个配置文件被称为主机配置文件。当网络上的所有设备都使用许多相同的命令时，可以使用这两个配置文件。网络配置文件包含为所有设备配置的标准命令。主机配置文件包含特定于一台主机的命令。如果加载这两个文件，应该优先使用主机配置文件。网络配置文件以及主机配置文件必须都位于通过 TFTP、RCP 或 FTP 可达的网络服务器上，且必须都可读。

如何管理配置文件信息

显示配置文件信息（CLI）

要显示配置文件的信息，请完成本节配置任务：

总步骤

1. **enable**
2. **show boot**
3. **more** *file-url*
4. **show running-config**
5. **show startup-config**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。

步骤 2	**show boot** **示例：** `Device# show boot`	显示设置的 BOOT 环境变量内容，CONFIG_FILE 环境变量指向的配置文件名，以及 BOOTLDR 环境变量的内容。
步骤 3	**more** *file-url* **示例：** `Device# more 10.1.1.1`	显示指定文件的特性。
步骤 4	**show running-config** **示例：** `Device# show running-config`	显示运行配置文件的内容（等同于 **more system:running-config** 命令）
步骤 5	**show startup-config** **示例：** `Device# show startup-config`	显示启动配置文件的内容（等同于 **more nvram:startup-config** 命令） 在除了 A 类闪存文件系统平台以外的所有平台上，默认的 startup-config 文件通常被保存在 NVRAM 中。 在 A 类闪存文件系统平台上，CONFIG_FILE 环境变量指向默认的 startup-config 文件。 CONFIG_FILE 变量默认值指向 NVRAM。

修改配置文件（CLI）

Inspur INOS软件支持每行使用一条配置命令。可以按需输入任意数量的命令。可以向配置文件中添加输入命令的注释描述，注释前使用一个感叹号（!）。因为注释不会保存到NVRAM或配置文件的活跃拷贝中，在用户使用EXEC命令**show running-config**或**more system:running-config**时，注释不会出现。使用**show startup-config**或**morenvram:startup-config** EXEC命令列出启动配置时，注释也不会被显示。在配置文件加载到设备上时，注释会被去除。然而，可以查看储存到文件传输协议（File Transfer Protocol，FTP）服务器，远程拷贝协议（RemoteCopy Protocol，RCP）服务器或简单文件传输协议（Trivial File Transfer Protocol，TFTP）服务器上的配置文件注释。使用CLI配置系统软件时，命令会在输入后执行。要使用CLI配置系统软件，请在特权EXEC模式中使用以下命令：

总步骤

1. **enable**
2. **configure terminal**
3. **configuration command**
4. 进行以下操作之一：
   - **end**
   - **^Z**
5. **copy system:running-config nvram:startup-config**

具体步骤

	命令或操作	目的

步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 3	**configuration command** **示例：** `Device(config)# configuration command`	输入必要的配置命令。Inspur INOS 文档集描述了按照不同技术组织的配置命令。
步骤 4	进行以下操作之一： • **end** • **^Z** **示例：** `Device(config)# end`	结束配置会话并退出到 EXEC 模式。 **注释：**同时按下 Ctrl 和 Z 键时，屏幕上会显示^Z。
步骤 5	**copy system:running-confignvram:startup-config** **示例：** `Device# copy system:running-confignvram:startup-config`	把运行配置保存为启动配置文件。也可以使用 **copy running-config startup-config** 命令，但是用户应该要知道此命令时不太准确的。在多数平台上，此命令会把配置保存到 NVRAM 中。在 A 类闪存文件系统平台上，此步骤会把配置保存到 CONFIG_FILE 环境变量指定的位置（默认的变量 CONFIG_FILE 指定文件应该被保存到 NVRAM）。

示例

以下示例配置了设备提示符名称。由感叹号（!）标注的注释不会执行任何命令。**hostname** 命令被用来把设备名从 device 改为 new_name。按下 Ctrl-Z（^Z）或输入 **end** 命令，用户会退出配置模式。**copy system:running-config nvram:startup-config** 命令会把当前配置保存为启动配置。

```
Device# configure terminal
Device(config)# !The following command provides the switch host name.
Device(config)# hostname new_name
new_name(config)# end
new_name# copy system:running-config nvram:startup-config
```

启动配置使用 NVRAM 时，系统会把当前配置信息按照配置命令文本形式存储，并只记录非默认的配置。系统会对内存文件计算校验和，以防止数据损坏。

注释： 一些特定的命令可能不会被保存到 NVRAM 中。重启机器后需要再次输入这些命令。这些文档在文档中标示出。建议用户保存一个这些命令的列表，以在设备重启后快速重新配置。

把配置文件从设备拷贝到 TFTP 服务器（CLI）

要把配置从设备拷贝到 TFTP 服务器，请完成本节的配置任务：

总步骤

1. enable

2. copy system:running-config tftp: [[[**//***location*]**/***directory*]**/***filename*]

3. copy nvram:startup-config tftp: [[[**//***location*]**/***directory*]**/***filename*]

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**copy system:running-config tftp:** [[[**//***location*]**/***directory*]**/***filename*] **示例：** `Device# copy system:running-config tftp://server1/topdir/file10`	把运行配置文件拷贝到 TFTP 服务器。
步骤 3	**copy nvram:startup-config tftp:** [[[**//***location*]**/***directory*]**/***filename*] **示例：** `Device# copy nvram:startup-config tftp://server1/1stdir/file10`	把启动配置文件拷贝到 TFTP 服务器。

示例

以下示例把配置文件从设备拷贝到 TFTP 服务器：

```
Device# copy system:running-config tftp://172.16.2.155/tokyo-confg
Write file tokyo-confg on host 172.16.2.155? [confirm] Y
Writing tokyo-confg!!! [OK]
```

接下来做什么？

输入 **copy** 命令后，系统可能提示用户输入额外信息或让用户确认操作。显示的提示取决于用户在 **copy** 命令中提供了多少信息以及当前的全局配置命令 **file prompt** 设置。

把配置文件从设备拷贝到 RCP 服务器（CLI）

要把启动配置文件或运行配置文件从设备拷贝到 RCP 服务器，请在特权 EXEC 模式中使用以下命令配置：

总步骤

1. enable

2. configure terminal

3. ip rcmd remote-username *username*

4. end

5. 进行以下操作之一：

- **copy system:running-config rcp:** [[[**//**[*username***@**]*location*]**/***directory*]**/***filename*]

- **copy nvram:startup-config rcp:** [[[**//**[*username***@**]*location*]**/***directory*]**/***filename*]

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 3	**ip rcmd remote-username** *username* **示例：** `Device(config)# ip rcmd remote-username NetAdmin1`	（可选）更改默认远程用户名。
步骤 4	**end** **示例：** `Device(config)# end`	（可选）退出全局配置模式。
步骤 5	进行以下操作之一： • **copy system:running-config rcp:** [[[**//**[*username***@**]*location*]**/***directory*]**/***filename*] • **copy nvram:startup-config rcp:** [[[**//**[*username***@**]*location*]**/***directory*]**/***filename*] **示例：** `Device# copy system:running-config rcp://NetAdmin1@example.com/dir-files/file1`	• 指定把设备运行配置文件保存到 RCP 服务器上 或 • 指定把设备启动配置文件保存到 RCP 服务器上

示例

在 RCP 服务器上保存运行配置文件

以下示例把名为 runfile2-confg 的运行配置文件保存到 IP 地址为 172.16.101.101 的远程主机的 netadmin1 目录下：

```
Device# copy system:running-config rcp://netadmin1@172.16.101.101/runfile2-confg
Write file runfile2-confg on host 172.16.101.101?[confirm]
Building configuration...[OK]
Connected to 172.16.101.101
Device#
```

在 RCP 服务器上保存启动配置文件

以下示例展示了使用 RCP 把启动配置文件拷贝到服务器上：

```
Device# configure terminal
Device(config)# ip rcmd remote-username netadmin2
Device(config)# end
Device# copy nvram:startup-config rcp:
Remote host[]? 172.16.101.101
Name of configuration file to write [start-confg]?
Write file start-confg on host 172.16.101.101?[confirm]
```

```
![OK]
```

接下来做什么？

输入 **copy** 命令后，系统可能提示用户输入额外信息或让用户确认操作。显示的提示取决于用户在 **copy** 命令中提供了多少信息以及当前的全局配置命令 **file prompt** 设置。

把配置文件从设备拷贝到 FTP 服务器（CLI）

要把启动配置文件或运行配置文件从设备拷贝到 FTP 服务器上，请完成以下配置任务：

总步骤

1. **enable**
2. **configure terminal**
3. **ip ftp username** *username*
4. **ip ftp password** *password*
5. **end**
6. 进行以下操作之一：
   - **copy system:running-config ftp:** [[[**//**[*username* [**:***password*]**@**]*location*]**/***directory*]**/***filename*]
   - **copy nvram:startup-config ftp:** [[[**//**[*username* [**:***password*]**@**]*location*]**/***directory*]**/***filename*]

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 3	**ip ftp username** *username* **示例：** `Device(config)# ip ftp username NetAdmin1`	（可选）指定默认远程用户名。
步骤 4	**ip ftp password** *password* **示例：** `Device(config)# ip ftp password adminpassword`	（可选）指定默认密码。
步骤 5	**end** **示例：** `Device(config)# end`	（可选）退出全局配置模式。只有覆盖了默认的远程用户名或密码时（见步骤 2 和步骤 3）才需要进行此步骤。
步骤 6	进行以下操作之一： • **copy system:running-config ftp:** [[[**//**[*username*[**:***password*]**@**]*location*]**/***directory*]**/***filename*] 或 • **copy nvram:startup-config ftp:**	把运行配置或启动配置拷贝到 FTP 服务器上的指定位置。

	[[[**//**[*username*[**:***password*]**@**]*location*]**/***directory*]**/***filename*] **示例：** `Device# copy system:running-config ftp:`	

示例

在 FTP 服务器上保存运行配置文件

以下示例把名为 runfile-confg 的运行配置文件保存到 IP 地址为 172.16.101.101 的远程主机的 netadmin1 目录下：

```
Device# copy system:running-config ftp://netadmin1:mypass@172.16.101.101/runfile-confg
Write file runfile-confg on host 172.16.101.101?[confirm]
Building configuration...[OK]
Connected to 172.16.101.101
Device#
```

在 FTP 服务器上保存启动配置文件

以下示例展示了使用 FTP 把启动配置文件拷贝到服务器上：

```
Device# configure terminal
Device(config)# ip ftp username netadmin2
Device(config)# ip ftp password mypass
Device(config)# end
Device# copy nvram:startup-config ftp:
Remote host[]? 172.16.101.101
Name of configuration file to write [start-confg]?
Write file start-confg on host 172.16.101.101?[confirm]
![OK]
```

接下来做什么？

输入 **copy** 命令后，系统可能提示用户输入额外信息或让用户确认操作。显示的提示取决于用户在 **copy** 命令中提供了多少信息以及当前的全局配置命令 **file prompt** 设置。

把配置文件从 TFTP 服务器拷贝到设备（CLI）

要把配置文件从 TFTP 服务器拷贝到设备上，请完成以下配置任务：

总步骤

1. enable

2. copy tftp: [[[**//***location*]**/***directory*]**/***filename*] **system:running-config**

3. copy tftp: [[[**//***location*]**/***directory*]**/***filename*] **nvram:startup-config**

4. copy tftp: [[[**//***location*]**/***directory*]**/***filename*]**flash-[n]:/directory/startup-config**

具体步骤

	命令或操作	目的
步骤 1	**enable**	启用特权 EXEC 模式，在提示时输入

	示例： `Device> enable`	密码。
步骤 2	**copy tftp:** [[[**//***location*]**/***directory*]**/***filename*] **system:running-config** **示例：** `Device# copy tftp://server1/dir10/datasource system:running-config`	把配置文件从 TFTP 服务器拷贝到运行配置中。
步骤 3	**copy tftp:** [[[**//***location*]**/***directory*]**/***filename*] **nvram:startup-config** **示例：** `Device# copy tftp://server1/dir10/datasource nvram:startup-config`	把配置文件从 TFTP 服务器拷贝到启动配置中。
步骤 4	**copy tftp:** [[[**//***location*]**/***directory*]**/***filename*]**flash-[n]:/directory/startup-config** **示例：** `Device# copy tftp://server1/dir10/datasource flash:startup-config`	把配置文件从 TFTP 服务器拷贝到启动配置中。

示例

以下示例使用 IP 地址 172.16.2.155 上的 tokyo-confg 文件配置系统：

```
Device# copy tftp://172.16.2.155/tokyo-confg system:running-config
Configure using tokyo-confg from 172.16.2.155? [confirm] Y
Booting tokyo-confg from 172.16.2.155:!!! [OK - 874/16000 bytes]
```

接下来做什么？

输入 **copy** 命令后，系统可能提示用户输入额外信息或让用户确认操作。显示的提示取决于用户在 **copy** 命令中提供了多少信息以及当前的全局配置命令 **file prompt** 设置。

把配置文件从 RCP 服务器拷贝到设备（CLI）

要把配置文件从 RCP 服务器拷贝到运行配置或启动配置中，请完成以下配置任务：

总步骤

1. **enable**
2. **configure terminal**
3. **ip rcmd remote-username** *username*
4. **end**
5. 进行以下配置之一：

- **copy rcp:**[[[**//**[*username***@**]*location*]**/***directory*]**/***filename*]**system:running-config**
- **copy rcp:**[[[**//**[*username***@**]*location*]**/***directory*]**/***filename*]**nvram:startup-config**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	（可选）通过终端进入配置模式。如果覆盖默认远程用户名（见步骤 3），则需要进行此步骤。
步骤 3	**ip rcmd remote-username** *username* **示例：** `Device(config)# ip rcmd remote-username NetAdmin1`	（可选）指定远程用户名。
步骤 4	**end** **示例：** `Device(config)# end`	（可选）退出全局配置模式。如果覆盖了默认远程用户名（见步骤 2），则需要进行此步骤。
步骤 5	进行以下配置之一： • **copy rcp:**[[[**//**[*username***@**]*location*]**/***directory*]**/***filename*]**system:running-config** • **copy rcp:**[[[**//**[*username***@**]*location*]**/***directory*]**/***filename*]**nvram:startup-config** **示例：** `Device# copy rcp://[user1@example.com/dir10/fileone] nvram:startup-config`	把配置文件从 RCP 服务器拷贝到运行配置或启动配置中。

示例

从 RCP 拷贝 Running-Config

以下示例拷贝了 IP 地址为 172.16.101.101 远程服务器上的 netadmin1 目录中的配置文件 host1-confg，在设备上加载并运行文件中的命令：

```
Device# copy rcp://netadmin1@172.16.101.101/host1-confg system:running-config
Configure using host1-confg from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file host1-confg:![OK]
Device#
%SYS-5-CONFIG: Configured from host1-config by rcp from 172.16.101.101
```

从 RCP 拷贝 Startup-Config

以下示例指定使用远程用户名 netadmin1，然后把 IP 地址为 172.16.101.101 远程服务器上的 netadmin1 目录中的配置文件 host2-confg 拷贝到启动配置中。

```
Device# configure terminal
Device(config)# ip rcmd remote-username netadmin1
Device(config)# end
Device# copy rcp: nvram:startup-config
Address of remote host [255.255.255.255]? 172.16.101.101
Name of configuration file[rtr2-confg]? host2-confg
Configure using host2-confg from 172.16.101.101?[confirm]
Connected to 172.16.101.101
Loading 1112 byte file host2-confg:![OK]
[OK]
Device#
%SYS-5-CONFIG_NV:Non-volatile store configured from host2-config by rcp from 172.16.101.101
```

接下来做什么？

输入 **copy** 命令后，系统可能提示用户输入额外信息或让用户确认操作。显示的提示取决于用户在 **copy** 命令中提供了多少信息以及当前的全局配置命令 **file prompt** 设置。

把配置文件从 FTP 服务器拷贝到设备（CLI）

要把配置文件从 FTP 服务器拷贝到运行配置或启动配置中，请完成以下配置任务：

总步骤

1. **enable**
2. **configure terminal**
3. **ip ftp username** *username*
4. **ip ftp password** *password*
5. **end**
6. 进行以下操作之一：
   - **copy ftp:** [[[**//**[*username*[**:***password*]**@**]*location*] **/***directory*]**/***filename*]**system:running-config**
   - **copy ftp:** [[[**//**[*username*[**:***password*]**@**]*location*]**/***directory*]**/***filename*]**nvram:startup-config**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	（可选）允许进入全局配置模式。如果希望覆盖默认的远程用户名或密码（见步骤 3 和步骤 4），则需要进行此步骤。
步骤 3	**ip ftp username** *username* **示例：** `Device(config)# ip ftp username`	（可选）指定默认远程用户名。

	NetAdmin1	
步骤 4	**ip ftp password** *password* **示例：** Device(config)# ip ftp password adminpassword	（可选）指定默认密码。
步骤 5	**end** **示例：** Device(config)# end	（可选）退出全局配置模式。如果覆盖默认远程用户名或密码（见步骤 3 和步骤 4），则需要进行此步骤。
步骤 6	进行以下操作之一： • **copy ftp:** [[[**//**[*username*[**:***password*]**@**]*location*] **/***directory*]**/***filename*]**system:running-config** • **copy ftp:** [[[**//**[*username*[**:***password*]**@**]*location*]**/***directory*]**/***filename*]**nvram:startup-config** **示例：** Device# copy ftp:nvram:startup-config	使用 FTP 从网络服务器上把配置文件拷贝到运行配置或启动配置中。

示例

从 FTP 拷贝 Running-Config

以下示例拷贝了 IP 地址为 172.16.101.101 远程服务器上的 netadmin1 目录中的配置文件 host1-confg，在设备上加载并运行文件中的命令：

```
Device# copy ftp://netadmin1:mypass@172.16.101.101/host1-confg system:running-config
Configure using host1-confg from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file host1-confg:![OK]
Device#
%SYS-5-CONFIG: Configured from host1-config by ftp from 172.16.101.101
```

从 FTP 拷贝到 Startup-Config

以下示例指定使用远程用户名 netadmin1，然后把 IP 地址为 172.16.101.101 远程服务器上的 netadmin1 目录中的配置文件 host2-confg 拷贝到启动配置中。

```
Device# configure terminal
Device(config)# ip ftp username netadmin1
Device(config)# ip ftp password mypass
Device(config)# end
Device# copy ftp: nvram:startup-config
Address of remote host [255.255.255.255]? 172.16.101.101
Name of configuration file[host1-confg]? host2-confg
Configure using host2-confg from 172.16.101.101?[confirm]
Connected to 172.16.101.101
Loading 1112 byte file host2-confg:![OK]
[OK]
Device#
```

```
%SYS-5-CONFIG_NV:Non-volatile store configured from host2-config by ftp from 172.16.101.101
```

接下来做什么？

输入 **copy** 命令后，系统可能提示用户输入额外信息或让用户确认操作。显示的提示取决于用户在 **copy** 命令中提供了多少信息以及当前的全局配置命令 **file prompt** 设置。

维护大于 NVRAM 的配置文件

要维护超过了 NVRAM 大小的配置文件，请执行以下所述的配置任务：

压缩配置文件（CLI）

要压缩配置文件，请完成本节中的配置任务：

总步骤

1. **enable**
2. **configure terminal**
3. **service compress-config**
4. **end**
5. 进行以下操作之一：
   - 使用FTP、RCP或TFTP 拷贝新的配置文件。
   - **configure terminal**
6. **copy system:running-config nvram:startup-config**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 3	**service compress-config** **示例：** `Device(config)# service compress-config`	指定要压缩的配置文件。
步骤 4	**end** **示例：** `Device(config)# end`	退出全局配置模式。
步骤 5	进行以下操作之一： • 使用FTP、RCP或TFTP 拷贝新的配置文件。 • **configure terminal** **示例：** `Device# configure terminal`	输入新配置： • 如果尝试加载比 NVRAM 大三倍以上的配置，系统会显示以下错误消息： “ [buffer overflow - *file-size*

		/*buffer-size* bytes].″
步骤 6	**copy system:running-config nvram:startup-config** **示例：** `Device(config)# copy system:running-confignvram:startup-config`	完成对运行配置的修改后，保存新的配置。

示例

以下示例展示了把一个 129 KB 的配置文件压缩到 11 KB 的过程。

```
Device# configure terminal
Device(config)# service compress-config
Device(config)# end
Device# copy tftp://172.16.2.15/tokyo-confg system:running-config
Configure using tokyo-confg from 172.16.2.155? [confirm] y
Booting tokyo-confg from 172.16.2.155:!!! [OK - 874/16000 bytes]
Device# copy system:running-config nvram:startup-config
Building configuration...
Compressing configuration from 129648 bytes to 11077 bytes
[OK]
```

在 A 类闪存文件系统中把配置存储到闪存（CLI）

要把启动配置保存到闪存中，请完成本节的配置任务：

总步骤

1. **enable**
2. **copy nvram:startup-config** *flash-filesystem:filename*
3. **configure terminal**
4. **boot config flash-filesystem:** *filename*
5. **end**
6. 进行以下操作之一：
   - 使用 FTP、RCP 或 TFTP 拷贝新配置文件。如果尝试加载比 NVRAM 大三倍以上的配置，系统会显示错误消息：“[buffer overflow - *file-size* /*buffer-size* bytes].″
   - **configure terminal**
7. **copy system:running-config nvram:startup-config**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**copy nvram:startup-config** *flash-filesystem:filename* **示例：**	把当前启动配置拷贝到新位置。

	Device# copy nvram:startup-config usbflash0:switch-config	
步骤 3	**configure terminal** **示例：** Device# configure terminal	进入全局配置模式。
步骤 4	**boot config flash-filesystem:** *filename* **示例：** Device(config)# boot config usbflash0:switch-config	设置环境变量 CONFIG_FILE，把启动配置文件保存到闪存中。
步骤 5	**end** **示例：** Device(config)# end	退出全局配置模式。
步骤 6	进行以下操作之一： • 使用 FTP、RCP 或 TFTP 拷贝新的配置文件。如果尝试加载比 NVRAM 大三倍以上的配置，系统会显示以下错误消息："[buffer overflow - *file-size* /*buffer-size* bytes]." • **configure terminal** **示例：** Device# configure terminal	输入新配置。
步骤 7	**copy system:running-config nvram:startup-config** **示例：** Device(config)# copy system:running-confignvram:startup-config	完成对运行配置的修改后，保存新的配置。

示例

以下示例把配置文件保存到 usbflash0 中：

```
Device# copy nvram:startup-config usbflash0:switch-config
Device# configure terminal
Device(config)# boot config usbflash0:switch-config
Device(config)# end
Device# copy system:running-config nvram:startup-config
```

通过网络加载配置命令（CLI）

要使用网络服务器存储大型配置，请完成本节中的配置任务：

总步骤

1. enable

2. copy system:running-config {**ftp:** | **rcp:** | **tftp:**}

3. configure terminal

4. boot network {**ftp:**[[[**//**[*username* [**:***password*]**@**]*location*]**/***directory*]**/***filename*] |

rcp:[[[**//**[*username***@**]*location*]**/***directory*]**/***filename*] | **tftp:**[[[**//***location*]**/***directory*]**/***filename*]}

5. service config

6. end

7. copy system:running-config nvram:startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**copy system:running-config** {**ftp:** | **rcp:** | **tftp:**} **示例：** `Device# copy system:running-config ftp:`	把运行配置保存到 FTP、RCP 或 TFTP 服务器上。
步骤 3	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 4	**boot network** {**ftp:**[[[**//**[*username* [**:***password*]**@**]*location*]**/***directory*]**/***filename*] | **rcp:**[[[**//**[*username***@**]*location*]**/***directory*]**/***filename*] | **tftp:**[[[**//***location*]**/***directory*]**/***filename*]} **示例：** `Device(config)# boot network ftp://user1:guessme@example.com/dir10/file1`	指定启动时要通过网络服务器加载的启动配置文件。
步骤 5	**service config** **示例：** `Device(config)# service config`	设备交换机在系统启动时下载配置文件。
步骤 6	**end** **示例：** `Device(config)# end`	退出全局配置模式。
步骤 7	**copy system:running-config nvram:startup-config** **示例：** `Device(config)# copy system:running-confignvram:startup-config`	保存配置。

把闪存中的配置文件拷贝为启动配置或运行配置（CLI）

要把配置文件从闪存直接拷贝到 NVRAM 中的启动配置或拷贝为运行配置，请执行步骤 2 中的命令：

总步骤

1. enable

2. 进行以下操作之一：

- **copy** *filesystem*: [*partition-number***:**][*filename*] **nvram:startup-config**

• **copy** *filesystem*: [*partition-number*:][*filename*] **system:running-config**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	进行以下操作之一： • **copy** *filesystem*: [*partition-number*:][*filename*] **nvram:startup-config** • **copy** *filesystem*: [*partition-number*:][*filename*] **system:running-config** **示例：** `Device# copy usbflash0:4:INOS-upgrade-1 nvram:startup-config`	• 把配置文件直接加载到 NVRAM 中 或 • 把配置文件拷贝为运行配置

示例

以下示例把名为 INOS-upgrade-1 的配置文件从闪存 PC 卡 usbflash0 分区 4 中拷贝到设备的启动配置：

```
Device# copy usbflash0:4:INOS-upgrade-1 nvram:startup-config
Copy 'INOS-upgrade-1' from flash device as 'startup-config' ? [yes/no] yes
[OK]
```

在闪存文件系统之间拷贝配置文件（CLI）

在有多个闪存文件系统的平台上，可以把一个闪存文件系统中的文件拷贝到另一个闪存中，比如从内部闪存拷贝到其他的闪存文件系统。把文件拷贝到不同的闪存文件系统让用户可以创建运行配置的备份，并对其他设备复用配置。要在闪存文件系统之间拷贝配置文件，请在 EXEC 模式中使用以下命令：

总步骤

1. enable

2. show *source-filesystem*:

3. copy *source-filesystem*: [*partition-number*:][*filename*] *dest-filesystem*:[*partition-number*:][*filename*]

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**show** *source-filesystem*: **示例：**	显示闪存的配置及内容，验证文件名。

	`Device# show flash:`	
步骤 3	**copy** *source-filesystem*:[*partition-number*:][*filename*] *dest-filesystem*:[*partition-number*:][*filename*] **示例：** `Device# copy flash: usbflash0:`	在闪存设备之间拷贝配置文件。 源设备与目的设备不能相同。比如，命令 **copy usbflash0: usbflash0:**是不合法的。

示例

以下示例把名为 running-config 的文件从内部闪存的分区 1 拷贝到设备 usbflash0 上的分区 1 中。此例中源分区未指定，所以设备提示确认分区编号：

```
Device# copy flash: usbflash0:
System flash
Partition Size Used Free Bank-Size State Copy Mode
1 4096K 3070K 1025K 4096K Read/Write Direct
2 16384K 1671K 14712K 8192K Read/Write Direct
[Type ?<no> for partition directory; ? for full directory; q to abort]
Which partition? [default = 1]
System flash directory, partition 1:
File Length Name/status
1 3142748 dirt/network/mars-test/c3600-j-mz.latest
2 850 running-config
[3143728 bytes used, 1050576 available, 4194304 total]
usbflash0 flash directory:
File Length Name/status
1 1711088 dirt/gate/c3600-i-mz
2 850 running-config
[1712068 bytes used, 2482236 available, 4194304 total]
Source file name? running-config
Destination file name [running-config]?
Verifying checksum for 'running-config' (file # 2)... OK
Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]
Copy 'running-config' from flash: device
as 'running-config' into usbflash0: device WITH erase? [yes/no] yes
Erasing device...
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased!
[OK - 850/4194304 bytes]
Flash device copy took 00:00:30 [hh:mm:ss]
Verifying checksum... OK (0x16)
```

把配置文件从 FTP 服务器拷贝到闪存设备（CLI）

要把配置文件从 FTP 服务器拷贝到闪存设备中，请完成本节中的配置任务：

总步骤

1. **enable**
2. **configure terminal**
3. **ip ftp username** *username*
4. **ip ftp password** *password*
5. **end**
6. **copy ftp:** [[**//***location*]**/***directory*]**/***bundle_name* **flash:**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	（可选）进入全局配置模式。如果覆盖默认的远程用户名或密码（见步骤 3 和步骤 4），则需要进行此步骤。
步骤 3	**ip ftp username** *username* **示例：** `Device(config)# ip ftp username Admin01`	（可选）指定远程用户名。
步骤 4	**ip ftp password** *password* **示例：** `Device(config)# ip ftp password adminpassword`	（可选）指定远程密码。
步骤 5	**end** **示例：** `Device(config)# end`	（可选）退出配置模式。如果覆盖默认的远程用户名或密码（见步骤 3 和步骤 4），则需要进行此步骤。
步骤 6	**copy ftp:** [[**//***location*]**/***directory*]**/***bundle_name* **flash:** **示例：** `Device>copyftp:/cat3k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.binflash:`	使用 FTP 把配置文件从网络服务器拷贝到闪存设备中。

接下来做什么？

输入 **copy** 命令后，系统可能提示用户输入额外信息或让用户确认操作。显示的提示取决于用户在 **copy** 命令中提供了多少信息以及当前的全局配置命令 **file prompt** 设置。

把配置文件从 RCP 服务器拷贝到闪存设备（CLI）

要把配置文件从 RCP 服务器拷贝到闪存设备，请完成本节的配置任务：

总步骤

1. enable

2. configure terminal

3. ip rcmd remote-username *username*

4. end

5. copy rcp: [[[**//**[*username***@**]*location*]**/***directory*] **/***bundle_name*] **flash:**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	（可选）进入全局配置模式。如果覆盖默认的远程用户名（见步骤 3），则需要进行此步骤。
步骤 3	**ip rcmd remote-username** *username* **示例：** `Device(config)# ip rcmd remote-username Admin01`	（可选）指定远程用户名。
步骤 4	**end** **示例：** `Device(config)# end`	（可选）退出配置模式。如果覆盖默认的远程用户名（见步骤 3），则需要进行此步骤。
步骤 5	**copy rcp:** [[[**//**[*username***@**]*location*]**/***directory*] **/***bundle_name*] **flash:** **示例：** `Device# copyrcp://netadmin@172.16.101.101/bundle1 flash:`	使用 RCP 把配置文件从网络服务器拷贝到闪存设备上。请响应或确认设备对其他信息的请求提示。显示的提示取决于用户在 **copy** 命令中提供了多少信息以及当前的全局配置命令 **file prompt** 设置。

把配置文件从 TFTP 服务器拷贝到闪存设备（CLI）

要把配置文件从 TFTP 服务器拷贝到闪存设备，请完成本节的配置任务：

总步骤

1. enable

2. copy tftp: [[[**//***location*]**/***directory*]**/**bundle_name **flash:**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**copy tftp:** [[[**//***location*]**/***directory*]**/**bundle_name **flash:**	把配置文件从 TFTP 网络服务器拷贝到闪存设备上。请响应或确认设备对

	示例： `Device#copy tftp:/cat3k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin flash:`	其他信息的请求提示。显示的提示取决于用户在 **copy** 命令中提供了多少信息以及当前的全局配置命令 **file prompt** 设置。

示例

以下示例展示了把名为 switch-config 的配置文件从 TFTP 服务器拷贝到 usbflash0 的闪存卡的过程。拷贝的文件被重命名为 new-config。

```
Device#
copy tftp:switch-config usbflash0:new-config
```

重新执行启动配置文件中的配置命令（CLI）

要重新执行启动配置文件中的命令，请完成本节的配置任务：

总步骤

1. enable

2. configure memory

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure memory** **示例：** `Device# configure memory`	重新执行配置文件中的配置命令。

清除启动配置（CLI）

可以清除启动配置中的配置信息。如果重启设备时没有启动配置，设备会进入 Setup 命令系统，用户可以从头开始配置设备。要清除启动配置的内容，请完成本节的配置任务：

总步骤

1. enable

2. erase nvram

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**erase nvram**	清除启动配置的内容。

	示例： `Device# erase nvram`	**注释：**对除了A类闪存文件系统平台以外的所有平台来说，此命令会擦除NVRAM。在A类闪存文件系统平台上，使用EXEC命令**erasestartup-config**时，设备会删除环境变量CONFIG_FILE指向的配置。如果变量指向NVRAM，则设备会擦除NVRAM。如果环境变量CONFIG_FILE指向了一个闪存设备及文件名，设备会删除该配置文件。即设备会把文件标记为“已删除”，但不会擦除文件。此特性允许用户恢复被删除的文件。

删除指定的配置文件（CLI）

要删除指定闪存设备上的指定配置，请完成本节的配置任务：

总步骤

1. enable

2. delete *flash-filesystem*:*filename*

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**delete** *flash-filesystem*:*filename* **示例：** `Device# deleteusbflash0:myconfig`	删除指定闪存设备上的指定配置文件。 注释：在 A 类以及 B 类闪存文件系统中，删除闪存的指定文件时系统会把文件标记为已删除。这允许用户在以后使用 **undelete** EXEC 命令恢复已删除的文件。已擦除的文件不能被恢复。要永久擦除配置文件，需使用 **squeeze** EXEC 命令。在 C 类闪存文件系统中，不能恢复已被删除的文件。如果尝试擦除或删除环境变量 CONFIG_FILE 指定的配置文件，系统会提示用户确认删除操作。

指定 A 类闪存文件系统的 CONFIG_FILE 环境变量（CLI）

在 A 类闪存文件系统中，可以配置 Inspur INOS 软件加载由 CONFIG_FILE 环境变量指定的启动配置文件。CONFIG_FILE 变量默认指向 NVRAM。要更改 CONFIG_FILE 环境变，请完成本节的配置任务：

总步骤

1. enable

2. copy [*flash-url* | *ftp-url* | *rcp-url* | *tftp-url* | **system:running-config** | **nvram:startup-config**] *dest-flash-url*

3. configure terminal

4. boot config *dest-flash-url*

5. end

6. copy system:running-config nvram:startup-config

7. show boot

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**copy** [*flash-url* | *ftp-url* | *rcp-url* | *tftp-url* | **system:running-config** | **nvram:startup-config**] *dest-flash-url* **示例：** `Device# copy system:running-config nvram:startup-config`	把配置文件从设备加载重启使用的位置拷贝到闪存文件系统中。
步骤 3	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 4	**boot config** *dest-flash-url* **示例：** `Device(config)# boot config 172.16.1.1`	设置 CONFIG_FILE 环境变量。此步骤会修改运行时的 CONFIG_FILE 变量值。
步骤 5	**end** **示例：** `Device(config)# end`	退出全局配置模式。
步骤 6	**copy system:running-config nvram:startup-config** **示例：** `Device# copy system:running-config nvram:startup-config`	把步骤 3 中执行的配置保存到启动配置中。
步骤 7	**show boot** **示例：** `Device# show boot`	（可选）验证 CONFIG_FILE 环境变量的内容。

示例

以下示例把运行配置文件拷贝到设备中。此配置在随后系统重启时会被用作启动配置：

```
Device# copy system:running-config usbflash0:config2
Device# configure terminal
Device(config)# boot config usbflash0:config2
Device(config)# end
```

```
Device# copy system:running-config nvram:startup-config
[ok]
Device# show boot
BOOT variable = usbflash0:rsp-boot-m
CONFIG_FILE variable = nvram:
Current CONFIG_FILE variable = usbflash0:config2
Configuration register is 0x010F
```

接下来做什么？

指定了启动配置文件的位置后，**nvram:startup-config** 命令会等同于启动配置文件的新位置。**more nvram:startup-config** EXEC 命令会显示启动配置文件，无论其位置如何。**erase nvram:startup-config** EXEC 命令会擦除 NVRAM 的内容，并删除 CONFIG_FILE 环境变量指向的文件。

使用 **copy system:running-config nvram:startup-config** 命令保存配置时，设备会把配置文件的完整版本保存到 CONFIG_FILE 环境变量指定的位置，并把一份提取版本保存到 NVRAM。提取的版本不包含访问列表信息。如果 NVRAM 中包含完整的配置文件，设备会提示用户确认使用提取版本覆盖完整版本。如果 NVRAM 中包含提取配置，设备不会提示用户确认操作，而是覆盖 NVRAM 中的现有提取配置文件。

注释： 如果把闪存设备中的一个文件指定为 CONFIG_FILE 环境变量的值，每次使用 **copy system:running-config nvram:startup-config** 命令保存配置文件时，旧的配置文件都会被标记为“已删除”，且新的配置文件会被保存到该设备。最终，闪存会被填满，因为旧的配置文件仍然占用空间。使用 **squeeze** EXEC 命令永久删除旧配置文件并回收空间。

配置设备下载配置文件

可以指定一个网络配置文件名以及主机配置文件名的有序列表。Inspur INOS 软件会扫描此列表，直到加载了恰当的网络或主机配置文件。

要配置设备在系统启动时下载配置文件，请执行以下所述任务中的至少一个：

- 配置设备下载网络配置文件（CLI）
- 配置设备下载主机配置文件（CLI）

如果设备启动时不能加载配置文件，它会每 10 分钟尝试一次（默认设置），直到主机提供了请求的文件。每次请求失败时，设备会在控制台终端上显示以下消息：

```
Booting host-confg... [timed out]
```

如果启动配置文件存在问题，或配置寄存器被设置为忽略 NVRAM，设备会进入 Setup 命令系统。

配置设备下载网络配置文件（CLI）

要配置 Inspur INOS 软件在启动时从服务器上下载网络配置文件，请完成本节中的配置任务：

总步骤

1. enable

2. configure terminal

3. boot network {**ftp:**[[[**//**[*username* [**:***password*]**@**]*location*]**/***directory*]**/***filename*] | **rcp:**[[[**//**[*username***@**]*location*]**/***directory*]**/***filename*] | **tftp:**[[[**//***location*]**/***directory*]**/***filename*]}

4. service config

5. end

6. copy system:running-config nvram:startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 3	**boot network** {**ftp:**[[[**//**[*username* [:*password*]**@**]*location*]**/***directory*]**/***filename*] |**rcp:**[[[**//**[*username***@**]*location*]**/***directory*]**/***filename*] | **tftp:**[[[**//***location*]**/***directory*]**/***filename*]} **示例：** `Device(config)# boot network tftp:hostfile1`	指定启动时要下载网络配置文件以及要使用的协议（TFTP、RCP 或 FTP）。 • 如果不指定网络配置文件名，Inspur INOS 软件会使用默认的文件名 network-confg。如果省略了地址，设备会使用广播地址。 • 可以指定多个网络配置文件。软件会按照输入顺序尝试加载，直到成功加载了一个文件。此过程可以用来保存加载到网络服务器上的配置信息不同的文件。
步骤 4	**service config** **示例：** `Device(config)# service config`	让系统在重启时自动加载网络文件。
步骤 5	**end** **示例：** `Device(config)# end`	退出全局配置模式。
步骤 6	**copy system:running-config nvram:startup-config** **示例：** `Device# copy system:running-config nvram:startup-config`	把运行配置保存到启动配置中。

配置设备下载主机配置文件（CLI）

要配置 Inspur INOS 软件在启动时从服务器上下载主机配置文件，请完成本节中的配置任务：

总步骤

1. enable

2. configure terminal

3. boot host {**ftp:**[[[**//**[*username* [**:***password*]**@**]*location*]**/***directory*]**/***filename*] | **rcp:**[[[**//**[*username***@**]*location*]**/***directory*]**/***filename*] | **tftp:**[[[**//***location*]**/***directory*]**/***filename*] }

4. service config

5. end

6. copy system:running-config nvram:startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device> enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 3	**boot host** {**ftp:**[[[**//**[*username* [:*password*]**@**]*location*]**/***directory*]**/***filename*] |**rcp:**[[[**//**[*username***@**]*location*]**/***directory*]**/***filename*] | **tftp:**[[[**//***location*]**/***directory*]**/***filename*] } **示例：** `Device(config)# boot host tftp:hostfile1`	指定启动时要下载主机配置文件以及要使用的协议（TFTP、RCP 或 FTP）。 • 如果不指定网络配置文件名，设备会使用自己的名称来构建主机配置文件名，把名称变为小写，移除所有域信息，并附加“-confg”。如果没有可用的主机名信息，系统会使用默认的主机配置文件名 device-confg。如果省略了地址，设备会使用广播地址。 • 可以指定多个主机配置文件。Inspur INOS 软件会按照输入顺序尝试加载，直到成功加载了一个文件。此过程可以用来保存加载到网络服务器上的配置信息不同的文件。
步骤 4	**service config** **示例：** `Device(config)# service config`	让系统在重启时自动加载主机文件。
步骤 5	**end** **示例：** `Device(config)# end`	退出全局配置模式。
步骤 6	**copy system:running-config nvram:startup-config** **示例：** `Device# copy system:running-config nvram:startup-config`	把运行配置保存到启动配置中。

示例

以下示例配置设备下载名为 hostfile1 的主机配置文件以及名为 networkfile1 的网络配置文件。设备会使用 TFTP 以及广播地址来获取文件：

```
Device# configure terminal
```

```
Device(config)# boot host tftp:hostfile1
Device(config)# boot network tftp:networkfile1
Device(config)# service config
Device(config)# end
Device# copy system:running-config nvram:startup-config
```

其他参考资料

相关文档

相关主题	文档标题
Inspur INOS 命令	Inspur INOS 主命令列表，所有版本
Inspur INOS 配置命令	*Inspur INOS 配置基础命令参考*

错误信息解释

描述	链接
为帮助管理员搜索并解决该版本中的系统错误信息，管理员可使用错误信息解释工具。	http://www.icntnetworks.com

标准

标准	标题
不支持新标准或修订的标准，且支持的现有标准未被修改	-

RFC

RFC	标题
不支持新 RFC 或修订的 RFC，且支持的现有 RFC 未被修改	-

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	http://www.icntnetworks.com

配置替换与配置回滚

配置替换与配置回滚的前提

用作配置替换与配置回滚特性的输入配置文件格式必须符合以下标准 Inspur 软件配置文件缩进规则：

- 在新行中开始所有命令，除非命令在配置子模式中，否则无缩进。
- 第一级配置子模式中的命令使用一个空格缩进。
- 第二级配置子模式中的命令使用两个空格缩进。
- 后续子模式中的命令按此规则缩进。

这些缩进规则描述了软件是如何为**showrunning-config**或**copy running-config** *destination-url*这样的命令创建配置文件的。Inspur设备上生成的任意配置文件都符合这些规则。
需要有比当前运行配置与保存的替换配置文件大小之和更大的空闲内存空间。

配置替换与配置回滚的限制

如果设备的空闲空间小于两配置文件（当前运行配置与保存的替换配置）的大小之和，配置替换操作不会进行。
特定的Inspur配置命令不能添加到运行配置或从中移除，如网络设备物理组件（如物理接口）附加的命令。如果Ethernet 0接口物理存在于设备上，配置替换操作不能从当前的运行配置中移除**interface ethernet 0**命令。类似的，如果Ethernet 1接口不存在于设备上，**interfaceethernet 1**命令不能被添加到运行配置中。尝试执行这些类型更改的配置替换操作会产生错误消息，表示这些特定的命令行替换失败。
在极少数情况下，如果不重载设备，特定的Inspur配置命令不能从运行配置中移除。尝试移除此类命令的配置替换操作会产生错误消息，表示这些特定的命令行替换失败。

关于配置替换与配置回滚的信息

配置存档

Inspur INOS配置存档功能旨在提供一种存储、组织及管理Inspur INOS配置文件存档的机制，增强 **configure replace** 命令提供的配置回滚能力。在引入此特性之前，用户可以使用命令 **copy running-config** *destination-url* 保存运行配置的副本，本地或远程存储替换文件。然而，这种方式缺少自动化的文件管理能力。配置替换与配置回滚特性有能力自动把运行配置的副本保存为 Inspur INOS 配置存档。这些存档文件作为检查点配置参考，可以被 **configure replace** 命令用来回退到之前的配置状态。
archive config 命令允许用户把 Inspur INOS 配置保存到配置存档中。存档配置会使用标准位置以及文件名前缀保存，并为连续的文件附加上递增版本号（以及可选的时间戳）。此功能提供了一种对保存的 Inspur INOS 配置文件一致性标识的方式。可以指定在存档中保存运行

配置的多少个版本。存档中保存了最大数量的文件之后，保存下一个新文件时最旧的文件会被自动删除。**show archive** 命令会显示保存在 Inspur INOS 配置存档中的所有配置文件信息。Inspur INOS 配置存档中保存着配置文件，并可为 **configure replace** 命令使用，存档可位于以下文件系统中：FTP、HTTP、RCP 以及 TFTP。

配置替换

特权 EXEC 命令 **configure replace** 提供了使用保存的 Inspur INOS 配置文件替换当前运行配置的能力。此功能可以用来回退到之前的配置状态，能有效地回滚自之前的配置状态保存以来进行的所有配置更改。

使用命令时，必须指定一个保存的Inspur INOS配置作为当前运行配置的替换配置文件。替换文件必须是Inspur INOS设备生成的完整配置（如**copyrunning-config** *destination-url*命令生成的配置），如果替换文件是外部生成的，其必须符合Inspur INOS设备生成文件的格式。输入**configure replace**命令时，当前运行配置会与指定的替换配置进行比较，并会产生一组diffs（文件差异）。用于比较两个文件的算法与**show archiveconfig differences**命令使用的算法相同。最终的diffs会被Inspur INOS解析器使用，以应用替换配置的状态。此过程中只有diffs会被应用，进而避免了因重新应用当前运行配置中已存在的配置命令而导致的潜在的服务中断可能。此算法能通过多遍处理过程有效地解决对于顺序相关命令（如访问列表）的配置更改。在正常情况下，三遍以下的处理就能完成配置替换操作；限制最多执行五遍，排除任何循环行为。

Inspur INOS特权EXEC命令**copy** *source-url* **running-config**常被用来把保存的Inspur INOS配置文件拷贝到运行配置中。使用**copy** *source-url* **running-config**命令替代特权EXEC命令**configure replace** *target-url*时，应注意以下几点主要的区别：

- **copy** *source-url* **running-config**命令是合并操作，会保留源文件以及当前运行配置中的所有命令。此命令不会从当前运行配置中移除源文件中没有的命令。相比之下，**configure replace** *target-url*命令会从当前运行配置中移除源文件中没有的命令，并把需要添加的命令添加到当前运行配置中。
- **copy** *source-url* **running-config**命令会应用源文件中的每一条命令，无论该命令是否已经存在于当前的运行配置中。此算法是低效的，且有时会导致服务中断。相比之下，**configure replace** *target-url*只应用需要被应用的命令——当前运行配置中已有命令不会被重新应用。
- 部分的配置文件可以用作 **copy** *source-url* **running-config** 命令的源文件，而 **configure replace** *target-url* 命令的替换文件必须使用完整的 Inspur INOS 配置文件。

配置替换操作引入了锁特性。使用 **configure replace** 命令时，运行配置文件在配置替换操作过程中会被锁住。锁机制避免了其他用户在替换操作进行时更改运行配置，这样的更改可能导致替换操作不成功终止。可以在输入 **configure replace** 命令时使用 **no lock** 关键字来禁用运行配置锁。

运行配置锁在配置替换操作结束时会被自动清除。可以使用**show configurationlock**命令显示当前可能应用到运行配置上的锁。

配置回滚

回滚的概念来源于数据库操作中常用的事务处理模型。在数据库事务中，用户可能对一个数据库表进行一组更改操作。之后用户必须选择是提交更改（永久应用更改）还是回滚更改（丢弃更改并退回到表的之前状态）。在这种情况下，回滚意味着包含更改记录的日志文件被丢

弃，且更改不被应用。回滚操作的结果是在更改应用之前回退到之前的状态。

configure replace 命令允许用户回退到之前的配置状态，高效地回滚自之前配置状态保存以来进行的配置更改。Inspur INOS 配置回滚功能不会回滚已应用的一组特定更改，而是使用回退到特定配置状态的概念，基于已保存的 Inspur INOS 配置文件进行回滚操作。此概念类似于数据库中通过保存检查点（一个保存的数据库版本）来保留特定状态的理念。

如果希望使用配置回滚功能，在执行任何配置更改前必须保存 Inspur INOS 的运行配置。之后用户可以输入配置更改，并使用保存的配置文件回滚更改（使用 **configure replace** *target-url* 命令）。此外，如同一些回滚模型一样，因为可以指定使用任何保存的 Inspur INOS 配置文件作为替换配置，用户只能被限制执行固定数量的回滚操作。

配置回滚已确认的更改

配置回滚已确认的更改特性允许在进行配置更改时可以对更改进行确认。如果没有收到确认，配置会返回应用更改前的状态。此特性能防护因配置更改而无意中导致的网络设备以及用户或管理应用的连接性丢失。

配置替换与配置回滚的益处

- 允许用户回退到之前的配置状态，高效地回滚配置更改。
- 允许用户使用在不进行设备重启且无需手动撤销 CLI 对运行配置更改的情况下，使用启动配置文件替换当前运行配置文件，进而检查系统停机时间。
- 允许用户回退到任何保存的 Inspur INOS 配置状态。
- 简化配置更改操作，允许用户给设备应用完整的配置文件，且仅有需要被添加或移除的命令会受影响。
- 使用 **configure replace** 命令替代 **copy** *source-url* **running-config** 命令，能提高效率并避免重新应用已存在命令而产生的服务中断风险。

如何使用配置替换与配置回滚

创建配置存档（CLI）

使用 **configure replace** 命令无需先决配置。可以把 **configure replace** 命令与 Inspur 配置存档的 **archive config** 命令组合使用，这能为配置回滚提供显著的好处。使用 **archive config** 命令之前，必须设置配置存档。执行此任务来进行配置存档特性设置。

总步骤

1. enable

2. configure terminal

3. archive

4. path *url*

5. maximum *number*

6. time-period *minutes*

7. end

8. archive config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 3	**archive** **示例：** `Device(config)# archive`	进入存档配置模式。
步骤 4	**path** *url* **示例：** `Device(config-archive)# pathflash:myconfiguration`	指定 Inspur INOS 配置存档的位置以及文件名前缀。 **注释：**如果 **path** 中指定一个目录，目录名必须后跟一个正斜线，如 **path** flash:/directory/。文件名后无需正斜线。
步骤 5	**maximum** *number* **示例：** `Device(config-archive)# maximum14`	（可选）设置要保存在 Inspur INOS 配置存档中的运行配置存档的最大数量。 • *number* 参数是要保存在 Inspur INOS 配置存档中的运行配置存档的最大数量。合法值范围从 1 到 14，默认值是 10。 **注释：**使用此命令前，必须配置 **path** 命令，指定 Inspur INOS 配置存档的位置以及文件名前缀。
步骤 6	**time-period** *minutes* **示例：** `Device(config-archive)#time-period 1440`	（可选）设置在 Inspur INOS 配置存档中自动保存当前运行配置存档文件的时间增量。 • *minutes* 参数以分钟为单位指定在 Inspur INOS 配置存档中自动保存当前运行配置存档文件的频率。 **注释：**使用此命令前，必须配置 **path** 命令，指定 Inspur INOS 配置存档的位置以及文件名前缀。
步骤 7	**end** **示例：** `Device(config-archive)# end`	返回特权 EXEC 模式。
步骤 8	**archive config** **示例：** `Device# archive config`	把当前运行配置文件保存到配置存档中。 **注释：**使用此命令前必须配置 **path** 命令。

执行配置替换或配置回滚操作（CLI）

执行此任务，使用保存的 Inspur INOS 配置文件替换当前的运行配置文件。

注释： 执行此过程之前必须创建配置存档，详细步骤见*创建配置存档（CLI）*。以下过程详述了在当前运行配置存在问题时如何返回存档的配置。

总步骤

1. enable

2. configure replace *target-url* [**nolock**] [**list**] [**force**] [**ignore case**] [**revert trigger** [**error**][**timer** *minutes*]| **time** *minutes*]]

3. configure revert { **now** | **timer** {*minutes* | **idle** *minutes*} }

4. configure confirm

5. exit

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>enable`	启用特权 EXEC 模式，在提示时输入密码。
步骤 2	**configure replace** *target-url*[**nolock**] [**list**] [**force**] [**ignorecase**] [**revert trigger** [**error**][**timer** *minutes*]	**time** *minutes*]] **示例：** `Device# configure replace flash: startup-config time120`	使用保存的 Inspur INOS 配置文件替换当前的运行配置文件。 • *target - url*参数是要替换当前运行配置的保存的Inspur INOS配置文件的URL（可被Inspur INOS文件系统访问）。这些配置文件可以使用**archiveconfig**命令创建。 • **list**关键字列出Inspur INOS软件解析器在每遍配置替换操作中使用的命令。执行的总解析次数也会被显示。 • **force**关键字会使用指定的保存Inspur INOS配置文件替换当前运行配置文件，且不提示用户进行确认。 • **time** *minutes*关键字及参数指定确认时间（单位为分钟），在此时间内用户必须输入**configure confirm**命令确认替换当前的运行配置文件。如果在指定的时间限制之内没有输入**configure confirm**命令，配置替换操作会被自动撤销（换句话说，当前的运行配置文件会被恢复为输入

		configurereplace命令之前的配置状态）。 • **nolock**关键字会禁用运行配置文件锁。运行配置文件锁会防止配置替换操作期间用户更改运行配置。 • **revert trigger** 关键字会设置在以下事件触发时回退到原始配置： ◦ **error**——在错误时回退到原始配置。 ◦ **timer** *minutes*——如果超过了指定的时间，回退到原始配置。 • **ignore case** 关键字允许配置忽略确认命令的大小写。
步骤 3	**configure revert** { **now** | **timer**{*minutes* | **idle** *minutes*} } **示例：** `Device# configure revert now`	（可选）在特权 EXEC 模式中使用 **configure revert** 命令，取消计时的回退并立即触发回退操作，或重置计时回退参数。 • **now**——立即触发回退。 • **timer**——重置回退计时器设置。 • 使用 **timer** 关键字以及 *minutes* 参数指定新的回退时间。 • 使用 **idle** 关键字以及分钟数来设置回退到保存配置之前允许的最大无活动时间。
步骤 4	**configure confirm** **Example:** `Device# configure confirm`	（可选）确认使用保存的 Inspur INOS 配置文件替换当前的运行配置文件。 **注释：**仅在指定了 **configure replace** 命令的 **time** *seconds* 关键字及参数时使用此命令。
步骤 5	**exit** **示例：** `Device# exit`	返回用户 EXEC 模式。

特性监控及故障排除（CLI）

要对配置替换与配置回滚特性进行监控与故障排除，请执行此任务。

总步骤

1. enable

2. show archive

3. debug archive versioning

4. debug archive config timestamp

5. exit

具体步骤

步骤1　enable

使用此命令启用特权 EXEC 模式，在提示时输入密码。

示例：

```
Device>enable
Device#
```

步骤 2　show archive

使用此命令显示保存在 Inspur INOS 配置存档中的文件信息。

示例：

```
Device# show archive
There are currently 1 archive configurations saved.
The next archive file will be named flash:myconfiguration-2
Archive #    Name
0
1            flash:myconfiguration-1 <- Most Recent
2
3
4
5
6
7
8
9
10
11
12
13
14
```

以下是运行配置的几个存档文件被保存之后的 **show archive** 命令示例输出。此例中，存档文件的最大数量被设置为 3。

示例：

```
Device# show archive
There are currently 3 archive configurations saved.
The next archive file will be named flash:myconfiguration-8
Archive #    Name
0
1            :Deleted
2            :Deleted
3            :Deleted
4            :Deleted
5            flash:myconfiguration-5
6            flash:myconfiguration-6
```

```
7            flash:myconfiguration-7 <- Most Recent
8
9
10
11
12
13
14
```

步骤 3　debug archive versioning

使用此命令启用 Inspur INOS 配置存档活动的调试，帮助进行配置替换与回滚的监控及故障排除。

示例：

```
Device# debug archive versioning
Jan 9 06:46:28.419:backup_running_config
Jan 9 06:46:28.419:Current = 7
Jan 9 06:46:28.443:Writing backup file flash:myconfiguration-7
Jan 9 06:46:29.547: backup worked
```

步骤 4　debug archive config timestamp

使用此命令显示每一步配置替换操作的处理时间以及处理的配置文件的大小的调试信息。

示例：

```
Device# debug archive config timestamp
Device# configure replace flash:myconfiguration force
Timing Debug Statistics for INOS Config Replace operation:
    Time to read file usbflash0:sample_2.cfg = 0 msec (0 sec)
    Number of lines read:55
    Size of file 1054
Starting Pass 1
    Time to read file system:running-config = 0 msec (0 sec)
    Number of lines read:93
    Size of file :2539
    Time taken for positive rollback pass = 320 msec (0 sec)
    Time taken for negative rollback pass = 0 msec (0 sec)
    Time taken for negative incremental diffs pass = 59 msec (0 sec)
    Time taken by PI to apply changes = 0 msec (0 sec)
    Time taken for Pass 1 = 380 msec (0 sec)
Starting Pass 2
    Time to read file system:running-config = 0 msec (0 sec)
    Number of lines read:55
    Size of file 1054
    Time taken for positive rollback pass = 0 msec (0 sec)
    Time taken for negative rollback pass = 0 msec (0 sec)
    Time taken for Pass 2 = 0 msec (0 sec)
Total number of passes:1
```

```
Rollback Done
```

步骤 5 **exit**

使用此命令返回用户 EXEC 模式。

示例：

```
Device# exit
Device>
```

配置替换与配置回滚的配置示例

创建配置存档

以下示例展示了如何进行 Inspur INOS 配置存档的初始配置。此例中，指定 flash:myconfiguration 作为配置存档的位置以及文件名前缀，设置保存的最大存档文件数量为 10。

```
configure terminal
!
archive
 path flash:myconfiguration
 maximum 10
end
```

使用保存的 Inspur INOS 配置文件替换当前的运行配置

以下示例展示了如何使用名为 flash:myconfiguration 的保存 Inspur INOS 配置文件替换当前的运行配置。**configure replace** 命令会交互式地提示用户确认操作。

```
Device# configure replace flash:myconfiguration
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y
Total number of passes: 1
Rollback Done
```

在以下示例中，指定的 **list** 关键字会显示在配置替换操作期间要应用的命令行：

```
Device# configure replace flash:myconfiguration list
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y
!Pass 1
!List of Commands:
no snmp-server community public ro
snmp-server community mystring ro
```

```
end
Total number of passes: 1
Rollback Done
```

回退到启动配置文件

以下示例展示了如何使用 **configure replace** 命令回退到 Inspur INOS 启动配置文件。此例也使用 **force** 关键字覆盖交换式的用户提示：

```
Device# configure replace flash:startup-config force
Total number of passes: 1
Rollback Done
```

执行带有 configure confirm 命令的配置替换操作

以下示例展示使用 **configure replace** 命令以及 **time** *minutes* 关键字。必须在指定的时间内输入 **configure confirm** 命令，确认替换当前的运行配置文件。如果在指定的时间限制内没有输入 **configure confirm** 命令，配置替换操作会自动撤销（换句话说，当前的运行配置文件会被恢复为输入 **configurereplace** 命令之前的配置状态）。

```
Device# configure replace flash:startup-config time 120
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y
Total number of passes: 1
Rollback Done
Device# configure confirm
```

以下示例展示了使用 **configure revert** 命令以及 **timer** 关键字。要取消计时回滚并立即触发回滚，或者重置计时回滚的参数，必须输入 **configure revert** 命令。

```
Device# configure revert timer 100
```

执行配置回滚操作

以下示例展示了如何更改当前的运行配置，并在此后回滚更改。作为配置回滚操作的一部分，必须在更改文件之前保存当前的运行配置。此例中，**archive config** 命令被用来保存当前的运行配置。**configure replace** 命令生成的输出表示完成回滚仅执行了一遍操作。

注释： 使用 **archive config** 命令之前，必须配置 **path** 命令指定 Inspur INOS 配置存档的位置以及文件名前缀。

先把当前运行配置保存到配置存档中：

```
archive config
```

更改运行配置文件后，假设希望回滚这些更改并返回到进行更改前的配置。**show archive** 命令用来验证用作替换文件的配置版本。**configure replace** 命令用来回退替换配置文件：

```
Device# show archive
There are currently 1 archive configurations saved.
The next archive file will be named flash:myconfiguration-2
```

```
Archive #    Name
  0
  1          flash:myconfiguration-1 <- Most Recent
  2
  3
  4
  5
  6
  7
  8
  9
  10
Device# configure replace flash:myconfiguration-1
Total number of passes: 1
Rollback Done
```

其他参考资料

相关文档

相关主题	文档标题
配置锁	*独占配置更改访问以及访问会锁*
管理配置文件的命令	*Inspur INOS 配置基础命令参考*
关于管理配置文件的信息	*管理配置文件*

错误信息解释

描述	链接
为帮助管理员搜索并解决该版本中的系统错误信息，管理员可使用错误信息解释工具。	http://www.icntnetworks.com

标准

标准	标题
不支持新标准或修订的标准，且支持的现有标准未被修改	-

RFC

RFC	标题
不支持新 RFC 或修订的 RFC，且支持的现有 RFC 未被修改	-

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的	http://www.icntnetworks.com

技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	

使用闪存文件系统

关于闪存文件系统的信息

闪存文件系统是一个可以存储文件的单独闪存设备。该设备提供几个常用命令用来管理软件包和配置文件。设备上缺省的闪存文件系统被命名为 flash：。
从活跃设备或者任何堆栈成员的角度看，flash:特指本地闪存设备，其与所查看文件系统所在的设备相同。在设备堆栈中，可以在活跃的设备上查看众多堆栈成员的每一个闪存设备。这些闪存文件系统的名字包含相应的设备成员编号。例如，从活跃设备上查看，flash-3：代表的文件系统与设备堆栈成员 3 上的 flash：相同。使用 **show file systems** 特权 EXEC 命令来查看所有文件系统，包含设备堆栈上的闪存文件系统。
每次只有一个用户可以管理设备堆栈的软件包和配置文件。

显示可用的文件系统

为了查看设备上可用的文件系统，使用 **show file systems** 特权 EXEC 命令，单个设备的具体示例如下：

```
Device# show file systems
File Systems:
       Size(b)         Free(b)          Type      Flags      Prefixes
*     15998976         5135872         flash         rw       flash:
            -               -        opaque         rw       bs:
            -               -        opaque         rw       vb:
        52428          520138         nvram         rw       nvram:
            -               -       network         rw       tftp:
            -               -        opaque         rw       null:
```

```
                -               -       opaque       rw       system:
                -               -       opaque       ro       xmodem:
                -               -       opaque       ro       ymodem:
```

该示例展示了一个设备栈。此例中，活跃设备是设备栈成员 1；flash-2 代表位于设备栈成员 2 的文件系统；flash-3 代表位于设备栈成员 3 的文件系统，以此类似，直到 flash-9 代表位于设备栈成员 9 的文件系统。该示例也列出了事故信息目录以及插到活跃设备上的 USB 闪存驱动器。

```
Device# show file systems
File Systems:
        Size(b)         Free(b)        Type     Flags     Prefixes
      145898496         5479424        disk       rw      crashinfo:crashinfo-1:
      248512512        85983232        disk       rw      crashinfo-2:stby-crashinfo:
      146014208        17301504        disk       rw      crashinfo-3:
      146014208               0        disk       rw      crashinfo-4:
      146014208         1572864        disk       rw      crashinfo-5:
      248512512        30932992        disk       rw      crashinfo-6:
      146014208         6291456        disk       rw      crashinfo-7:
      146276352        15728640        disk       rw      crashinfo-8:
      146276352        73400320        disk       rw      crashinfo-9:
*     741621760       481730560        disk       rw      flash:flash-1:
     1622147072      1360527360        disk       rw      flash-2:stby-flash:
      729546752       469762048        disk       rw      flash-3:
      729546752       469762048        disk       rw      flash-4:
      729546752       469762048        disk       rw      flash-5:
     1622147072      1340604416        disk       rw      flash-6:
      729546752       469762048        disk       rw      flash-7:
     1749549056      1487929344        disk       rw      flash-8:
     1749549056      1487929344        disk       rw      flash-9:
              0               0        disk       rw      unix:
              -               -        disk       rw      usbflash0:usbflash0-1:
              -               -        disk       rw      usbflash0-2: stby-usbflash0:
              -               -        disk       rw      usbflash0-3:
              -               -        disk       rw      usbflash0-4:
              -               -        disk       rw      usbflash0-5:
              -               -        disk       rw      usbflash0-6:
              -               -        disk       rw      usbflash0-7:
              -               -        disk       rw      usbflash0-8:
              -               -        disk       rw      usbflash0-9:
              0               0        disk       ro      webui:
              -               -      opaque       rw      system:
              -               -      opaque       rw      tmpsys:
        2097152         2055643       nvram       rw      stby-nvram:
```

```
          -             -       nvram       rw       stby-rcsf:
          -             -      opaque       rw       null:
          -             -      opaque       ro       tar:
          -             -     network       rw       tftp:
    2097152       2055643       nvram       rw       nvram:
          -             -      opaque       wo       syslog:
          -             -     network       rw       rcp:
          -             -     network       rw       http:
          -             -     network       rw       ftp:
          -             -     network       rw       scp:
          -             -     network       rw       https:
          -             -      opaque       ro       cns:
          -             -      opaque       rw       revrcsf:
```

表 202：文件系统字段描述

域	**数值**
大小（比特）	文件系统的内存比特大小。
可用（比特）	文件系统的可用内存比特大小。
类型	文件系统类型。 **disk**——闪存内存设备、USB 或者事故信息文件文件系统。 **network**——网络设备文件系统，例如 FTP 服务器或者 HTTP 服务器。 **nvram**——非易失随机存取存储设备文件系统（NVRAM）。 **opaque**——本地生成的伪文件系统（例如，该系统）或者下载接口（例如 brimux）文件系统。 **unknown**——未知类型文件系统。
标志位	文件系统权限。 **ro**——只读。 **rw**——读/写。 **wo**——只写。
前缀	文件系统别名。 **cashinfo**——事故信息文件。 **flash**: ——闪存文件系统。 **ftp**: —— FTP 服务器。 **http**: —— HTTP 服务器。 **https**: ——安全 HTTP 服务器。 **nvram**: ——NVRAM。 **null**: ——空拷贝，你可以拷贝一个远程文件到空拷贝，然后查看该文件大小。 **rcp:** ——远程拷贝协议（Remote Copy Protocol，RCP）服务器。

	scp: ——会话控制协议（Session Control Protocol，SCP）服务器。 **system**: ——包括系统内存和当前运行配置。 **tftp**: —— TFTP 网络服务器。 **usbflash0**: —— USB 闪存内存设备。 **xmodem**: ——使用 Xmodem 协议从网络中获取文件。 **ymodem**: ——使用 Ymodem 协议从网络中获取文件。

设置默认文件系统

cd*filesystem:*特权 EXEC 命令用来指定默认文件系统的文件或目录。可以设置文件系统的相关命令省略 *filesystem:*参数。例如，对于所有带有 *filesystem:*参数的特权 EXEC 命令，可以使用 **cd** 命令指定的文件系统。
默认文件系统为 *flash:*。
可以使用 **pwd** 特权 EXEC 命令来查看由 **cd** 命令指定的当前默认文件系统。

显示文件系统上的文件信息

在对文件系统进行操作前，可以查看文件系统上的内容。例如，在把新的配置文件拷贝到至闪存之前，管理员也许希望确认文件系统上是否存在重名文件。类似情况，拷贝闪存配置文件到另一个位置之前，管理员也许想确认该文件名的文件名，以便在其他命令中使用。使用下表所列的特权 EXEC 命令来查看文件系统上的文件信息。

表 203：显示文件信息的命令

命令	描述
dir/all [*filesystem:filename*]	查看文件系统上的文件列表。
show file systems	查看文件系统每个文件的更多信息。
show file information*file-url*	查看指定某个文件的信息。
show file descriptors	查看打开文件的描述符列表。文件描述符为打开文件的内部标志，可以通过这个命令查看是否其他用户打开了某个文件。

例如，使用 **dir** 特权 EXEC 命令查看文件系统上的所有文件。

```
device# dir flash:
Directory of flash:/
7386 -rwx 2097152 Jan 23 2013 14:06:49 +00:00 nvram_config
7378 drwx 4096 Jan 23 2013 09:35:11 +00:00 mnt
7385 -rw- 221775876 Jan 23 2013 14:15:13 +00:00
cat3k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin
7389 -rwx 556 Jan 21 2013 20:47:30 +00:00 vlan.dat
712413184 bytes total (445063168 bytes free)
```

```
device#
```

改变目录与查看工作目录(CLI)

按照以下步骤改变目录并显示工作目录。

总步骤

1. **enable**
2. **dir** *filesystem*
3. **cd** *directory_name*
4. **pwd**
5. **cd**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** Device**> enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**dir** *filesystem* **示例：** Device# dir flash:	查看指定文件系统的所有目录。 对于 *filesystem:*参数，使用 flash：查看系统板闪存设备。 使用 flash-*n* 来读取设备成员栈中的闪存分区，*n* 为设备栈成员的编号，例如，flash-4。
步骤 3	**cd** *directory_name* **示例：** Device# cd new_configs	切换到指定的目录。 该命令展现了如何切换到 *new_configs* 目录。
步骤 4	**pwd** **示例：** Device# pwd	查看工作目录。
步骤 5	**cd** **示例：** Device# cd	切换到默认目录。

创建目录(CLI)

在特权 EXEC 模式中按照以下步骤创建目录。

总步骤

1. **dir***filesystem:*
2. **mkdir** *directory_name*
3. **dir** *filesystem:*

具体步骤

	命令或操作	目的
步骤 1	**dir***filesystem:* **例如：** Device# dir flash:	查看指定文件系统的所有目录。对于 *filesystem:*参数，使用 flash：查看系统板闪存设备。
步骤 2	**mkdir** *directory_name* **例如：** Device# mkdir new_configs	创建新目录。目录名称区分大小写且斜线（/）之间字符数量小于 45 个；目录名称不能包含控制字符、空格、斜线、引用符号、分号或者冒号。
步骤 3	**dir** *filesystem:* **例如：** Device# dir flash:	验证创建的目录。

删除目录

要移除目录以及其中所有文件与子目录，使用 **delete /force /recursive** *filesystem:/file-url* 特权 EXEC 命令。

使用**/recursive** 关键字来删除指定的目录以及其中包含的所有子目录与文件。使用**/force** 关键字抑制每个确认删除目录中文件的提示。在删除过程开始时，用户只被提示一次。

对于 *filesystem*，使用 **flash:**来指定系统板闪存设备。对于 *file-url*，输入要删除的目录名称。目录中的所有文件以及子目录都会移除。

注意： 目录被删除后其内容不能被恢复。

拷贝文件

要把文件从源拷贝到目的位置，请使用 **copy** *source-url destination-url* 特权 EXEC 命令。对于源以及目的 URL，可以使用 **running-config** 以及 **startup-config** 关键字。比如，**copy running-config startup-config** 命令会把当前的运行配置保存到闪存的 NVRAM 区，以在系统初始化时使用。

也可以使用Xmodem或Ymodem协议，对网络机器上的文件使用特殊的文件系统（**xmodem:**、**ymodem:**）进行拷贝。

网络文件系统URL包括ftp:、rcp:以及tftp:，语法如下：

- FTP——ftp:[[//username [:password]@location]/directory]/filename
- RCP——rcp:[[//username@location]/directory]/filename
- TFTP——tftp:[[//location]/directory]/filename

本地可写的文件系统包括flash:。

存在一些不合法的源目组合。具体来说，用户不能拷贝这些组合：

- 从运行配置到运行配置
- 从启动配置到启动配置
- 从一台设备到相同的设备（如 **copy flash: flash:**命令是非法的）

把文件从一个堆栈中的设备拷贝到相同堆栈中的另一台设备

要把文件从一个堆栈中的设备拷贝到相同堆栈中的另一台设备，使用 **flash-X:**标注，其中 **X** 是设备编号。

要查看堆栈中的所有设备，请在特权EXEC模式中使用 **show switch** 命令。如以下有9个成员的设备堆栈示例所示：

```
Device# show switch
Switch/Stack Mac Address : 0006.f6b9.b580 - Local Mac Address Mac persistency wait time:
Indefinite
H/W Current
Switch# Role Mac Address Priority Version State
------------------------------------------------------------
*1 Active 0006.f6b9.b580 15 P3B Ready
2 Standby 0006.f6ba.0c80 14 P3B Ready
3 Member 0006.f6ba.3300 7 P3B Ready
4 Member 0006.f6b9.df80 6 P3B Ready
5 Member 0006.f6ba.3880 13 P1A Ready
6 Member 1ce6.c7b6.ef00 4 PP Ready
7 Member 2037.06ce.2580 3 P2A Ready
8 Member 2037.0653.7e00 2 P5A Ready
9 Member 2037.0653.9280 1 P5B Ready
```

要显示特定设备上的所有可用文件系统，请使用 **copy** 命令，如以下有5个成员的堆栈示例所示：

```
Device# copy flash: ?
crashinfo-1: Copy to crashinfo-1: file system
crashinfo-2: Copy to crashinfo-2: file system
crashinfo-3: Copy to crashinfo-3: file system
crashinfo-4: Copy to crashinfo-4: file system
crashinfo-5: Copy to crashinfo-5: file system
crashinfo: Copy to crashinfo: file system
flash-1: Copy to flash-1: file system
flash-2: Copy to flash-2: file system
```

```
flash-3: Copy to flash-3: file system
flash-4: Copy to flash-4: file system
flash-5: Copy to flash-5: file system
flash: Copy to flash: file system
ftp: Copy to ftp: file system
http: Copy to http: file system
https: Copy to https: file system
null: Copy to null: file system
nvram: Copy to nvram: file system
rcp: Copy to rcp: file system
revrcsf: Copy to revrcsf: file system
running-config Update (merge with) current system configuration
scp: Copy to scp: file system
startup-config Copy to startup configuration
stby-crashinfo: Copy to stby-crashinfo: file system
stby-flash: Copy to stby-flash: file system
stby-nvram: Copy to stby-nvram: file system
stby-rcsf: Copy to stby-rcsf: file system
stby-usbflash0: Copy to stby-usbflash0: file system
syslog: Copy to syslog: file system
system: Copy to system: file system
tftp: Copy to tftp: file system
tmpsys: Copy to tmpsys: file system
usbflash0-1: Copy to usbflash0-1: file system
usbflash0-2: Copy to usbflash0-2: file system
usbflash0-3: Copy to usbflash0-3: file system
usbflash0-4: Copy to usbflash0-4: file system
usbflash0-5: Copy to usbflash0-5: file system
usbflash0: Copy to usbflash0: file system
Device#
```

此示例展示了如何把保存在设备 2 闪存分区的配置文件拷贝到设备 4 的闪存分区中。示例中假设设备 2 和设备 4 在相同堆栈。

```
Device# copy flash-2:config.txt flash-4:config.txt
```

删除文件

如何不再需要一个闪存设备上的文件，可以将其永久删除。要删除指定闪存设备的文件或目录，请使用特权 EXEC 命令 **delete** [**/force**] [**/recursive**] [*filesystem:*]*/file-url*。

使用**/recursive** 关键字删除一个目录以及其中包含的所有子目录与文件。使用**/force** 关键字来抑制每个提示用户确认删除目录中文件的确认提示。用户仅在此删除过程开始时会被提示一次。使用**/force** 以及**/recursive** 关键字，可以删除由 **archive download-sw** 命令安装但不再需要的软件镜像。

如果省略了 *filesystem:*选项，设备会使用 **cd** 命令指定的默认设备。对于 *file-url*，用户需指定

要删除文件的路径（目录）以及文件名。
尝试删除任何文件时，系统会提示用户确认删除。
注意： 文件被删除后无法被恢复。
此示例展示了如何删除默认闪存设备中的 *myconfig* 文件：

```
Device# delete myconfig
```

创建、显示与提取文件

用户可以创建一个文件并向其中写入多个文件、列出一个文件中的多个文件，并从一个文件中提取出多个文件，如下一节所述。
在特权 EXEC 模式中按照以下步骤创建文件，显示文件内容并提取文件：

总步骤

1. **archive tar /create** *destination-url* **flash:** */file-url*
2. **archive tar /table** *source-url*
3. **archive tar /xtract** *source-url* **flash:***/file-url* [*dir/file...*]
4. **more** [**/ascii** | **/binary** | **/ebcdic**] */file-url*

具体步骤

	命令或操作	目的
步骤 1	**archive tar /create** *destination-url***flash:** */file-url* **示例：** `device# archive tar /create tftp:172.20.10.30/saved. flash:/new-configs`	创建一个文件并向其中添加文件。 对于 *destination-url*，请指定本地或网络文件系统的别名目的 URL，以及要创建的文件名： • 本地闪存文件系统语法：**flash:** • FTP 语法：**ftp**:[[*//username*[:*password*]*@location*]/*directory*]/*-filename* • RCP 语法：**rcp**:[[*//username@location*]/*directory*]/*-filename* • TFTP 语法：**tftp**:[[*//location*]/*directory*]/*-filename* 对于 **flash:***/file-url*，请指定要创建新文件的本地文件系统位置。也可以在源目录中指定可选的要添加到新文件中的文件或目录列表。如果不指定，此级别下的所有文件与目录都会被写到新创建的文件中。
步骤 2	**archive tar /table** *source-url* **示例：** `device# archive tar /tableflash:`	显示文件内容。 对于 *source-url*，请指定本地或网络文件系统的别名源 URL。*-filename.*是要显示的

	/new_configs	文件。支持以下选项： • 本地闪存文件系统语法：**flash:** • FTP 语法：**ftp**:[[//*username*[:*password*]@*location*]/*directory*]/*-filename* • RCP 语法：**rcp**:[[//*username@location*]/*directory*]/*-filename* • TFTP 语法：**tftp**:[[//*location*]/*directory*]/*-filename* 也可以在此文件后指定一个文件列表，限制显示的文件。此时仅会显示这些文件。如果未指定，所有文件及目录都会显示。
步骤 3	**archive tar /xtract** *source-url* **flash:**/*file-url* [*dir/file...*] **示例：** device# archive tar /xtract tftp:/172.20.10.30/saved.flash:/new-configs	把文件提取到闪存文件系统的一个目录中。 对于 *source-url*，指定本地文件系统的源 URL。 *-filename* 是要从中提取文件的文件名。支持以下选项： • 本地闪存文件系统语法：**flash:** • FTP 语法：**ftp**:[[//*username*[:*password*]@*location*]/*directory*]/*-filename* • RCP 语法：**rcp**:[[//*username@location*]/*directory*]/*-filename* • TFTP 语法：**tftp**:[[//*location*]/*directory*]/*-filename* 对于 **flash:**/*file-url* [*dir/file...*]，指定文件要被提取到的本地文件系统位置。使用 *dir/file...*选项指定在文件中的要被提取的文件或目录列表。如果未指定，所有文件及目录都会被提取。
步骤 4	**more** [**/ascii** | **/binary** | **/ebcdic**]/*file-url* **示例：** device# moreflash:/new-configs	显示任何可读文件的内容，包括远程文件系统中的文件。

其他参考资料

相关文档

相关主题	文档标题
管理 flash:文件系统的命令	*Inspur INOS 配置基础命令参考*

错误信息解释

描述	链接
为帮助管理员搜索并解决该版本中的系统错误信息，管理员可使用错误信息解释工具。	http://www.icntnetworks.com

标准

标准	标题
不支持新标准或修订的标准，且支持的现有标准未被修改	-

RFC

RFC	标题
不支持新 RFC 或修订的 RFC，且支持的现有 RFC 未被修改	-

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	http://www.icntnetworks.com

条件性调试和放射性跟踪

查询特征信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

条件性调试的介绍

条件性调试特性允许管理员基于定义的一组条件选择性地为特定特性启用调试及日志功能。此功能在支持大量特征的系统中非常有用。
注释: 在 InspurINOS11.3.1 中只支持控制平面上的跟踪。
条件性调试允许对规模较大且运行大量特性的网络进行粒度化的调试。它允许管理员观察系统中粒度化实例的详细调试信息。当需要仅调试成千上万个会话中的特定会话时，这是非常有用的。该特性也可以指定多个条件。
条件是指特征或身份，其中身份可以是接口、IP 地址或 MAC 地址等等。
注释: 在 InspurINOS11.3.1 中支持的条件只有 MAC 地址，对其他功能的支持将在随后的版本中引入。
该特性与一般调试命令不同。一般调试命令产生的输出不区分正在处理的特性对象，会消耗大量的系统资源，且影响系统性能。

放射性跟踪的介绍

放射性跟踪可以按照详细度递增的方式把系统中希望执行的操作连接成链。这提供了一种条件性地打印线程、进程以及功能调用之间调试信息的方式（到 DEBUG 级别或指定级别）。
注释: 在 InspurINOS11.3.1 中默认级别是 **DEBUG**，用户不能更改为其他级别。对其他级别的支持将在随后的版本中引入。

条件性调试和放射性跟踪

放射性跟踪结合条件性调试，使管理员能够在一个单独的调试 CLI 中调试所有与条件相关的执行上下文。可以在不知道设备中各种特性的控制流过程的情况下完成此操作，且无需单独为这些过程输入调试命令。

跟踪文件的位置

默认情况下，将为每个进程生成跟踪文件日志，并保存到/tmp/rp/trace 或/tmp/fp/trace 目录。

在此临时目录中，跟踪日志会被写入文件，每个文件大小为 1 MB。对于特定的进程，该目录最多可容纳 25 个这样的文件。当一个在/tmp 目录中的跟踪文件达到其 1MB 限制或在启动期间为其配置的任何大小限制时，它转存到位于 tracelogs 目录下的/crashinfo 分区中的归档位置。

/tmp 目录仅保留特定进程的单个跟踪文件。一旦文件达到其文件大小限制会被转存到/crashinfo/tracelogs。在归档目录中，最多累积 25 个文件，之后最旧的一个会被来自/tmp 的新转存文件替换。

crashinfo 目录中的跟踪文件格式如下：

1 Process-name_Process-ID_running-counter.timestamp.gz
  示例: INOSRP_R0-0.bin_0.14239.20151101234827.gz
2 Process-name_pmanlog_Process-ID_running-counter.timestamp.bin.gz
  示例: wcm_pmanlog_R0-0.30360_0.20151028233007.bin.gz

配置条件性调试

按照以下步骤配置条件性调试：

总步骤

1. enable
2. debug platform condition mac {*mac-address*}
3. debug platform condition start
4. show platform condition OR show debug
5. debug platform condition stop
6. request platform software trace archive [last {*number*} days] [target {*crashinfo: | flashinfo:*}]
7. request platform software trace filter-binary {*wire* } [context {*mac-address*} | level | module]
8. show platform software trace [filter-binary | level | message]
9. clear platform condition all

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>enable`	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**debug platform condition mac** *{mac-address}* **示例：** `Device# debug platform condition mac bc16.6509.3314`	为指定 MAC 地址配置条件性调试。
步骤 3	**debug platform condition start** **示例:** Device# **debug platform condition start**	开始条件性调试（如果在上述条件之一上有匹配，则将启动放射性跟踪）。
步骤 4	**show platform condition OR show debug** **示例:**	显示当前设置的条件。

	Device# **show platform condition** Device# **show debug**	
步骤 5	**debug platform condition stop** **示例:** Device# **debug platform condition stop**	停止条件性调试（会停止放射性跟踪）。
步骤 6	**request platform software trace archive [last{number} days] [target** *{crashinfo:* **|** *flashinfo:}***]** **示例:** Device# **request platform software trace archive last 2 days**	（可选）显示系统上合并的 tracefiles 的历史日志。按照任意天数或位置的组合进行过滤。
步骤 7	**request platform software trace filter-binary{***wire* **} [context {***mac-address***} | level | module]**	（可选）过滤模块以核对信息，然后在上下文中核对指定的 Mac 地址的。这些日志可以离线查看。
步骤 8	**show platform software trace [filter-binary | level| message]** **示例:** Device# **show platform software trace message**	（可选）显示从最新跟踪文件合并的日志。按照应用条件、跟踪模块名称和跟踪级别的任意组合进行过滤。 • **filter-binary** - 过滤模块以进行核对。 • **level** - 显示跟踪级别。 • **message** - 显示跟踪消息环内容 注释在设备上： ・ 除了 linux shell，也可从 INOS 控制台进行。 ・ 在设备生成包含合并日志的文件。 ・ 仅显示暂存区域的合并日志。
步骤 9	**clear platform condition all** **示例:** Device# **clear platform condition all**	清除所有条件。

接下来做什么？

注释： 命令 **request platform software trace filter-binary** 和 **show platform software trace filter-binary** 工作方式类似。唯一的区别是：

- **request platform software trace filter-binary** 从历史日志信息中获得数据。
- **show platform software trace filter-binary** 从闪存 Temp 目录中获得数据

其中，*mac_log <..date..>*是最重要的文件，因为它给出了正在调试的 MAC 的消息。

命令 **show platform software trace filter-binary** 也生成相同的 Flash 文件，并且还在屏幕上打印 mac_log。

对 L2 组播进行放射性跟踪

要识别特定的组播接收者，请指定加入者或接收者客户端的 MAC 地址，组组播 IP 地址和侦

听 VLAN。此外，启用调试的跟踪级别。此调试级别将提供详细的跟踪信息以及更好的系统可见性。

debug platform condition feature multicast controlplane mac *client MAC address***ip***Group IP address***vlan***id***level** *debug level*

跟踪文件的推荐工作流

跟踪文件的建议工作流程如下所示：

1 请求特定时间段的跟踪日志。
例如要请求 1 天的日志，使用命令：
Device#request platform software trace archive last 1 day

2 系统在位置/flash:中生成跟踪日志的 tarball（.gz 文件）

3 将文件从交换机上复制。通过复制文件，可以离线使用跟踪日志。有关复制文件的更多详细信息，请参阅下面的部分。

4 从/ flash：location 中删除跟踪日志文件（.gz）文件。这将确保交换机有足够的空间用于其他操作。

拷贝出文件

跟踪文件的示例如下：

```
Device# dir crashinfo:/tracelogs
Directory of crashinfo:/tracelogs/
50664 -rwx 760 Sep 22 2015 11:12:21 +00:00 plogd_F0-0.bin_0.gz
50603 -rwx 991 Sep 22 2015 11:12:08 +00:00 fed_pmanlog_F0-0.bin_0.9558.20150922111208.gz
50610 -rw- 11 Nov 2 2015 00:15:59 +00:00 timestamp
50611 -rwx 1443 Sep 22 2015 11:11:31 +00:00
auto_upgrade_client_sh_pmanlog_R0-.bin_0.3817.20150922111130.gz
50669 -rwx 589 Sep 30 2015 03:59:04 +00:00 cfgwr-8021_R0-0.bin_0.gz
50612 -rwx 1136 Sep 22 2015 11:11:46 +00:00 reflector_803_R0-0.bin_0.1312.20150922111116.gz
50794 -rwx 4239 Nov 2 2015 00:04:32 +00:00 INOSRP_R0-0.bin_0.14239.20151101234827.gz
50615 -rwx 131072 Nov 2 2015 00:19:59 +00:00 linux_INOSd_image_pmanlog_R0-0.bin_0
--More--
```

可以使用以下各种选项之一复制跟踪文件：

```
Device# copy crashinfo:/tracelogs ?
crashinfo: Copy to crashinfo: file system
flash: Copy to flash: file system
ftp: Copy to ftp: file system
http: Copy to http: file system
https: Copy to https: file system
null: Copy to null: file system
nvram: Copy to nvram: file system
rcp: Copy to rcp: file system
running-config Update (merge with) current system configuration
```

```
scp: Copy to scp: file system
startup-config Copy to startup configuration
syslog: Copy to syslog: file system
system: Copy to system: file system
tftp: Copy to tftp: file system
tmpsys: Copy to tmpsys: file system
```

复制到 TFTP 服务器的一般语法如下：

```
Device# copy source: tftp:
Device# copy crashinfo:/tracelogs/INOSRP_R0-0.bin_0.14239.20151101234827.gz tftp:
Address or name of remote host []? 2.2.2.2
Destination filename [INOSRP_R0-0.bin_0.14239.20151101234827.gz]?
```

注释： 清除交换机上生成的报告或归档文件很重要，以便给 tracelog 和其他用途留出闪存空间。

条件性调试配置示例

以下是 *show platform condition* 命令的输出示例。

```
Device# show platform condition
Conditional Debug Global State: Stop
Conditions Direction
------------------------------------------------------------------------------------------
    --------|---------
MAC Address 0024.D7C7.0054 N/A
Feature Condition Type Value
----------------------|----------------------|--------------------------------
Device#
```

以下是 *show debug* 命令的输出示例。

```
Device# show debug
INOSXE Conditional Debug Configs:
Software Configuration Guide, Inspur INOS 11.3.1 (Inspur 6650 Switches)
2769
Monitoring Conditional Debugging
Conditional Debug Global State: Start
Conditions Direction
------------------------------------------------------------------------------------------
    --------|---------
MAC Address 0024.D7C7.0054 N/A
Feature Condition Type Value
----------------------|----------------------|--------------------------------
Packet Infra debugs:
Ip Address Port
------------------------------------------------------|----------
Device#
```

以下是 *debug platform condition stop* 命令的输出示例。

```
Device# debug platform condition stop
Conditional Debug Global State: Stop
```

监控条件性调试

下表显示了可用于监视条件性调试的各种命令。

命令	目的
show platform condition	显示当前的条件设置。
show debug	显示当前的调试条件设置。
show platform software trace filter-binary	显示从最新跟踪文件合并的日志。
request platform software trace filter-binary	显示系统上合并的跟踪文件的历史日志。

软件配置故障排除

本章介绍了如何发现并解决交换机上与 Inspur INOS 系统相关软件问题。根据问题的性质，用户可以使用命令行界面（CLI）、设备管理器或网络助手来发现并解决问题。
至于其他的故障排除信息，如 LED 说明，会在硬件安装指南中提供相关的说明。

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于软件配置故障排除的信息

交换机上的软件故障

交换机在升级期间如果下载了错误的文件或者删除了镜像文件，那么可能会导致交换机软件的损坏。在所有类似的情况下，交换机将无法通过开机自检（power-on self-test，POST），并且无法提供连通性。

设备密码遗忘或丢失

在交换机通电启动期间，设备的默认配置允许那些对设备具有物理访问权限的终端用户通过中断引导程序并输入新密码而恢复使用。这些恢复过程要求用户具有对设备的物理访问权限。

注释： 在这些设备上，系统管理员可以禁用某些特性的一些功能，只有当终端用户同意设备恢复出厂设置，终端用户才能为设备重设密码。如果终端用户想在密码恢复功能被禁用的时候重设密码，系统会有一个状态消息来提醒用户，设备在恢复的过程中会回到出厂设置。

以太网供电端口

如果交换机检测到电路上没有电，以太网供电（Power over Ethernet，PoE）的交换机端口会自动为以下这些连接的设备供电：

- Inspur 准标准化用电设备（例如 Inspur IP 电话或 Inspur Aironet 接入点）
- 兼容 IEEE 802.3af 的用电设备
- 兼容 IEEE 802.3at 的用电设备

当用电设备连接到以太网供电的交换机端口以及 AC（交流）电源时，设备可以有冗余电力。只连接到 PoE 端口时，设备没有冗余电力。

在交换机检测到用电设备之后，交换机会去确定用电设备功率需求，然后决定是否准许为设备供电。交换机还可以通过监视和限制电量的使用情况，来检测设备的实时功耗。

如要查看更多详细信息，请参阅*接口和硬件组件配置指南（Inspur 6650 交换机）*中的“配置 PoE”一章。

断电导致端口被禁用

如果一个连接到 PoE 设备端口并且由 AC 电源供电的用电设备（例如 Inspur IP 电话 7910）失去了 AC 电源供电，该设备可能会进入错误禁用的状态。要从错误禁用状态中恢复，请输入 **shutdown** 接口配置命令，然后输入 **no shutdown** 接口命令。管理员还可以在设备上配置

自动恢复，使得设备恢复正常运行。
在一个设备上，**errdisable recovery cause loopback** 和 **errdisable recovery interval***seconds* 全局配置命令支持在特定的时间段后，使得端口自动从错误禁用状态中恢复。

错误 Link-Up 导致端口被禁用
在用电设备已经连接到某端口的情况下，如果用户使用 **power inline never** 接口配置命令配置此端口，可能会出现错误 link-up 状态，端口也将进入错误禁用状态。要使端口从错误禁用状态中恢复，输入 **shutdown** 和 **no shutdown** 接口配置命令。
请勿将 Inspur 用电设备连接到已使用 **power inline never** 命令进行配置的端口。

Ping

设备支持 IP ping，用户可以使用该命令测试与远程主机的连接性。Ping 向目的地址发送一个回显请求包，并等待回复。Ping 会返回的响应如下：

- 正常响应——正常响应（*主机名*是存活的）发生在 1 到 10 秒内，具体取决于网络流量状况。
- 目标不响应——如果主机没有响应，则返回*无应答*消息。
- 未知主机——如果主机不存在，则返回*未知主机*消息。
- 目的地不可达——如果默认网关不能到达指定的网络，则返回*目的地不可达*消息。
- 网络或主机不可达——如果主机或网络的路由表中没有路由表，则返回*网络或主机不可达*的消息。

二层 Traceroute

第 2 层 traceroute 特性允许交换机识别数据包从源设备到目标设备的物理路径。二层 traceroute 只支持单播的源目 MAC 地址。
traceroute 使用路径上设备的 MAC 地址表查找路径。当设备检测到路径中有不支持第 2 层 traceroute 的设备时，设备继续发送第 2 层 trace 查询请求并使其超时。
设备只能识别从源设备到目的设备的路径。它不能识别从源主机到源设备或从目的设备到目的主机的路径。

二层 Traceroute 指南

- 网络中的所有设备必须启用 Inspur 发现协议（Cisco Discovery Protocol，CDP）。如果要使二层 traceroute 正常工作，请不要禁用 CDP。
  如果物理路径中的某些设备对 CDP 透明，交换机则不能识别经过这些设备的路径。
- 当管理员使用特权 EXEC**ping** 命令可检测到两设备间的连通性，则说明一个设备到另一个设备具有可达性。物理路径中的所有设备必须彼此可达。
- 路径中的最大跳数为 10。
- 在不位于从源设备到目的设备的物理路径的交换机上，管理员可输入特权 EXEC**traceroute mac** 或 **traceroute mac ip** 命令。其中，这个交换机必须可以连通路径中的所有设备。

- 只有当指定的源目 MAC 地址属于同一 VLAN 时，**traceroute mac** 命令才会输出二层的路径。如果指定的源目 MAC 地址属于不同 VLAN，则无法识别第 2 层路径，并输出错误消息。
- 如果指定了组播源 MAC 地址或目的 MAC 地址，路径不会被标识出，且会输出错误消息。
- 如果源或目的 MAC 地址属于多个 VLAN，必须指定源目 MAC 地址同时属于的 VLAN。如果不指定此 VLAN，路径不会被标识出，且会输出错误消息。
- 当指定的源目 IP 地址属于同一子网时，**traceroute mac ip** 命令输出二层路径。当管理员指定 IP 地址时，设备使用地址解析协议（Address Resolution Protocol，ARP）将 IP 地址与相应的 MAC 地址和 VLAN ID 关联起来。
    - 如果指定 IP 地址存在 ARP 表项，设备将使用关联的 MAC 地址并标识该物理路径。
    - 如果 ARP 表项不存在，设备将发送 ARP 查询并尝试解析 IP 地址。如果无法解析 IP 地址，则无法识别路径，并显示错误消息。
- 当多个设备通过集线器连接到一个端口时（例如，在端口上检测到多个 CDP 邻居），无法支持二层 traceroute 特性。当端口检测到多个 CDP 邻居时，无法识别二层路径，并显示错误消息。
- 令牌环 VLAN 中不支持二层 traceroute 特性。

IP Traceroute

管理员可以使用 IP traceroute 追踪数据包通过网络时的逐跳路径。该命令的输出会显示数据包在到达目的地的途中经过的所有网络层（第 3 层）设备，例如路由器。

管理员的设备可以作为 **traceroute** 特权 EXEC 命令的源或目的设备，并且可选择是否作为其中一跳在 **traceroute** 命令输出中显示。如果该设备是 traceroute 的目的设备，在 traceroute 输出中它将显示为最终目的设备。如果中间设备仅将数据包从某端口桥接到同 VLAN 中的另一端口，该设备将不会在 traceroute 输出中显示。但是，如果中间设备是一个多层的、为特定数据包进行路由的设备，该设备将会作为一跳在 traceroute 输出中显示。

traceroute 特权 EXEC 命令通过使用 IP 头中的生存时间（Time to live，TTL）字段，让路由器和服务器生成特定的返回消息。Traceroute 首先向目的主机发送 TTL 字段置为 1 的用户数据报协议（User Datagram Protocol，UDP），如果路由器发现 TTL 值为 1 或 0，则丢弃该数据报并向发送端发送网络控制消息协议（Internet Control Message Protocol，ICMP）生存时间超时的消息。Traceroute 通过检查 ICMP 生存时间超时消息的源地址字段来找出第一跳的地址。为了识别下一跳，traceroute 发送一个 TTL 值为 2 的 UDP 包。第一个路由器将 TTL 字段减 1，并将数据报送至下一个路由器。第二个路由器看到 TTL 值为 1，丢弃数据报，并将生存时间超时消息返给源。这个过程将持续到 TTL 值增到足以使得数据报到达目的主机（或者 TTL 值增到最大）。

为了了解数据报何时到达其目的地，traceroute 将数据报中 UDP 的目的端口号设置为目的主机不太可能使用的超大值。当主机接收到包含本地未使用的目的端口号的数据报时，它会向源端发送 ICMP *端口不可达*错误。因为除了端口不可达之外的所有错误都来自中间跳，所以接收到端口不可到达错误意味着该消息由目的端口发送。

时域反射器指南

用户可以使用时域反射器（Time Domain Reflector，TDR）特性来诊断并解决布线问题。当运行 TDR 时，本地设备通过电缆发送信号，并将反射信号与初始信号进行比较。

TDR 支持 10/100/1000 的铜线以太网端口和千兆位以太网（100Mbps / 1 / 2.5 / 5/10 Gbps）端口。 SFP 模块端口不支持 TDR。

TDR 可以检测到以下线路问题：

- 未闭合的、断开的或切断的双绞线——电线未连接到远程设备的电线。
- 双绞线短路——电线彼此接触或与远程设备的电线接触。例如，如果双绞线的一条电线焊接到另一条电线上，则双绞线发生短路。

  如果双绞线中的一条电线未闭合，TDR 可找到未闭合点位置。

注释： 当千兆位以太网端口使用此特性时，仅当检测到电线未闭合或短路情况时才显示故障位置。

以下情况中可使用 TDR 诊断并解决线路问题：

- 设备替换
- 建立配线柜
- 当链接无法建立或不正常运行时，对两个设备之间的连接进行故障排除

运行 TDR 时，设备会在以下情况报告准确的信息：

- 用于千兆位链路的电缆是实芯电缆。
- 末端未闭合的电缆无终点。

运行 TDR 时，设备不会在以下情况报告准确的信息：

- 用于千兆位链路的电缆是双绞线电缆或与实芯电缆串联。
- 链路是 10 兆位或 100 兆位的链路。
- 电缆是绞合电缆。
- 链接伙伴是 Inspur IP 电话。
- 链接伙伴不兼容 IEEE 802.3 标准。

调试命令

注意： 由于调试的输出在 CPU 进程中被分配了较高优先级，它可能导致系统不可用。因此，仅在排除特定问题或与 Inspur 技术支持人员进行故障排除会话期间使用 **debug** 命令。最好在网络流量较低和用户较少时使用调 **debug** 命令。在此期间的调试可减少增加的 **debug** 命令处理开销影响系统使用的可能性。

所有 **debug** 命令都在特权 EXEC 模式下输入，大多数 **debug** 命令不带参数。

系统报告

系统报告或 crashinfo 文件保存的信息有助于 Inspur 技术代表人员调试 Inspur INOS 镜像失败（崩溃）的问题。因此，有必要快速地、可靠地收集具有高保真和完整性的关键崩溃信息。此外，也有必要收集并打包该信息，让其可以关联或标识特定的崩溃事件。

系统报告会在以下情况产生：

- 在交换机故障的情况下，系统报告会在故障的成员上生成；堆栈中的其他成员不会生成报告。

- 在交换机切换的情况下，仅在高可用性的（High Available，HA）成员交换机上生成系统报告；non- HA 成员交换机不会生成报告。

系统不会在重载的情况下生成报告。

在进程崩溃期间，交换机从本地收集以下信息：

1 完整进程的核心

2 跟踪日志

3 INOS 系统日志（在非活动崩溃的情况下不保证有该日志）

4 系统进程信息

5 启动日志

6 重载日志

7 某些类型的/ proc 信息

这些信息分别存储在单独的文件中，然后压缩并存到一个包中。这样方便管理员通过一个文件得到崩溃快照，然后将其移出进行分析。这个报告会在交换机切换到 rommon / bootloader 模式之前生成。

除了完整的核心和跟踪日志，其他文件都以文本形式存储。

Crashinfo 文件

默认情况下，生成的系统报告文件都将保存到/crashinfo 目录中。如果 crashinfo 分区空间不足，文件将被保存到/flash 目录。

如果要显示文件，请输入 **dir crashinfo:** 命令。下面是 crashinfo 目录的输出示例：

```
Switch#dir crashinfo:
Directory of crashinfo:/
46553 drwx 1024 Jun 29 2015 14:52:09 +00:00 ap_crash
12 -rw- 0 Jan 1 1970 00:00:11 +00:00 koops.dat
11 -rw- 0 Mar 22 2013 07:50:30 +00:00 deleted_crash_files
13 -rwx 594269 Mar 22 2013 07:50:30 +00:00 crashinfo_platform_mgr_20130322-075017-UTC
14 -rw- 44 Sep 9 2015 09:28:47 +00:00 last_crashinfo
15 -rw- 355 Sep 9 2015 09:29:31 +00:00 last_systemreport_log
16 -rw- 105753 Mar 22 2013 07:50:47 +00:00 system-report_1_20130322-075017-UTC.gz
17 -rw- 39 Sep 9 2015 09:29:31 +00:00 last_systemreport
18 -rwx 585996 Mar 22 2013 08:01:58 +00:00 crashinfo_platform_mgr_20130322-080144-UTC
19 -rw- 105065 Mar 22 2013 08:02:15 +00:00 system-report_1_20130322-080144-UTC.gz
20 -rwx 3426209 Sep 9 2015 06:49:12 +00:00 crashinfo_INOSd_20150909-064754-UTC
21 -rwx 9540376 Sep 9 2015 06:49:13 +00:00 fullcore_INOSd_20150909-064754-UTC
22 -rw- 469476 Sep 9 2015 06:49:56 +00:00 system-report_1_20150909-064754-UTC.gz
23 -rwx 3425350 Sep 9 2015 09:28:47 +00:00 crashinfo_INOSd_20150909-092728-UTC
24 -rwx 9535535 Sep 9 2015 09:28:47 +00:00 fullcore_INOSd_20150909-092728-UTC
25 -rw- 459709 Sep 9 2015 09:29:28 +00:00 system-report_1_20150909-092728-UTC.gz
26 -rw- 0 Sep 22 2015 11:11:33 +00:00 tracelogs.J8C
50601 drwx 10240 Oct 28 2015 22:42:50 +00:00 tracelogs
248354816 bytes total (204800000 bytes free)
```

系统报告位于 crashinfo 目录中，格式如下：

```
system-report_[switch number]_[date]-[timestamp]-UTC.gz
```

交换机崩溃后，请检查系统报告文件。最新生成的系统报告文件名存储在 crashinfo 目录下的 last_systemreport 文件中。系统报告和 crashinfo 文件可以帮助 TAC 解决故障问题。
生成的系统报告可以使用 TFTP、HTTP 或其他选项进行复制。

```
Switch#copy crashinfo: ?
crashinfo: Copy to crashinfo: file system
flash: Copy to flash: file system
ftp: Copy to ftp: file system
http: Copy to http: file system
https: Copy to https: file system
null: Copy to null: file system
nvram: Copy to nvram: file system
rcp: Copy to rcp: file system
running-config Update (merge with) current system configuration
scp: Copy to scp: file system
startup-config Copy to startup configuration
syslog: Copy to syslog: file system
system: Copy to system: file system
tftp: Copy to tftp: file system
tmpsys: Copy to tmpsys: file system
```

复制到 TFTP 服务器的一般语法如下：

```
Switch#copy crashinfo: tftp:
Source filename [system-report_1_20150909-092728-UTC.gz]?
Address or name of remote host []? 1.1.1.1
Destination filename [system-report_1_20150909-092728-UTC.gz]?
```

可以通过发出跟踪存档命令收集堆栈中所有成员的跟踪日志。该命令提供时间段选项。命令语法如下：

```
Switch#request platform software trace archive ?
last Archive trace files of last x days
target Location and name for the archive file
```

可以收集过去 6650 天内存储在 crashinfo：或 flash：目录中的跟踪日志。

```
Switch# request platform software trace archive last ?
<1-6650> Number of days (1-6650)
Switch#request platform software trace archive last 6650 days target ?
crashinfo: Archive file name and location
flash: Archive file name and location
```

注意： 为了有足够的空间存储跟踪日志或用作其他目的，一旦系统报告或跟踪存档被拷贝出去，及时从 flash 或 crashinfo 目录下清除它们很重要。

交换机上的板载故障记录

用户可使用板载故障日志记录（OnBoard Failure Logging，OBFL）功能来收集有关设备的信息。这些信息包括正常的运行时间，温度和电压信息，它们可以帮助 Inspur 技术支持人员排除设备故障。我们建议用户保持 OBFL 启用状态，不要擦除闪存中存储的数据。
默认情况下，OBFL 处于启用状态。OBFL 收集了关于该设备和小型可插拔模块（Small

Form-factor Pluggable，SFP）的信息。这些信息存储在设备中的闪存中：

- CLI 命令——在独立设备或交换机堆叠成员上输入的 OBFL CLI 命令的记录。
- 环境数据——独立设备或交换机堆栈成员以及所有连接的 FRU 设备的唯一设备标识符（Unique Device Identifier，UDI）信息：产品标识（Product Identifier，PID），版本标识（Version Identifier，VID）以及序列号。
- 消息——由独立设备或交换机堆栈成员生成的与硬件相关的系统消息的记录。
- 以太网供电（Power over Ethernet，PoE）——独立设备或交换机堆栈成员上 PoE 端口的功耗记录。
- 温度——独立设备或交换机堆栈成员的温度。
- 正常运行时间的数据——独立设备或交换机堆栈成员启动用时，设备重新启动的原因，以及设备自上次重启以来运行的时长。
- 电压——独立设备或交换机堆叠成员的系统电压。

用户应手动设置系统时钟或使用网络时间协议（Network Time Protocol，NTP）配置。
当设备运行时，用户可以使用 **show logging onboard** 特权 EXEC 命令获取 OBFL 数据。如果设备出现故障，请联系您的 Inspur 技术支持人员了解如何获取数据。
当重新启动已启用 OBFL 的设备时，在开始记录新数据之前会有 10 分钟的延迟。

风扇故障

默认情况下，该特性被禁用。当现场可更换单元（Field-Replaceable Unit，FRU）或电源中有多个风扇出现故障时，设备不会关闭，并显示以下错误消息：

```
Multiple fan(FRU/PS) failure detected. System may get overheated. Change fan quickly.
```

设备可能会过热并关闭。
要启用风扇故障特性，请输入 **system env fan-fail-action shut** 特权 EXEC 命令。如果设备中有多个风扇出现故障，设备将自动关闭，并显示以下错误消息：

```
Faulty (FRU/PS) fans detected, shutting down system!
```

在第一个风扇关闭后，如果设备检测到有第二个风扇故障，设备将会等待 20 秒后再关闭。若要重新启动设备，必须关闭后再打开。

高 CPU 使用率的可能征兆

CPU 利用率过高可能会导致以下症状，但这些症状也可能由其他原因引起：

- 生成树拓扑变化
- 由于通信中断导致的 EtherChannel 链路关闭
- 停止响应管理请求（ICMP ping，SNMP 超时，Telnet 或 SSH 会话速度变慢）
- UDLD 抖动
- 由于 SLA 响应超过可接受的门限值导致的 IP SLA 失效
- DHCP 或 IEEE802.1x 失败（如果交换机不能转发或响应请求）

如何进行软件配置故障排除

从软件故障中恢复

在开始前

恢复过程要求用户具有对交换机的物理访问权限。

此过程使用引导程序命令和 TFTP 从损坏或不正确的镜像文件中恢复。

步骤 1 在 PC 端从 incntnetworks.com 下载软件镜像文件（image.bin）。

步骤 2 将软件的镜像加载到 TFTP 服务器中。

步骤 3 将 PC 连接到交换机以太网管理端口。

步骤 4 拔下交换机电源线。

步骤 5 按下 **Mode** 按钮的同时将电源线重新连接到交换机。

步骤 6 在引导程序 (ROMMON)提示符处，确保可以 ping 通 TFTP 服务器。

a） 设置 IP 地址 **switch: set IP_ADDR**ip_address subnet_mask

示例：

```
switch: set IP_ADDR 192.0.2.123/255.255.255.0
```

b） 设置默认路由 IP 地址 **switch: set DEFAULT_ROUTER**ip_address

示例：

```
switch: set DEFAULT_ROUTER 192.0.2.1
```

c） 请确认您可以 ping 通 TFTP 服务器 **switch: ping** ip_address_of_TFTP_server

示例：

```
switch: ping 192.0.2.15

ping 192.0.2.1 with 32 bytes of data...

Host 192.0.2.1 is alive.

switch:
```

步骤 7 请确认您的恢复分区（sda9 :)中是否有恢复镜像。

在使用紧急安装特性进行恢复时需要该恢复镜像。

示例：

```
switch: dir sda9:

Directory of sda9:/

     2 drwx 1024 .

     2 drwx 1024 ..

    11 -rw- 18923068 c3850-recovery.bin

36939776 bytes available (20830208 bytes used)

switch:
```

步骤 8 在引导程序（ROMMON）提示符处，启动紧急安装特性，帮助您恢复交换机上的软件镜像。

警告：紧急安装命令将擦除整个引导的闪存！

示例：

```
Switch#

emergency-install

tftp://192.0.2.47/cat3k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin
```

```
The bootflash will be erased during install operation, continue (y/n)?y
Starting emergency recovery
(tftp://192.0.2.47/cat3k/cat3k_caa-universalk9.SPA.03.02.00.SE.150-1.EX.bin)...
Reading full image into memory......................done
Nova Bundle Image
--------------------------------------
Kernel Address : 0x6042e5cc
Kernel Size : 0x318261/3244641
Initramfs Address : 0x60746830
Initramfs Size : 0xdb0fb9/14356409
Compression Format: .mzip
Bootable image at @ ram:0x6042e5cc
Bootable image segment 0 address range [0x81100000, 0x81b80000] is in range [0x80180000, 0x90000000].
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
File "sda9:c3850-recovery.bin" uncompressed and installed, entry point: 0x811060f0
Loading Linux kernel with entry point 0x811060f0 ...
Bootloader: Done loading app on core_mask: 0xf
### Launching Linux Kernel (flags = 0x5)
Initiating Emergency Installation of bundle
tftp://192.0.2.47/cat3k/cat3k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin
Downloading bundle
tftp://192.0.2.47/cat3k/cat3k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin...
Validating bundle
tftp://192.0.2.47/cat3k/cat3k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin...
Installing bundle
tftp://192.0.2.47/cat3k/cat3k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin...
Verifying bundle
tftp://192.0.2.47/cat3k/cat3k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin...
Package cat3k_caa-base..pkg is Digitally Signed
Package cat3k_caa-drivers.SPA.03.02.00.SE.pkg is Digitally Signed
Package cat3k_caa-infra.SPA.03.02.00.SE.pkg is Digitally Signed
Package cat3k_caa-IOSd-universalk9.SPA.03.02.00.SE.pkg is Digitally Signed
Package cat3k_caa-platform.SPA.03.02.00.SE.pkg is Digitally Signed
Package cat3k_caa-wcm.SPA.03.02.00.SE.pkg is Digitally Signed
Preparing flash...
Syncing device...
Emergency Install successful... Rebooting

Restarting system.
Booting...(use DDR clock 667 MHz)Initializing and Testing RAM
        +++@@@@####...++@@++@@++@@++@
```

恢复丢失或忘记的密码

在交换机通电启动期间，设备的默认配置允许那些对设备具有物理访问权限的终端用户通过中断引导程序并输入新密码而恢复使用。这些恢复过程要求用户具有对设备的物理访问权限。

注释： 在这些设备上，系统管理员可以禁用某些特性的一些功能，只有当终端用户同意设备恢复出厂设置，终端用户才能为设备重设密码。如果终端用户想在密码恢复功能被禁用的时候重设密码，系统会发出一个状态消息来提醒用户。

总步骤

1. 将终端或 PC 连接到交换机。
2. 将仿真软件上的线路速度设置为 9600 波特。
3. 关闭独立交换机或整个交换机堆栈。
4. 将电源线重新连接到交换机或活跃交换机上。并在 15 秒内按下 **Mode** 按钮，同时系统 LED 仍然闪烁绿色。继续按下 **Mode** 按钮，直到所有系统 LED 指示灯亮起并保持不变；然后释放 **Mode** 按钮。
5. 在恢复密码后，重新加载交换机或活跃交换机。
6. 打开堆栈中其余交换机的电源。

具体步骤

步骤 1 将终端或 PC 连接到交换机。

- 将终端或 PC 与终端仿真软件连接到交换机控制台端口。如果用户要为一个交换机堆栈恢复密码，请连接到活跃交换机的控制台端口。
- 将 PC 连接到以太网管理端口。如果用户要为一个交换机堆栈恢复密码，请连接到堆栈成员的以太网管理端口。

步骤 2 将仿真软件上的线路速度设置为 9600 波特。

步骤 3 关闭独立交换机或整个交换机堆栈。

步骤 4 将电源线重新连接到交换机或活跃交换机上。并在 15 秒内按下 **Mode** 按钮，同时系统 LED 仍然闪烁绿色。继续按下 **Mode** 按钮，直到所有系统 LED 指示灯亮起并保持不变；然后释放 **Mode** 按钮。

```
Switch:

Xmodem file system is available.

Base ethernet MAC Address: 20:37:06:4d:e9:80

Verifying bootloader digital signature.

The system has been interrupted prior to loading the operating

system software, console will be reset to 9600 baud rate.
```

请继续执行*启用密码恢复的过程*部分，然后按照步骤操作。

步骤 5 在恢复密码后，重新加载交换机或活跃交换机。

在交换机上：

```
Switch>reload

Proceed with reload? [confirm] y
```

在活跃交换机上：

```
Switch>reload slot <stack-active-member-number>

Proceed with reload? [confirm] y
```

步骤 6 打开堆栈中其余交换机的电源。

启用密码恢复时的过程

如果启用了密码恢复机制，则会显示该信息：

步骤 1 初始化闪存文件系统。

```
Device: flash_init
```

步骤 2 使用以下命令忽略启动配置：

```
Device: SWITCH_IGNORE_STARTUP_CFG=1
```

步骤 3 使用闪存中的 *packages.conf* 文件启动交换机。

```
Device: boot flash:packages.conf
```

步骤 4 通过回答 **NO** 终止初始配置对话。

```
Would you like to enter the initial configuration dialog? [yes/no]: No
```

步骤 5 在交换机提示符下，进入特权 EXEC 模式。

```
Device>enable

Switch#
```

步骤 6 将启动配置复制到正在运行的配置中。

```
Device# copy startup-config running-config Destination filename [running-config]?
```

确认提示符下按回车键。现在配置文件被重新加载，用户可以修改密码。

步骤 7 输入全局配置模式并修改 **enable** 密码。

```
Device# configure terminal

Device(config)#
```

步骤 8 将正在运行的配置写入启动配置文件中。

```
Device# copy running-config startup-config
```

步骤 9 确认已启用手动引导模式。

```
Device# show boot

BOOT variable = flash:packages.conf;

Manual Boot = yes

Enable Break = yes
```

步骤 10 重新加载交换机。

```
Device# reload
```

步骤 11 将 Bootloader 参数（之前在步骤 2 和 3 中更改的）返回到其原始值。

```
Device: switch: SWITCH_IGNORE_STARTUP_CFG=0
```

步骤 12 使用闪存中的 *packages.conf* 文件启动交换机。

```
Device: boot flash:packages.conf
```

步骤 13 设备启动后，在设备上禁用手动引导。

```
Device(config)# no boot manual
```

禁用密码恢复时的过程

如果禁用了密码恢复机制，则会显示该信息：

```
The password-recovery mechanism has been triggered, but
is currently disabled. Access to the boot loader prompt
through the password-recovery mechanism is disallowed at
this point. However, if you agree to let the system be
```

reset back to the default system configuration, access

to the boot loader prompt can still be allowed.

Would you like to reset the system back to the default configuration (y/n)?

注意： 将交换机返回默认配置会导致丢失所有当前配置。建议用户联系系统管理员确认是否具有备用设备和 VLAN 配置文件。

- 如果输入 **n**（否），将正常执行引导进程，就像没有按 **Mode** 按钮；用户无法进入引导加载提示符，也无法输入新密码。用户将看到以下信息：

  Press Enter to continue.....

- 如果输入 **y**（是），将删除闪存中的配置文件和 VLAN 数据库文件。
  加载默认配置时，用户可以重置密码。

步骤 1 选择继续执行密码恢复并删除当前配置：

Would you like to reset the system back to the default configuration (y/n)? Y

步骤 2 显示闪存内容：

Device: dir flash:

设备文件系统显示：

Directory of flash:/

.

.

.i'

15494 drwx 4096 Jan 1 2000 00:20:20 +00:00 kirch

15508 -rw- 258065648 Sep 4 2013 14:19:03 +00:00

cat3k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin

162196684

步骤 3 启动系统：

Device: boot

提示用户开始设置程序。如要继续执行密码恢复，请在提示符后输入 **N**：

Continue with the configuration dialog? [yes/no]: N

步骤 4 在设备提示符下，进入特权 EXEC 模式：

Device>enable

步骤 5 进入全局配置模式：

Device# configure terminal

步骤 6 更改密码：

Device(config)# enable secret password

密码可以是 1-25 位数字和字母组合的字符，可以以数字开头，区分大小写，可以使用空格但会忽略空格。

步骤 7 返回特权 EXEC 模式：

Device(config)# exit

Device#

注释： 继续执行步骤 9 之前，请开启所有连接的堆栈成员，直到它们完全初始化。

步骤 8 将正在运行的配置写入启动配置文件中：

Device# copy running-config startup-config

现在，启动配置中包含新的密码。

步骤 9 现在，用户必须重新配置交换机。如果系统管理员拥有可用的备份设备和 VLAN 配置文件，用户应该使用。

防止交换机堆栈问题

为了防止交换机堆栈问题，用户应该按以下操作执行：

- 请确保向交换机堆栈添加或从中移除的设备已关闭电源。交换机堆栈中的所有供电注意事项，请参阅硬件安装指南中的“交换机安装”一章。
- 按下堆栈成员上的 **Mode** 按钮，直到堆栈模式的 LED 亮起。设备上最后两个端口的 LED 应为绿色。根据设备型号，最后两个端口是 10/100/1000 端口或小型可插拔（Small Form-factor Pluggable，SFP）模块。如果最后两个端口的 LED 有一个或都不是绿色，说明堆栈操作没有占用全部带宽。
- 建议在管理交换机堆栈时仅使用一个 CLI 会话。在活跃交换机上使用多个 CLI 会话时要小心。在一个会话中输入的命令不会显示在其他会话中。因此，用户可能无法分别所输入命令的对话。
- 根据设备在堆栈中的位置手动分配堆栈成员编号，可以更方便地对交换机堆栈进行远程故障排除。但是，在以后用户要添加、移除或重新排列设备时，用户要记住已经为设备手动分配了编号。使用 **switch** current-stack-member-number **renumber** new-stack-member-number 全局配置命令来手动分配堆栈成员编号。

如果使用相同型号的交换机替换堆栈成员，假设新设备与被替换设备使用相同的成员编号，那么新设备使用与被替换设备完全相同的配置进行工作。

移除已打开电源的堆栈成员会导致交换机堆栈划分（分割）为两个或多个交换机堆栈，每个交换机堆栈具有相同的配置。如果用户希望交换机堆栈保持分离，请更改新创建的交换机堆栈的 IP 地址。要从交换机堆栈分区恢复，请按照以下步骤操作：

1. 将新创建的交换机堆栈的电源关闭。
2. 通过 StackWise Plus 端口将它们重新连接到原始交换机堆栈。
3. 将设备电源打开。

有关可用于监控交换机堆栈及其成员的命令，请参阅*显示交换机堆栈信息*部分。

防止自动协商不匹配

IEEE802.3ab 自动协商协议管理设备速度（10Mb/s、100Mb/s 和 1000Mb/s，不包括 SFP 模块端口）和双工（半/全双工）的设置。有些情况下，此协议未能匹配这些设置，这降低了性能。在以下情况中会发生不匹配：

- 手动设置的速度或双工参数不同于所连接端口上手动设置的速度或双工参数。
- 某端口设置为自动协商，但其连接的端口设置为全双工无自动协商。

为了最大限度地提高设备性能并保证链路通信，请在更改双工和速度设置时遵循其中一个指导方案：

- 让两个端口自动协商速度和双工信息。
- 手动设置连接两端端口的速度和双工参数。

注释： 如果远程设备未自动协商，请将两个端口上的双工设置设为匹配。即使连接的端口未自动协商，速度参数也可以自行调整。

SFP 模块安全及标识的故障排除

Inspur 小型可插拔（Small Form-factor Pluggable，SFP）模块装有 EEPROM，其中包含模块序列号、供应商名称和 ID、唯一安全代码和循环冗余校验（Cyclic redundancy check，CRC）。当 SFP 模块插入设备时，设备软件会读取 EEPROM 来确认序列号、供应商名称和供应商 ID，并重新计算安全代码和 CRC。如果序列号、供应商名称或供应商 ID、安全代码或 CRC 无效，软件将生成安全错误消息并将接口置于错误禁用状态。

注释： 安全错误消息提及到 GBIC_SECURITY 功能。设备支持 SFP 模块，不支持 GBIC 模块。从字面上看安全错误消息指的是 GBIC 接口和模块，但实际上是指 SFP 模块和模块接口。

如果用户正在使用非 Inspur 的 SFP 模块，请从设备中移除该 SFP 模块，并用 Inspur 模块替换该模块。在插入 Inspur SFP 模块后，请使用 **errdisable recovery cause gbic-invalid** 全局配置命令确认端口状态，并输入从错误禁用状态恢复的时间间隔。在经过此时间间隔后，接口将从错误禁用状态中恢复，并重新运行。有关 **errdisable recovery** 命令的更多信息，请参阅此版本的命令参考。

如果模块被识别为 Inspur SFP 模块，但系统无法读取供应商数据信息以确认其准确性，则会生成 SFP 模块错误消息。在这种情况下，用户应移除并重新插入 SFP 模块。如果仍然出现故障，则说明 SFP 模块本身可能有故障。

监控 SFP 模块状态

用户可以使用 **show interface transceiver** 特权 EXEC 命令检查 SFP 模块的物理状态或运行状态。该命令显示了运行状态，例如特定接口上 SFP 模块的温度和电流以及警报状态。用户还可以使用该命令检查 SFP 模块上的速度设置和双工设置。有关更多信息，请参阅此版本命令参考中的 **show interfaces transceiver** 命令。

运行 Ping

如果用户试图 ping 不同 IP 子网中的主机，那么必须为网络定义静态路由，或者配置 IP 路由以帮助数据包在这些子网之间选路。

在默认情况下，所有设备上的 IP 路由处于禁用状态。

注释： 尽管 **ping** 命令可以使用其他协议的关键字，但此版本不支持这些关键字。

使用此命令从设备上 ping 网络中的其他设备：

命令	目的
ping ip*host* l*address* Device# ping 172.20.52.3	通过 IP，或者通过使用主机名或网络地址来 ping 远程主机。

温度监控

设备会监控温度，并使用温度信息来控制风扇。

使用 **show env temperature status** 特权 EXEC 命令显示温度值、状态和门限值。温度值是设

备内部的温度（而不是外部温度）。用户只能使用 **system env temperature threshold yellow***value* 全局配置命令配置黄色门限的基准（摄氏度），以便设置黄色和红色门限值之间的差值。用户不能配置绿色或红色门限值。有关更多信息，请参阅此版本的命令参考。

物理路径监控

用户可以使用以下特权 EXEC 命令来监控数据包从源设备到目的设备的物理路径：

表格 204：物理路径监控

命令	目的
tracetroute mac [**interface***interface-id*] {*source-mac-address*} [**interface***interface-id*] {*destination-mac-address*} [**vlan***vlan-id*] [**detail**]	显示从指定的源 MAC 地址到指定的目的 MAC 地址的报文所经过的二层路径。
tracetroute mac ip{*source-ip-address*| *source-hostname*}{*destination-ip-address*| *destination-hostname*} [**detail**]	显示从指定的源 IP 地址或主机名到指定的目的 IP 地址或主机名的数据包所采用的第 2 层路径。

执行 IP Traceroute

注释： 尽管 **traceroute** 特权 EXEC 命令可以使用其他协议的关键字，但此版本不支持这些关键字。

命令	目的
traceroute ip*host* Device# traceroute ip 192.51.100.1	跟踪数据包通过网络时所走路径。

运行 TDR 及显示结果

要运行 TDR，请输入 **test cable-diagnostics tdr interface***interface-id* 特权 EXEC 命令。

要显示结果，请输入 **show cable-diagnostics tdr interface***interface-id* 特权 EXEC 命令。

重定向调试和错误消息输出

默认情况下，网络服务器将 **debug** 命令和系统错误消息的输出发送到控制台。如果使用此默认值，用户可以使用虚拟终端连接来监视调试的输出，而不必连接到控制台端口或以太网管理端口。

目的地包括控制台、虚拟终端、内部缓冲区和运行系统日志服务器的 UNIX 主机。系统日志格式与 4.3 版 Berkeley 标准分发（Berkeley Standard Distribution，BSD）UNIX 及其衍生版本兼容。

注释： 请注意，用户使用的调试目标会影响系统开销。当您将消息记录到控制台时，会产生非常高的开销。当您将消息记录到虚拟终端时，会产生较少的开销。将消息记录到系统日志服务器可以产生更少的开销，如果是记录到内部缓冲器产生的开销将最小。

了解更多有关系统消息记录的详细信息，请参阅*配置系统消息记录*。

使用 show platform forward 命令

如果通过系统将数据包发送到接口，**show platform forward** 特权 EXEC 命令的输出可以提供一些有关转发结果的有用信息。根据输入的有关数据包的参数，输出可提供用于计算转发目的地、位图和出口信息的查找表结果和端口映射。

命令输出中的大多数信息主要给技术支持人员使用，他们可以访问有关设备专用集成电路（Application-Specific Integrated Circuits，ASIC）的详细信息。然而，数据包转发信息也有助于故障排除。

使用 show debug 命令

在特权 EXEC 模式下输入 **show debug** 命令。此命令显示交换机上所有可用的调试选项。

要查看所有条件调试选项，请运行命令 **show debug condition**。可以通过选择条件标识符 *<1-1000>*或 *all* 条件来列出这些命令。

要禁用调试，请使用 **no debug all** 命令。

注意： 由于调试的输出在 CPU 进程中被分配了较高优先级，它可能导致系统不可用。因此，仅在排除特定问题或与 Inspur 技术支持人员进行故障排除会话期间使用 **debug** 命令。最好在网络流量较低和用户较少时使用调 **debug** 命令。在此期间的调试可减少增加的 **debug** 命令处理开销影响系统使用的可能性。

有关详细信息，请参阅 *Inspur INOS 配置基础命令参考*，*Inspur INOS（Inspur 3850 交换机）*。

配置 OBFL

注意： 建议用户不要禁用 OBFL，并且不要删除存储在闪存中的数据。

- 要启用 OBFL，请使用 **hw-switch switch** [*switch-number*] **logging onboard** [**message level***level*]全局配置命令。在交换机上，*switch-number* 的范围在 1 到 9 之间。使用 **message level***level* 参数可以指定交换机生成并存储在闪存中的有关硬件信息的重要性。
- 要将 OBFL 数据复制到本地网络或指定文件系统，请使用 **copy onboard switch** *switch-number***url***url-destination* 特权 EXEC 命令。
- 要禁用 OBFL，请使用 **no hw-switch switch** [*switch-number*] **logging onboard** [**message level**] 全局配置命令。
- 要清除闪存中除了正常运行时间和 CLI 命令信息之外的所有 OBFL 数据，请使用 **clear onboard switch***switch-number* 特权 EXEC 命令。
- 在交换机堆栈中，用户可以使用 **hw-switch switch** [*switch-number*] **logging onboard** [**message level***level*]全局配置命令在独立交换机或所有堆栈成员上启用 OBFL。
- 用户可以从活跃交换机上启用或禁用成员交换机上的 OBFL。

有关本节中的命令的更多信息，请参阅此版本的命令参考。

验证软件配置的故障排除

显示 OBFL 信息

表格 205：显示 OBFL 信息的命令

命令	目的
show onboard switch*switch-number***clilog** Device# show onboard switch 1 clilog	显示独立交换机或指定堆栈成员上输入的 OBFL CLI 命令。
show onboard switch*switch-number***environment** Device# show onboard switch 1 environment	显示独立交换机或指定堆栈成员及所有连接的 FRU 设备上的 UDI 信息，包括：PID、VID 和序列号。
show onboard switch*switch-number***message** Device# show onboard switch 1 message	显示独立交换机或指定堆栈成员生成的有关硬件的消息。
show onboard switch*switch-number***counter** Device# show onboard switch 1 counter	显示独立交换机或指定堆栈成员的计数器信息。
show onboard switch*switch-number***temperature** Device# show onboard switch 1 temperature	显示独立交换机或指定交换机堆栈成员的温度。
show onboard switch *switch-number***uptime** Device# show onboard switch 1 uptime	显示独立交换机或指定堆栈成员启动的时间，重新启动的原因，以及自上次重新启动以来运行的时间。
show onboard switch *switch-number***voltage** Device# show onboard switch 1 voltage	显示独立交换机或指定堆栈成员的系统电压。
show onboard switch *switch-number***status** Device# show onboard switch 1 status	显示独立交换机或指定堆栈成员的状态。

示例：确认高 CPU 使用率的问题及原因

要确定高 CPU 使用率是否有问题，请输入 **show processes cpu sorted** 特权 EXEC 命令。请注意输出示例第一行中带下划线的信息。

```
Device# show processes cpu sorted
CPU utilization for five seconds: 8%/0%; one minute: 7%; five minutes: 8%
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
309 42289103 752750 56180 1.75% 1.20% 1.22% 0 RIP Timers
140 8820183 4942081 1784 0.63% 0.37% 0.30% 0 HRPC qos request
100 3427318 16150534 212 0.47% 0.14% 0.11% 0 HRPC pm-counters
192 3093252 14081112 219 0.31% 0.14% 0.11% 0 Spanning Tree
143 8 37 216 0.15% 0.01% 0.00% 0 Exec
...
<输出已删节>
```

示例显示了正常的 CPU 利用率。输出显示最近 5 秒的使用率是 8%/0%，具有以下含义：

- CPU 总的利用率为 8%，包括运行 Inspur INOS 进程和处理中断所花费的时间。
- 处理中断所花费的时间为 0%。

表格 206：排除 CPU 利用率问题

问题类型	原因	修正措施
中断百分比值几乎与 CPU 总	CPU 从网络中接收到过多数	确定网络数据包的来源。停止

利用率值一样高。	据包。	网络流量，或更改交换机配置。 请参阅“分析网络流量”部分。
在中断用时最少的情况下，CPU 总利用率大于 50%。	一个或多个 Inspur INOS 进程消耗过多的 CPU 时间。这通常由激活进程的事件触发。	识别异常事件，并解决根本原因。请参阅“调试活跃进程”部分。

软件配置故障排除场景

以太网供电故障排除场景

表格 207：以太网供电故障排除场景

症状或问题	可能的原因和解决方法
只有一个端口没有 PoE。 故障仅出现在一个交换机端口上。PoE 和非 PoE 设备在此端口不工作，但在其他端口工作。	请确认用电设备是否在另一个 PoE 端口上工作。 使用 **show run** 或 **show interface status** 用户 EXEC 命令确认端口未关闭或处于错误禁用状态。 **注意：** 当端口关闭时，大多数交换机会关闭端口电源，即使在 IEEE 规范中也仅将其列为可选项。 确认用电设备到交换机端口的以太网电缆是否正常：将已知良好的非 PoE 以太网设备连接到以太网电缆，并确保用电设备可与另一主机建立链路并交换流量。 确认从交换机前面板到用电设备的电缆总长度不超过 100 米。 断开以太网电缆与交换机端口的连接。使用短以太网电缆将已知良好的以太网设备直接连接到交换机前面板（不是接线板）上的此端口。确认它可以与另一台主机建立以太网链路并交换流量，或 ping 端口 VLAN SVI。接下来，将用电设备连接到此端口，并确认是否接通电源。如果用电设备使用跳线连接到交换机端口时未接通电源，请将连接的用电设备总数与交换机功率预算（PoE 可用）进行比较。使用 **show inline power** 命令确认可用的电量。
在所有端口或一组端口上没有 PoE。 故障发生在所有交换机端口上。 未通电的以太网设备不能在任何端口上建立以太网链路，并且 PoE 设备也无法通电。	如果电源出现连续的、间歇的或重复的警报，若电源是一个现场可更换单元，请更换电源。否则，请更换交换机。 如果问题出现在一组连续的端口上，但并非所有端口，说明电源可能没有问题，问题可能与交换机中的 PoE 调节器有关。 使用 **show log** 特权 EXEC 命令查看之前报告的 PoE 条件或状态更改的警报或系统消息。 如果没有警报，请使用 **show interface status** 命令确认端口是否关闭或处于错误禁用状态。如果端口处于错误禁

	用状态，请使用 **shut** 和 **no shut** 接口配置命令重新启用端口。 使用 **show env power** 和 **show power inline** 特权 EXEC 命令可查看 PoE 状态和功率预算（PoE 可用）。 查看运行的配置以确认未在端口上配置 **power inline never**。 将未通电的以太网设备直接连接到交换机端口。仅使用跳线，不使用已有的配线电缆。输入 **shut** 和 **no shut** 接口配置命令，然后确认以太网链路是否已建立。如果此连接良好，请使用短跳线将用电设备连接到此端口，并确认其通电。如果设备通电，请确认是否所有的中间接线板已正确连接。 在交换机选择一个端口，断开其余端口的以太网电缆。使用短跳线，将用电设备连接到该 PoE 端口。确认用电设备所用功率小于该交换机端口可提供的功率。 使用 **show power inline** 特权 EXEC 命令确认用电设备可以在端口未关闭时接收电量。或者，通过观察用电设备确认其处于通电状态。 如果用电设备在当只有一个用电设备连接到交换机时才能处于通电状态，请在其余端口上输入 **shut** 和 **no shut** 接口配置命令，然后将以太网电缆依次重新连接到交换机 PoE 端口。使用 **show interface status** 和 **show power inline** 特权 EXEC 命令来监管内联电源统计信息和端口状态。 如果仍然没有任何端口能 PoE，则电源中 PoE 部分的保险丝可能会断开。这通常会产生警报。请再次检查日志，了解更早的系统消息报告的警报。
Inspur IP 电话断开连接或重置。 正常工作后，Inspur 电话会间歇性地重新加载或断开与 PoE 的连接。	确认交换机到用电设备的所有电力连接。任何不可靠的连接可能会导致电力中断和用电设备无法稳定运行，例如不稳定的用电设备会断开连接和重新加载。 确认从交换机端口到用电设备的电缆长度不超过 100 米。 请注意在用电设备出现无法连接的情况时，交换机所处位置的电气环境以及用电设备是否发生变化。 请注意在用电设备出现无法连接的情况时，是否有任何错误消息出现。使用 **show log** 特权 EXEC 命令查看错误消息。 确认 IP 电话在重新加载之前不会立即失去对呼叫管理器的访问（这可能是网络问题，而非 PoE 问题）。 使用非 PoE 设备更换用电设备，并确认设备是否正常工作。如果非 PoE 设备有链路问题或高误码率，则问题可能出在交换机端口和用电设备之间的不可靠电缆连接。
非 Inspur 的用电设备不能在 Inspur PoE 交换机上工作。	请使用 **show power inline** 命令确认交换机的功率预算（PoE 可用）在用电设备连接之前或之后是否会耗尽。

当非 Inspur 的用电设备连接到 Inspur PoE 交换机上，要么无法连通电源，要么开机后马上关机。非 PoE 设备正常工作。	在这类用电设备连接之前，请确认有足够的电源可用。使用 **show interface status** 命令确认交换机是否检测到连接的用电设备。 使用 **show log** 命令查看端口上电流情况的系统消息。请准确地识别症状：是否初时用电设备通电，而后断电？如果是，则问题可能是初始涌入（或流入）电流超过端口的电流门限限制。

软件故障排除的配置示例

示例：Ping 某 IP 主机

此示例说明如何 ping 一个 IP 主机：

```
Device# ping 172.20.52.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echoes to 172.20.52.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Device#
```

表格 208：Ping 输出显示的字符

字符	描述
!	每个感叹号表示接收到一个应答。
.	每个句号表示等待一个应答时，网络服务超时。
U	收到一个目的地不可达的 PDU。
C	收到一个经历过拥塞的数据包。
I	用户中断测试。
?	未知数据包类型。
&	数据包生存期已过。

若要结束此次 ping 会话，请输入退出序列（默认情况下为 **Ctrl- ^ X**）。同时按下并释放 **Ctrl**，**Shift** 和 **6** 键，然后按 **X** 键。

示例：对 IP 主机执行 Traceroute

此示例说明如何对 IP 主机执行 **traceroute**：

```
Device# traceroute ip 192.0.2.10
Type escape sequence to abort.
Tracing the route to 192.0.2.10
1 192.0.2.1 0 msec 0 msec 4 msec
2 192.0.2.203 12 msec 8 msec 0 msec
3 192.0.2.100 4 msec 0 msec 0 msec
4 192.0.2.10 0 msec 4 msec 0 msec
```

结果显示了跳数、路由器的 IP 地址以及发送的三个探针的各自往返时间（以毫秒为单位）。

表格 209：Traceroute 输出显示的字符

字符	描述
*	探针超时。
?	未知数据包类型。
A	管理不可达。通常，此输出说明有访问列表限制了流量。
H	主机不可达。
N	网络不可达。
P	协议不可达。
Q	源抑制。
U	端口不可达。

要结束跟踪进程，请输入退出序列（默认情况下为 **Ctrl- ^ X**）。同时按下并释放 **Ctrl**，**Shift** 和 **6** 键，然后按 **X** 键。

示例：启用所有系统诊断

注意： 由于调试输出的优先级高于其他网络流量，并且 **debug all** 特权 EXEC 命令比其他 **debug** 命令生成的输出多，所以它可能导致交换机性能严重降低甚至不可用。所以无论在哪种场景下，最好使用更具体的 **debug** 命令。

这个命令会禁用所有系统诊断：

```
Device# debug all
```

no debug all 特权 EXEC 命令会禁用所有诊断输出。使用 **no debug all** 命令便于用户确保将所有启用的 debug 命令停止。

其他参考资料

相关文档

相关主题	文档标题
系统管理命令	系统管理命令参考文献（Inspur 6650 交换机）
独立平台命令参考	配置基础命令参考文献，Inspur INOS （Inspur 6650 交换机）
独立平台配置信息	配置基础配置指南，Inspur INOS（Inspur 6650 交换机）

标准和 RFC

标准/RFC	标题
无	—

技术助手

描述	链接
Inspur 支持网站提供了广泛的在线资源，包括用于故障排除以及解决 Inspur 产品问题和技术问题的文档和工具。	http://www/icntnetworks.com/icnt

要获得有关产品的安全和技术信息，用户可以订阅不同的服务，例如产品警报工具（从现场记录获取），Inspur 技术服务实时通讯和简单讯息聚合订阅（RSS）。 要获取 Inspur 支持网站上的大多数工具，需要有 icntnetworks.com 的用户 ID 和密码。	

软件配置故障排除的历史特性与信息

版本	修订
Inspur INOS 11.3.1	引入了这一特性。

第 16 部分 VLAN

配置 VTP

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

配置 VTP 的前提条件

用户在创建 VLAN 之前，必须确定是否在网络中使用 VTP（VLAN Trunking Protocol，VTP）。使用 VTP，用户可以集中在一个或多个设备上进行配置变更，这些变更的配置可以被自动通告给网络中的其他设备。不使用 VTP，用户无法向其他设备发送关于 VLAN 的信息。
VTP 的工作环境要求更新要在一个单独的设备上进行，更新的内容通过 VTP 发送到域中的其他设备上。如果有很多对 VLAN 数据库的更新同时发生在一个域中的多台设备上，这会导致 VLAN 数据库不一致。
设备总共支持 4094 个 VLAN。但是，配置的特性数量会影响设备硬件的使用。如果 VTP 通告设备有一个新 VLAN，且此时设备已经在使用最大可用硬件资源，那么设备会发送一条消息指出没有足够的硬件资源可用，然后关闭 VLAN。**show vlan** 用户 EXEC 命令的输出显示 VLAN 处于挂起状态。
由于 VTP 通告通过中继端口发送和接收，所以用户必须确保在设备或设备堆栈上至少配置一个中继端口，并且此中继端口必须与另一个设备的中继端口相连通。否则，设备无法接收

任何 VTP 通告。

配置 VTP 的限制条件

配置 VTP 的限制条件如下所示：

- 用户不能在设备堆栈中混用 Inspur 3850 和 Inspur 6650 交换机。

注意： 在将 VTP 客户端添加到 VTP 域之前，请先确认其 VTP 配置修订号低于 VTP 域中其他设备的配置修订号。VTP 域中的设备会使用 VTP 配置修订号最高设备的 VLAN 配置。如果用户添加的设备修订号高于 VTP 域中的修订号，那么该设备可以清除 VTP 服务器和 VTP 域中所有的 VLAN 信息。

关于 VTP 的信息

VTP

VTP 是一种二层消息传递协议，在全网的基础上管理 VLAN 的添加、删除和重命名，从而维护 VLAN 配置的一致性。VTP 能减少由于错误配置或配置不一致导致的一系列问题，例如重复的 VLAN 名称，错误的 VLAN 类型规范和安全违规。

VTP 的功能支持跨堆栈工作，并且堆栈中的所有设备都维护从活跃设备继承的相同 VLAN 和 VTP 配置。当设备通过 VTP 消息得知有新的 VLAN 出现，或当用户配置了新的 VLAN，该新 VLAN 的信息将传播到堆栈中的其他所有交换机上。

当一台设备加入堆栈或者多个堆栈进行合并的时候，新设备会从活跃设备获取到 VTP 信息。

VTP 域

VTP 域（也成为 VLAN 管理域），由一个或一个以上共享 VTP 域名的具有相同管理责任的、相互连接的设备或设备堆栈组成。一个设备只能在一个 VTP 域中工作。用户可以在域中做全局 VLAN 配置的更改。

默认情况下，设备会处于 VTP 非管理域（no-management-domain）状态，直到交换机通过中继链路（携带很多 VLAN 流量的链路）接收到关于域的通告，或者用户在交换机上配置了域名。只有当管理域的域名被确定或者被习得，用户才能在 VTP 服务器上创建或修改 VLAN，且更改 VLAN 的消息才能在网络上传播。

如果设备通过中继链路收到了 VTP 通告，该设备会继承管理域的域名和 VTP 配置的修订号。然后设备会忽略来自其他域名或修订号更小的通告。

当用户在 VTP 服务器上对 VLAN 配置进行了更改，这些变更会通告到 VTP 域中的所有设备。VTP 通告会通过所有的 IEEE 中继连接进行传播，包括 IEEE802.1Q。VTP 能跨越多种 LAN 类型进行 VLAN 的动态映射，每种 VLAN 都有唯一的名称和与之关联的内部索引。这种映射为管理员减少了大量的设备管理任务。

如果交换机配置为透明模式，用户可以创建或修改 VLAN，但所做的修改只会影响单个交换机，不会被发送到域中的其他设备上。但是，设备在这种模式下所做的配置更改会存到设备正在运行的配置中，并可保存到设备的启动配置文件中。

VTP 模式

表格 211：VTP 模式

VTP 模式	描述
VTP 服务器	在 VTP 服务器模式下，用户可以创建、修改和删除 VLAN，还可以为整个 VTP 域确定其他配置参数（例如 VTP 版本）。VTP 服务器向位于同一个 VTP 域中的其他设备通告 VLAN 配置，并且通过在中继链路上接收通告消息来实现与其他设备进行 VLAN 配置的同步。 VTP 服务器是默认模式。 在 VTP 服务器模式下，VLAN 的配置存储与 NVRAM。如果设备在将配置写入 NVRAM 时检测到故障，VTP 模式会自动从服务器模式更改为客户端模式。如果发生这种情况，除非 NVRAM 运行，否则设备无法返回 VTP 服务器模式
VTP 客户端	VTP 客户端与服务器模式的工作方式相似，通过中继传送、接收 VTP 更新，但是用户不能在 VTP 客户端上创建、修改和删除 VLAN。VLAN 在域中其他处于服务器模式的设备上进行配置。 在 VTP 客户端模式的 VTP 版本 1 和版本 2 中，NVRAM 并不保存 VLAN 的配置信息。在 VTP 版本 3 中，在客户端模式下 NVRAM 会保存 VLAN 的配置信息
VTP 透明	处于透明模式的设备不会加入 VTP 域中。一个 VTP 透明设备不会通告自己的 VLAN 配置，也不会根据接收到的通告信息同步自己的 VLAN 配置。但是在 VTP 版本 2 或版本 3 中，透明设备会从中继接口转发从其他设备接收到的 VTP 通告。用户可以在处于 VTP 透明模式的设备上创建、修改和删除 VLAN 配置。当设备处于 VTP 透明模式，VTP 和 VLAN 配置存储于 NVRAM 中，不会被通告到其他设备上。在这种模式下，VTP 模式和域名存储于设备正在运行的配置中，并且用户可以使用 **copy running-config startup-config** 特权 EXEC 命令将这些信息存储到设备的启动配置文件中。 在一个设备堆栈中，对于堆栈中的所有设备，正在运行的配置和已经保存的配置没有区别
VTP 关闭	VTP 关闭模式与 VTP 透明模式的交换机工作方式相同，但它不会通过中继转发 VTP 通告

VTP 通告

在 VTP 域中的每个设备都会通过中继端口向保留的多播地址发送周期性的全局配置通告。相邻设备会接收通告并更新他们的 VTP 和 VLAN 配置。

VTP 通告发送以下全域信息：

- VTP 域名
- VTP 配置修订号
- 更新者身份和更新时间戳
- VLAN 配置的 MD5 摘要散列码，包括每个 VLAN 的最大传输单元（Maximum Transmission Unit，MTU）大小
- 帧格式

VTP 通告为每个配置好的 VLAN 发送以下 VLAN 信息：

- VLAN ID（包括 IEEE802.1Q）

- VLAN 名字
- VLAN 类型
- VLAN 状态
- 针对 VLAN 类型的附加 VLAN 配置信息

在 VTP 版本 3 中，VTP 通告也包括主服务器 ID、一个实例编号和一个开始索引。

VTP 版本 2

如果用户在网络中使用 VTP，必须决定使用 VTP 的哪个版本。VTP 默认运行版本 1。

与版本 1 相比，VTP 版本 2 提供了下列额外功能：

- 支持令牌环——VTP 版本 2 支持令牌环网桥中转功能（Token Ring Bridge Relay Function，TrBRF）和令牌环集中器中继功能（Token Ring Concentrator Relay Function，TrCRF）VLAN；
- 支持无法识别的类型长度值（Type-Length-Value，TLV）——VTP 服务器或客户端将配置变更消息传播到其他中继，即使对无法解析的 TLV 也是如此。当设备在 VTP 服务器模式下运行时，无法识别的 TLV 会保存在 NVRAM 中；
- 依赖于版本的透明模式——在 VTP 版本 1 中，VTP 透明设备会检查 VTP 消息的域名和版本，并仅在版本和域名都匹配时才能转发消息。尽管 VTP 版本 2 只支持一个域，但 VTP 版本 2 的透明设备仅在域名匹配时就能转发消息；
- 一致性检查——在 VTP 版本 2 中，只有当用户通过 CLI 或 SNMP 输入新信息的时候，才执行一致性检查（例如 VLAN 名和值）。对于从 VTP 消息和 NVRAM 中获取的信息不进行一致性检查。如果接收到的 VTP 消息的 MD5 摘要散列码是正确的，说明信息可靠。

VTP 版本 3

与版本 1 和版本 2 相比，VTP 版本 3 提供了下列额外功能：

- 认证加强——用户可以将认证配置为 **hidden** 或 **secret**。当处于 **hidden** 模式下，密码字符串中的密钥保存在 VLAN 数据库文件中，但在配置信息中不以纯文本显示。相反，与密码关联的密钥以十六进制格式保存在运行的配置中。如果用户在域中输入接管命令，则需要重新输入密码。当用户输入 **secret** 关键字时，可以直接配置密码的密钥；
- 支持扩展的 VLAN（VLAN 1006 到 4094）数据库传播——VTP 版本 1 和版本 2 只能传播 VLAN1 到 1005；

注释： VTP 修剪仍然只适用于 VLAN 1 到 1005，VLAN 1002 到 1005 仍然保留，不能修改。

- 支持域中的任意数据库——除了传播 VTP 信息，版本 3 可以传播多生成树（Multiple Spanning Tree，MST）协议数据库信息。为每个使用 VTP 的应用程序分别运行 VTP 协议的实例；
- VTP 主服务器和 VTP 辅助服务器——VTP 主服务器负责更新数据库信息，并发送满足系统中所有设备的更新。VTP 辅助服务器只能将从主服务器接收到的 VTP 配置更新备份到其 NVRAM 中。

  默认情况下，所有的设备都是辅助服务器。用户可以输入 **vtp primary** 特权 EXEC 命令指定一个主服务器。当管理员在域中发出接管消息时，主服务器的状态仅用于数据库更新。VTP 域可以在没有主服务器的环境下工作。如果设备重新加载或设备的域参数被更改，即使在设备上配置了密码，主服务器的状态也会丢失；
- 基于每个中继（每端口）VTP 打开或关闭的选项——用户可以通过输入**[no] vtp** 接口配

置命令来启用或禁用每个端口的 VTP。当用户在中继端口上禁用 VTP 时，将禁用该端口的所有 VTP 实例。用户不能将 MST 数据库的 VTP 设为*关闭*的同时将 VLAN 数据库设为*开启*。

当用户将全局的 VTP 模式设为关闭时，会应用于系统中的所有中继端口。但是，用户可以指定某 VTP 是 on 或 off。例如，用户可以为了 VLAN 数据库将设备配置成 VTP 服务器，再将 MST 数据库的 VTP *关闭*。

VTP 修剪

VTP 修剪通过将泛洪数据流限制到流量到达目的设备必须经过的中继链路上，从而提高网络可用带宽。如果没有 VTP 修剪，设备会通过某 VTP 域中所有中继链路来泛洪广播、多播和未知单播，即便接收设备可能会丢弃接收到的帧。默认情况下 VTP 修剪被禁用。

VTP 修剪会在可修剪列表中的中继端口上阻止不必要的泛洪数据流流向 VLAN。只有在可修剪列表中的 VLAN 可以被修剪。默认情况下，VLAN2 到 1001 是可修剪设备的中继端口。如果 VLAN 被配置为可修剪，泛洪将继续进行。所有 VTP 版本都支持 VTP 修剪。

VTP 修剪在交换网络中被禁用。设备 A 上的端口 1 和设备 D 上的端口 2 分配给红色 VLAN。如果广播从连接到设备 A 的主机发送出来，设备 A 会洪泛广播，并且网络中的每个设备都会接收广播，即使设备 C，E 和 F 与红色 VLAN 没有直连端口。

图 137：未启用 VTP 修剪的流量泛洪

Switch x	交换机 x
Port x	端口 x
Red VLAN	红色 VLAN

VTP 修剪在交换网络中被启用。来自设备 A 的广播流量不会被转发到设备 C，E 和 F，因为红色 VLAN 的流量已在所示链路（设备 B 上的端口 5 和设备 D 上的端口 4）上被修剪。

图 138：启用 VTP 修剪的优化泛洪流量

Switch x	交换机 x
Port x	端口 x
Red VLAN	红色 VLAN
Flooded traffic is pruned	泛洪流量被修剪

对于 VTP 版本 1 和 2，当用户在 VTP 服务器上启用修剪时，整个 VTP 域将启用修剪。在 VTP 版本 3 中，用户必须在域中的每个设备上手动启用修剪。VLAN 的可修剪或不可修剪只影响在中继上的（不是在 VTP 域中的所有设备上）那些 VLAN 的可修剪性。

VTP 修剪在启用后几秒钟内生效。 VTP 修剪不会修剪那些不可修剪的 VLAN 流量。 VLAN 1 和 VLAN 1002 到 1005 是不可修剪的;来自这些 VLAN 的流量不能修剪。扩展范围的 VLAN（高于 1005 的 VLAN ID）也是不可修剪的。

VTP 和设备堆栈

设备堆栈中所有成员使用相同的 VTP 配置。当设备堆栈处于 VTP 服务器、客户端或透明模式下，堆栈中的所有设备都携带相同 VTP 配置。

- 当设备加入堆栈后，它会从活跃交换机继承 VTP 和 VLAN 性质。
- 所有 VTP 更新都在堆栈中传送。

- 当堆栈中的设备更改 VTP 模式时，堆栈中的其他设备也会更改 VTP 模式，并且设备的 VLAN 数据库会保持一致。

VTP 版本 3 在独立设备或堆栈上具有相同功能，除非该设备堆栈是 VTP 数据库的主服务器。在这种情况下，活跃交换机的 MAC 地址将用作主服务器 ID。如果活跃设备正在重新加载或已关机，则选择一台新的活跃交换机。

- 如果不配置永久 MAC 地址，当选择新的活跃设备时，它将使用当前堆栈的 MAC 地址发送接管消息。

注释： 默认情况下，永久 MAC 地址已启用。

VTP 配置指南

VTP 配置需求

当用户需要配置 VTP 的时候，用户必须配置中继端口，以便设备可以从域中的其他设备发送和接收 VTP 通告。

VTP 设置

VTP 信息保存在 VTP VLAN 数据库中。当 VTP 模式为透明时，VTP 域名和模式也保存在设备正在运行的配置文件中，通过输入 **copy running-config startup-config** 特权 EXEC 命令，可以将其保存到设备启动配置文件中。如果要将 VTP 模式保存为透明，即使设备重置，用户也必须使用此命令。

当用户将 VTP 信息保存到设备启动配置文件中并重启设备时，设备配置选择如下：

- 如果启动配置中的 VTP 模式为透明，且 VLAN 数据库和 VLAN 数据库中的 VTP 域名与启动配置文件中的 VTP 域名匹配，则忽略（清除）VLAN 数据库，并使用启动配置文件中的 VTP 和 VLAN 配置。VLAN 数据库版本号在 VLAN 数据库中保持不变。
- 如果启动配置中的 VTP 模式或域名与 VLAN 数据库不匹配，那么 VLAN ID 1 到 1005 的域名、VTP 模式和配置将使用 VLAN 数据库中的信息。

配置 VTP 的域名

在初次配置 VTP 的时候，用户必须分配域名。用户需要为 VTP 域中的所有设备配置相同的域名。处于 VTP 透明模式的设备不与其他设备交换 VTP 信息，用户不需要为它们配置 VTP 域名。

注释： 如果 NVRAM 和 DRAM 有足够存储空间，则 VTP 域中的所有设备都应处于 VTP 服务器模式。

注意： 如果所有的设备都以 VTP 客户端模式运行，请不要配置 VTP 域。如果用户配置了 VTP 域，那么该域的 VLAN 配置将无法修改。请确保 VTP 域中至少有一台设备被配置为 VTP 服务器模式。

VTP 域的密码

用户可以为 VTP 域配置密码，但并不是必须配置。如果用户配置了域密码，那么所有设备都将使用相同的密码，并且用户需要在管理域中的每个设备上配置该密码。没有密码或密码错误的设备将拒绝 VTP 通告。

如果为域配置 VTP 密码，在使用正确的密码配置之前，未使用 VTP 配置引导的设备将不会接受 VTP 通告。配置之后，设备会接受通告中使用相同密码和域名的 VTP 通告。

如果用户要向具有 VTP 功能的网络添加新设备，只有在该设备上配置了适正确的密码之后，新设备才会习得域名。

注意： 配置 VTP 域密码时，如果不为域中的每个设备分配管理域密码的话，管理域将无法正常工作。

VTP 版本

决定使用哪个 VTP 版本时，请遵循以下指导：

- VTP 域中的所有设备必须使用相同的域名，但它们不需要使用相同的版本。
- 如果一台 VTP 版本 2 可用（默认情况下版本 2 未启用）的设备禁用了版本 2，则该设备可以与运行 VTP 版本 1 的设备在同一个域中工作。
- 如果一台运行 VTP 版本 1 但也能运行 VTP 版本 2 的设备接收到了 VTP 版本 3 的通告，它将自动转换到版本 2。
- 如果运行 VTP 版本 3 的设备与运行 VTP 版本 1 的设备相连，则 VTP 版本 1 的设备将转换到 VTP 版本 2，VTP 版本 3 设备发送缩减版的 VTP 数据包，以便 VTP 版本 2 设备更新其数据库。
- 如果一台运行 VTP 版本 3 的设备有扩展 VLAN，则不能转换到版本 1 或版本 2。
- 除非同一个 VTP 域中的所有设备都能使用 VTP 版本 2，否则不要启用设备上的 VTP 版本 2。当用户在一台设备上启用了版本 2，域中所有能使用 VTP 版本 2 的设备都将启用版本 2。如果此时域中某台设备只有版本 1，该设备将不会与启用版本 2 的设备交换 VTP 信息。
- Inspur 建议将使用版本 1 和版本 2 的设备放置在网络的边缘，因为处于边缘的设备不需要转发 VTP 版本 3 的通告。
- 如果用户的网络环境支持令牌环网桥中转功能和令牌环集中器中继功能，用户必须启用 VTP 版本 2 或版本 3，以便令牌环 VLAN 交换能正常运行。要运行令牌环和令牌环网，请禁用 VTP 版本 2。
- VTP 版本 1 和版本 2 不为扩展的 VLAN（VLAN1006 到 4094）传播配置信息。用户必须手动配置这些设备。VTP 版本 3 支持扩展的 VLAN 数据库的传播。
- 当 VTP 版本 3 设备的中继端口接收到了来自 VTP 版本 2 设备的消息后，该设备会在那个中继上以 VTP 版本 2 的格式发送缩减版的 VLAN 数据库。VTP 版本 3 设备不会在中继上发送 VTP 版本 2 格式的数据包，除非它先在中继上收到了 VTP 版本 2 数据包。
- 当 VTP 版本 3 设备在中继端口上检测到了 VTP 版本 2 设备时，该设备将继续发送 VTP 版本 3 数据包，同时也发送 VTP 版本 2 数据包，使得两种类型的邻接设备在同一中继上共存。
- 一台 VTP 版本 3 设备不会接受来自 VTP 版本 2 或 VTP 版本 1 设备的配置信息。
- 两个 VTP 版本 3 区域只能通过 VTP 版本 1 或版本 2 区域以透明模式通信。
- 只能使用 VTP 版本 1 的设备不能与 VTP 版本 3 设备进行互操作。
- VTP 版本 1 和 VTP 版本 2 不能为扩展的 VLAN（VLAN1006 到 4094）传递配置信息。用户必须在每台设备上手动配置 VLAN。

如何配置 VTP

配置 VTP 模式（CLI）

用户可以将 VTP 模式配置为以下之一：

- VTP 服务器模式——在 VTP 服务器模式下，用户可以修改 VLAN 配置并将修改的配置信息通过网络传播。
- VTP 客户端模式——在 VTP 客户端模式下，用户不能修改 VLAN 配置。客户端设备接收来自 VTP 域中的 VTP 服务器的 VTP 更新，根据接收到的更新信息修改配置。

- VTP 透明模式——在 VTP 透明模式下，设备上的 VTP 被禁用。设备不发送 VTP 更新，如果收到来自其他设备的 VTP 更新也不修改配置。但是，一台运行 VTP 版本 2 的 VTP 透明设备会将中继链路上收到的 VTP 通告转发出去。
- VTP 关闭模式——VTP 关闭模式与 VTP 透明模式除了不在中继上转发 VTP 通告这点，其余运行效果一样。

当用户配置域名的时候，域名不能被移除；用户只能将设备分配到其他域中。

总步骤

1. **enable**
2. **configure terminal**
3. **vtp domain** *domain-name*
4. **vtp mode{client| server| transparent| off}{vlan| mst| unknown}**
5. **vtp password** *password*
6. **end**
7. **show vtp status**
8. **copy running-config startup-config**

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	启用特权 EXEC 模式。如果出现提示，请输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**vtp domain** *domain-name* **示例：** `Device(config)#` **vtp domain eng_group**	配置 VTP 管理域名。名称可以是 1 到 32 个字符。具有相同管理责任的所有以 VTP 服务器或客户端模式运行的设备必须配置相同的域名。 对于服务器模式以外的模式，此命令是可选项。VTP 服务器模式需要一个域名。如果设备具有到 VTP 域的中继连接，则设备从域中的 VTP 服务器获取域名。 用户应该在配置其他 VTP 参数之前配置 VTP 域。
步骤 4	**vtp mode {client | server | transparent |off} {vlan | mst | unknown}** **示例：** `Device(config)#` **vtp mode server**	为设备配置 VTP 模式（客户端、服务器、透明或关闭） • **vlan**——VLAN 数据库为默认配置。 • **mst**——多生成树数据库 • **unknown**——未知类型数据库
步骤 5	**vtp password** *password* **示例：** `Device(config)#` **vtp password mypassword**	（可选）设置 VTP 域的密码。密码可以是 8 到 64 个字符。如果配置了 VTP 密码，若用户没有为域中的每个设备分配相同的密码，VTP 域无法正常工作。
步骤 6	**end**	返回特权 EXEC 模式。

	示例： `Device(config)#` **end**	
步骤 7	**show vtp status** **示例：** `Device#` **show vtp status**	请在显示的 *VTP 操作模式*和 *VTP 域名*字段中确认配置的条目。
步骤 8	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）在启动配置文件中保存配置信息。 只有 VTP 模式和域名可以保存在设备正在运行的配置中，并可被复制到启动配置文件中。

配置 VTP 版本 3 的密码（CLI）

用户可以在设备上配置 VTP 版本 3 的密码。

总步骤

1. enable

2. configure terminal

3. vtp password *password***[hidden | secret]**

4. end

5. show vtp password

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	启用特权 EXEC 模式。如果出现提示，请输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**vtp domain** *password***[hidden| secret]** **示例：** `Device(config)#` **vtp password mypasswordhidden**	(可选)为 VTP 域设置密码。密码可以是 8 到 64 个字符。 • （可选）**hidden** ——保存根据 nvram:valn.dat 文件中的密码串生成的秘钥。如果用户通过配置 VTP 主服务器配置接管，系统将提示重新输入密码。 • （可选）**secret**——直接配置密码。密码必须包含 32 个十六进制字符。
步骤 4	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 5	**show vtp password** **示例：**	请确认输入。输出如下： VTP 密码:

	Device# **show vtp password**	89914640C8D90868B6A0D8103847A733
步骤 6	**copy running-config startup-config** **示例：** Device# **copy running-configstartup-config**	（可选）在配置文件中保存配置的条目。

配置 VTP 版本 3 的主服务器（CLI）

当用户将一台 VTP 服务器配置为主服务器时，接管操作开始。

总步骤

1. vtp primary [vlan | mst] [force]

具体步骤

	命令或操作	目的
步骤 1	**vtp primary [vlan | mst] [force]** **示例：** Device# **vtp primary vlanforce**	将设备的操作状态从辅助服务器（默认）更改为主服务器，并将配置发布到域。如果设备密码配置为 **hidden**，系统将提示用户重新输入密码。 • （可选）**vlan**——选择 VLAN 数据库为接管特性。这是默认选项。 • （可选）**mst**——选择多生成树数据库作为接管特性。 • （可选）**force**——覆盖任何有冲突的服务器配置。如果用户不输入 **force**，系统将在接管前提示确认。

启用 VTP 版本（CLI）

VTP 版本 2 和版本 3 在默认情况下被禁用。

- 当用户在一台设备上启用了版本 2，域中所有能使用 VTP 版本 2 的设备都将启用版本 2。若要启动 VTP 版本 3，用户必须在每台设备上手动配置。
- 在使用 VTP 版本 1 和版本 2 的情况下，用户只能在 VTP 服务器或透明模式下的设备上配置版本。如果设备运行的是 VTP 版本 3，则当设备处于客户端模式时，若没有扩展 VLAN，并且未配置隐藏密码，可以将版本更改为版本 2。

注意： 在同一 VTP 域中的 VTP 版本 1 和 VTP 版本 2 设备不可进行互操作。除非 VTP 域中的每个设备都支持版本 2，否则不要启用 VTP 版本 2。

- 在 TrCRF 和 TrBRF 令牌环网络环境下，用户必须启用 VTP 版本 2 或版本 3，以便令牌环 VLAN 交换能正常运行。要运行令牌环和令牌环网，请禁用 VTP 版本 2。

注意： 在 VTP 版本 3 中，主服务器和辅助服务器都可以在域中实例上运行。

总步骤

1. enable

2. configure terminal

3. vtp version {1 | 2 | 3}

4. end

5. show vtp status

6. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	启用特权 EXEC 模式。如果出现提示，请输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**vtp version {1 | 2 | 3}** **示例：** `Device(config)#` **vtp version 2**	在设备上启用 VTP 版本。默认为 VTP 版本 1。
步骤 4	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 5	**show vtp status** **示例：** `Device#` **show vtp status**	请确认配置的 VTP 版本已经启用。
步骤 6	**copy running-config startup-config** **示例：** `Device#` **copy running-configstartup-config**	（可选）在配置文件中保存配置的条目。

启用 VTP 修剪（CLI）

在开始前

VTP 修剪不是为在 VTP 透明模式下工作而设计的。如果网络中的一个或多个设备处于 VTP 透明模式，则应执行以下操作之一：

- 关闭整个网络的 VTP 修剪。
- 通过使设备中继上的所有 VLAN 到其上行的 VTP 透明设备都不可修剪，从而关闭 VTP 修剪。

在接口上配置 VTP 修剪，请使用 **switchport trunk pruning vlan** 接口配置命令。VTP 修剪在接口中继时运行。用户可以设置 VLAN 的可修剪性，是否为 VTP 域启用 VTP 修剪，是否存在任何给定的 VLAN，以及接口是否正在中继。

总步骤

1. enable

2. configure terminal

3. vtp pruning

4. end

5. show vtp status

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	启用特权 EXEC 模式。如果出现提示，请输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**vtp pruning** **示例：** `Device(config)#` **vtp pruning**	在 VTP 管理域启用修剪。 默认情况下，修剪被禁用。用户仅需要为一台 VTP 服务器模式下的设备启用修剪。
步骤 4	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 5	**show vtp status** **示例：** `Device#` **show vtp status**	请在显示的 *VTP 修剪模式*字段中验证配置的条目。

基于端口配置 VTP（CLI）

在 VTP 版本 3 下，用户可以在每个端口上启用或禁用 VTP。用户只能在处于中继模式的端口上启用 VTP。入向和出向 VTP 流量被阻止，不会被转发。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. vtp

5. end

6. show running-config interface *interface-id*

7. show vtp status

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	启用特权 EXEC 模式。如果出现提示，请输入密码。
步骤 2	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**interface** *interface-id* **示例：** `Device(config)#` **interface**	标识接口，并进入接口配置模式。

	gigabitethernet1/0/1	
步骤 4	**vtp** **示例：** `Device(config)#` **vtp**	在指定端口启用 VTP。
步骤 5	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 6	**show running-config interface** *interface-id* **示例：** `Device#` **show running-config interfacegigabitethernet1/0/1**	确认对端口的修改。
步骤 7	**show vtp status** **示例：** `Device#` **show vtp status**	确认配置。

向 VTP 域添加 VTP 客户端（CLI）

在将设备添加到 VTP 域*之前*，请按照以下步骤确认并重置设备上的 VTP 配置修订号。

在开始前

在将 VTP 客户端添加到 VTP 域之前，请确认其 VTP 配置修订号*低于* VTP 域中其他设备的配置修订号。VTP 域中的设备会使用 VTP 配置修订号最高设备的 VLAN 配置。在 VTP 版本 1 和版本 2 中，如果用户添加的设备修订号高于 VTP 域中的修订号，那么该设备可以清除 VTP 服务器和 VTP 域中所有的 VLAN 信息。在 VTP 版本 3 中，VLAN 信息不会被擦除。

用户可以在不影响 VTP 域中的其他设备的情况下，使用 **vtp mode transparent** 全局配置命令禁用设备上的 VTP，然后更改其 VLAN 信息。

总步骤

1. enable

2. show vtp status

3. configure terminal

4. vtp domain domain-name

5. end

6. show vtp status

7. configure terminal

8. vtp domain domain-name

9. end

10. show vtp status

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device>`**enable**	启用特权 EXEC 模式。如果出现提示，请输入密码。
步骤 2	**show vtp status**	检查 VTP 配置修订号。

	示例： `Device#` **show vtp status**	如果修订号为 0，向 VTP 域添加设备。 如果修订号比 0 大，按下面步骤操作： • 记录下域名。 • 记录下配置修订号。 • 继续下一步骤来重设设备配置修订号。
步骤 3	**configure terminal** **示例：** `Device#`**configure terminal**	进入全局配置模式。
步骤 4	**vtp domain** *domain-name* **示例：** `Device(config)#` **vtp domain domain123**	将步骤一中显示的原始域名修改为新域名。
步骤 5	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。设备上的 VLAN 信息被更新，配置修订号被重置为 0。
步骤 6	**show vtp status** **示例：** `Device#` **show vtp status**	确认配置修订号已被重置为 0。
步骤 7	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 8	**vtp domain** *domain-name* **示例：** `Device(config)#` **vtp domain domain012**	在设备上输入原始域名。
步骤 9	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。设备上的 VLAN 信息被更新。
步骤 10	**show vtp status** **示例：** `Device#` **show vtp status**	（可选）请确认现在的域名与步骤一时的域名相同，以及配置修订号为 0。

监控 VTP

这部分描述了用于显示和监控 VTP 配置的命令。
用户通过查看 VTP 配置信息来监视 VTP：域名、当前 VTP 版本和 VLAN 个数。用户还可以查看设备发送和接收的通告的统计信息。

表格 212：VTP 监控命令

命令	目的
show vtp counters	显示已发送和已接收 VTP 消息的数量。
show vtp devices[conflict]	显示域中所有有关 VTP 版本 3 设备的信息。冲突是指与主服务器冲突的 VTP 版本 3 设备。

	在设备处于透明或关闭模式时，**show vtp devices** 命令不显示设备的信息。
show vtp interface[*interface-id***]**	显示所有接口或特定接口的 VTP 状态和配置。
show vtp password	显示 VTP 密码。密码显示的形式依赖于用户是否输入关键字 **hidden** 以及设备上是否启用加密技术。
show vtp status	显示 VTP 设备配置信息。

VTP 配置示例

示例：将交换机配置为主服务器

以下示例表明在配置隐藏或秘密密码时，如何将设备配置为 VLAN 数据库的主服务器（默认）：

```
Device# vtp primary vlan
Enter VTP password: mypassword
This switch is becoming Primary server for vlan feature in the VTP domain
VTP Database Conf Switch ID Primary Server Revision System Name
------------ ---- -------------- -------------- -------- --------------------
VLANDB Yes 00d0.00b8.1400=00d0.00b8.1400 1 stp7
Do you want to continue (y/n) [n]? y
```

接下来做什么？

VTP 配置完成后，用户可以配置以下事项：

- VLAN
- VLAN 群组
- VLAN 中继
- 语音 VLAN

其他参考文档

相关文档

相关主题	文档标题
有关本章中使用命令的完整语法和使用信息。	*VLAN 命令参考（Inspur6650 交换机）* *二/三层命令参考（Inspur6650 交换机）*
附加配置命令和步骤	*LAN 交换配置指南，Inspur INOS（Inspur6650 交换机）* *二/三层配置指南（Inspur6650 交换机）*

错误信息解释

描述	链接
为了便于用户研究和解决此版本中的系统错误消息，可以使用错误消息解码器工具。	http://www.icntnetworks.com

标准和 RFC

标准/RFC	标题
RFC 1573	MIB-II 的接口组的演变
RFC 1757	远程网络监控管理
RFC 2021	使用 SMIv2 的传输控制协议的 SNMPv2 管理信息库

技术助手

描述	链接
Inspur 支持网站提供了广泛的在线资源，包括用于故障排除以及解决 Inspur 产品问题和技术问题的文档和工具。 要获得有关产品的安全和技术信息，用户可以订阅不同的服务，例如产品警报工具（从现场记录获取），Inspur 技术服务实时通讯和简单讯息聚合订阅（RSS）。 要获取 Inspur 支持网站上的大多数工具，需要有 icntnetworks.com 的用户 ID 和密码。	http://www.icntnetworks.com

VTP 的历史特性和信息

版本	修订
Inspur INOS 11.3.1	此特性被引入。

配置 VLAN

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

VLAN 的前提

以下是配置 VLAN 的前提和注意事项：

- 在创建 VLAN 之前，用户必须决定是否使用 VLAN 中继协议（VLAN Trunking Protocol，VTP）来维护网络的全局 VLAN 配置。
- 如果用户要在设备上配置多个 VLAN 并启用路由功能，可以将交换机数据库管理（Switch Database Management，SDM）功能设置为 VLAN 模板，该模板可配置系统资源以便支持最大单播 MAC 地址数。
- 运行 LAN Base 特性集的设备仅支持 SVI 上的静态路由。
- 设备中应有一个 VLAN，以便能够将其添加到 VLAN 组。

VLAN 的限制

以下是 VLAN 的限制：

- 设备支持每 VLAN 生成树加（per-VLAN spanning-tree plus，PVST +）或快速 PVST +最多包含 128 个生成树实例。每个 VLAN 允许有一个生成树实例。
- 设备支持 IEEE 802.1Q 中继方法，用于通过以太网端口发送 VLAN 流量。
- 不支持配置接口 VLAN 路由器的 MAC 地址。接口 VLAN 已有默认分配的 MAC 地址。
- 设备不支持私有 VLAN。
- 用户不能在交换机堆栈中混用 Inspur3850 和 Inspur6650 交换机。

关于 VLAN 的信息

逻辑网络

VLAN 是根据功能、项目团队或应用程序逻辑划分的交换网络，无需考虑用户的物理位置。VLAN 具有与物理 LAN 相同的属性，但可将物理上位于不同 LAN 段的终端设备分成一组。任何设备端口都可以属于 VLAN，并且单播、广播和组播数据包只能转发和泛洪到 VLAN 中的终端设备上。每个 VLAN 被认为是一个逻辑网络，并且目的地不属于 VLAN 的数据包必须通过路由器或支持回退桥接的设备转发。在设备堆栈中，VLAN 可以由不同堆栈的端口形成。由于 VLAN 被认为是一个单独的逻辑网络，所以它包含自己的网桥管理信息库（Management Information Base，MIB）信息，并且可以支持实现自己的生成树。

图 139：VLAN 作为逻辑定义网络

Cisco router	思科路由器
Gigabit Ethernet	千兆以太网
Engineering VLAN	工程部 VLAN
Marketing VLAN	市场部 VLAN
Accounting VLAN	会计部 VLAN
Floor x	层 x

VLAN 通常与 IP 子网关联。例如，特定 IP 子网中的所有终端设备属于同一 VLAN。设备上的接口 VLAN 成员资格是基于逐个接口手动分配的。使用此方法将设备接口分配给 VLAN 时，

称为基于接口或静态的 VLAN 成员资格。
VLAN 之间的流量必须被路由。
设备可以通过使用设备虚拟接口（Switch Virtual Interface，SVI）在 VLAN 之间路由流量。必须显式配置 SVI 并为其分配一个 IP 地址，以便 SVI 在 VLAN 之间路由流量。

VLAN 支持

设备支持 VTP 客户端，服务器和透明模式下的 VLAN。VLAN 由 1 到 4094 之间的数字标识。VLAN 1 是默认 VLAN，是在系统初始化期间创建的。VLAN ID 1002 到 1005 被保留用于令牌环和 FDDI VLAN。除了 1002 到 1005 之外的所有 VLAN 都可用于用户配置。
有 3 个 VTP 版本：VTP 版本 1，版本 2 和版本 3。所有的 VTP 版本都支持正常的和扩展范围的 VLAN，但只有在 VTP 版本 3 中设备才支持传播扩展范围的 VLAN 配置信息。当在 VTP 版本 1 和 2 中创建扩展范围的 VLAN 时，不会传播其配置信息。即使设备上的本地 VTP 数据库记录不更新，但扩展范围的 VLAN 配置信息也会创建并存储在设备正在运行的配置文件中。
用户可以在设备上配置多达 4094 个 VLAN。

VLAN 端口成员资格模式

用户通过分配成员资格模式来配置端口属于某 VLAN，该成员资格模式指定端口承载的流量类型以及它可以被多少个 VLAN 使用。
当一个端口属于某 VLAN 时，该设备将基于每个 VLAN 了解和管理与该端口关联的地址。
表格 213：端口成员资格模式和特性

成员资格模式	VLAN 成员资格特性	VTP 特性
静态接入	一个静态接入端口可以属于一个 VLAN，并且可以手动分配给该 VLAN。	VTP 配置不是必需的。如果用户不希望 VTP 全局传播信息，请将 VTP 模式设置为透明。要参与 VTP，设备或设备堆栈上必须至少有一个中继端口连接到第二个设备或设备堆栈的中继端口上。
中继（IEEE802.1Q）： • IEEE802.1Q——行业标准的中继封装。	中继端口是所有 VLAN 的默认成员，包括扩展范围的 VLAN，但是成员资格可受限于允许传输的 VLAN 列表的配置，用户还可以修改可修剪列表，阻止在中继端口上把流量泛洪到列表中的 VLAN。	建议使用 VTP，但不是必需的。VTP 通过在全网的基础上管理 VLAN 的添加、删除和重命名，从而维护 VLAN 配置的一致性。VTP 与其他设备通过中继链路交换 VLAN 配置信息。
语音 VLAN	语音 VLAN 端口是连接到 Inspur IP 电话的接入端口，对于连接到电话的设备，使用一个 VLAN 用于传输语音流量，另一个 VLAN 用于传输数据流量。	VTP 配置不是必需的；VTP 对语音 VLAN 没有影响。

VLAN 配置文件

VLAN ID 1 到 1005 的配置写在 vlan.dat 文件（VLAN 数据库），用户可以通过输入 **show vlan** 特权 EXEC 命令查看配置信息。vlan.dat 文件存储在闪存中。如果 VTP 模式为透明，文件也将保存到设备正在运行的配置文件中。

在设备堆栈中，整个堆栈使用相同的 vlan.dat 文件和运行配置。在某些设备上，vlan.dat 文件存储在活跃设备的闪存中。

用户可以使用接口配置模式定义端口成员资格模式，以及从 VLAN 添加和删除端口。这些命令的结果将写入设备正在运行的配置文件中，用户可以通过输入 **show running-config** 特权 EXEC 命令查看该文件。

当用户将 VLAN 和 VTP 信息（包括扩展范围的 VLAN 配置信息）保存到设备启动配置文件中并重启设备时，设备的配置选择如下：

- 如果启动配置中的 VTP 处于透明模式，且 VLAN 数据库和 VLAN 数据库中的 VTP 域名与启动配置文件中的 VTP 域名匹配，则忽略（清除）VLAN 数据库，并使用启动配置文件中的 VTP 和 VLAN 配置。VLAN 数据库版本号在 VLAN 数据库中保持不变。
- 如果启动配置中的 VTP 模式或域名与 VLAN 数据库不匹配，那么 VLAN ID 1 到 1005 的域名、VTP 模式和配置将使用 VLAN 数据库中的信息。
- 在 VTP 版本 1 和 2 中，如果 VTP 是服务器模式，VLAN ID 1 至 1005 的域名和 VLAN 配置将使用 VLAN 数据库信息。VTP 版本 3 还支持 VLAN 1006 到 4094。
- 从 15.0（02）SE6 镜像开始，在 VTP 透明和关闭模式下，即使 VLAN 不应用于接口，它们仍从 startup-config 创建。

正常范围 VLAN 配置指南

正常范围的 VLAN 指 ID 从 1 到 1005 的 VLAN。

在网络中创建和修改正常范围 VLAN 时，请参考以下指南：

- 正常范围的 VLAN 标识编号介于 1 和 1001 之间。VLAN 编号 1002 到 1005 为令牌环和 FDDI VLAN 保留。
- VLAN ID 1 到 1005 的 VLAN 配置一般都保存到 VLAN 数据库中。如果 VTP 模式为透明，VTP 何 VLAN 配置也将保存到设备正在运行的配置文件中。
- 如果设备处于 VTP 服务器或 VTP 透明模式，用户可以添加、修改或移除 VLAN 数据库中 VLAN 2 到 1001 的配置信息（VLAN ID 为 1 和 1002 到 1005 之间的配置信息是自动创建的，无法删除）。
- 在 VTP 透明模式下创建的扩展范围 VLAN 不会保存于 VLAN 数据库中，也不会传播。VTP 版本 3 支持在 VTP 服务器模式下的扩展范围 VLAN（VLAN 1006 到 4094）的数据库传播。
- 在用户能创建 VLAN 之前，设备必须处于 VTP 服务器模式或 VTP 透明模式。如果设备是一个 VTP 服务器，用户必须定义 VTP 域，否则 VTP 将无法运行。
- 设备不支持令牌环或 FDDI。设备不转发 FDDI，FDDI-Net，TrCRF 或 TrBRF 流量，但它通过 VTP 传播 VLAN 配置。
- 设备支持 128 个生成树实例。如果设备具有比支持的生成树实例更多的活跃 VLAN，那么可以在 128 个 VLAN 上启用生成树，并在其余 VLAN 上禁用生成树。如果用户已经在设备上使用了所有可用的生成树实例，那么在 VTP 域中的任何位置添加另一个 VLAN 会在该设备上创建一个未运行生成树的 VLAN。如果用户在该设备的中继端口上具有默认

允许列表（允许所有 VLAN），则所有中继端口上都具有 VLAN。根据网络的拓扑情况，新 VLAN 中可能会创建出一个不会断开的环路，特别是在有几个相邻设备都已用完生成树实例的情况下。用户可以通过在已耗尽其对生成树实例的分配的设备的中继端口上设置允许列表来防止这种情况发生。

如果设备上的 VLAN 数量超过支持的生成树实例数，我们建议用户在设备上配置 IEEE 802.1s Multiple STP（MSTP），便于将多个 VLAN 映射到单个生成树实例上。

- 当堆栈中的设备习得一个新 VLAN，或删除、修改现有 VLAN（通过网络端口上的 VTP 或通过 CLI）时，VLAN 信息将通告给所有堆栈成员。
- 当一台设备加入堆栈或堆栈进行合并时，新设备上的 VTP 信息（vlan.dat 文件）将与活跃设备保持一致。

扩展范围 VLAN 的配置指南

扩展范围的 VLAN 指 VLAN ID 编号从 1006 到 4094.

在创建扩展范围的 VLAN 时，请参考以下指南：

- 扩展范围的 VLAN ID 不保存在 VLAN 数据库中，VTP 无法识别，除非设备使用 VTP 版本 3。
- 用户不能将扩展范围的 VLAN 包含进可修剪范围。
- 对于 VTP 版本 1 或 2，用户可以在全局配置模式下将 VTP 模式设置为透明。用户应该将此配置保存到启动配置，以便设备以 VTP 透明模式启动。否则，一旦设备重置，扩展范围 VLAN 的配置将丢失。如果在 VTP 版本 3 中创建扩展范围的 VLAN，它们将无法转换到 VTP 版本 1 或 2。
- 在设备堆栈中，整个堆栈使用相同的运行配置和保存的配置，并且在堆栈中共享扩展范围 VLAN 的信息。

如何配置 VLAN

如何配置正常范围 VLAN

在 VLAN 数据库中创建一个新的正常范围 VLAN 或修改一个现有 VLAN 时，可以设置以下参数：

- VLAN ID
- VLAN 名
- VLAN 类型
  - 以太网
  - 光纤分布式数据接口（Fiber Distributed Data Interface，FDDI）
  - FDDI 网络实体名称（Network Entity Title，NET）
  - 令牌环网桥中转功能或令牌环集中器中继功能
  - 令牌环
  - 令牌环网
- VLAN 状态（活跃或暂停）
- VLAN 的最大传输单元（Maximum Transmission Unit，MTU）

- 安全性关联标识符（Security Association Identifier，SAID）
- TrBRF VLAN 的网桥标识符
- FDDI 和 TrCRF VLAN 的环号
- TrCRF VLAN 的父 VLAN 号
- TrCRF VLAN 的生成树协议（Spanning Tree Protocol，STP）类型
- 从一个 VLAN 类型转换到另一个 VLAN 类型时使用的 VLAN 号

如果用户尝试手动删除 vlan.dat 文件，可能会导致 VLAN 数据库不一致。如果要修改 VLAN 配置，请按照本节中的步骤操作。

创建或修改一个以太网 VLAN（CLI）

在开始前

对于 VTP 版本 1 和 2，如果设备处于 VTP 透明模式，可以分配大于 1006 的 VLAN ID，但它们将不会被添加到 VLAN 数据库。

设备仅支持以太网接口。由于不支持本地 FDDI 和令牌环 VLAN，因此用户仅为其他设备配置用于 VTP 全局通告的 FDDI 和令牌环特定介质的特性。

虽然设备不支持令牌环连接，但是具有令牌环连接的远程设备可以通过支持令牌环连接的设备进行管理。运行 VTP 版本 2 的设备会通告关于这些令牌环 VLAN 的信息：

- 令牌环 TrBRF VLAN
- 令牌环 TrCRF VLAN

总步骤

1. configure terminal

2. vlan *vlan-id*

3. name *vlan-name*

4. media { ethernet | fd-net | fddi | tokenring | trn-net }

5. remote-span

6. end

7. show vlan {name*vlan-name***| id** *vlan-id***}**

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 2	**vlan** *vlan-id* **示例：** `Device(config)# vlan 20`	输入 VLAN ID 和 VLAN 配置模式。输入一个新的 VLAN ID 来创建 VLAN，或输入现有 VLAN ID 来修改该 VLAN。 **注释：** 这个命令可用的 VLAN ID 范围从 1 到 4094。
步骤 3	**name***vlan-name* **示例：** `Device(config-vlan)# name test20`	（可选）为 VLAN 输入一个名称。如果没有为 VLAN 输入名称，默认名是将带有前导零的 *vlan-id* 值附加到单词 VLAN 后。例如，VLAN0004 是 VLAN 4 的默认 VLAN 名称。
步骤 4	**media{ethernet| fd-net| fddi| tokenring| trn-net}** **示例：**	配置 VLAN 介质类型。可选命令有： • ethernet——将 VLAN 介质类型设为以太网。

	Device(config-vlan)# **media ethernet**	• fd-net——将 VLAN 介质类型设为 FDDI 网。 • fddi——将 VLAN 介质类型设为 FDDI。 • tokenring——将 VLAN 介质类型设为令牌环。 • trn-net——将 VLAN 介质类型设为令牌环网。
步骤 5	**remote-span** **示例：** Device(config-vlan)# **remote-span**	（可选）为远程 SPAN 会话将 VLAN 配置为 RSPAN VLAN。有关远程 SPAN 的详细信息，请参阅 *Inspur 6650 网络管理配置指南*。
步骤 6	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式。
步骤 7	**show vlan{ name** *vlan-name***| id** *vlan-id***}** **示例：** Device# **show vlan name test20 id 20**	验证配置的条目。

删除 VLAN（CLI）

当用户从处于 VTP 服务器模式的设备中删除 VLAN 时，VTP 域中所有设备的 VLAN 数据库都会删除该 VLAN。若从处于 VTP 透明模式的设备中删除 VLAN 时，仅在特定设备或设备堆栈上删除该 VLAN。

用户不能删除不同介质类型的默认 VLAN：以太网 VLAN 1 和 FDDI 或令牌环 VLAN 1002 至 1005。

注意： 用户删除 VLAN 时，分配给该 VLAN 的端口都将变为非活跃状态。除非用户将它们分配给新的 VLAN，否则将一直保持与 VLAN 的关联（因此处于非活跃状态）。

总步骤

1. enable

2. configure terminal

3. no vlan *vlan-id*

4. end

5. show vlan brief

6. copy running-config startup-config

详细步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** Device>**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式。
步骤 3	**no vlan** *vlan-id* **示例：** Device(config)# **no vlan 4**	通过输入 VLAN ID 来删除 VLAN。
步骤 4	**end** **示例：**	返回特权 EXEC 模式。

	Device(config)# **end**	
步骤 5	**show vlan brief** **示例：** Device# **show vlan brief**	验证删除的 VLAN。
步骤 6	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）在配置文件中保存用户配置条目。

为 VLAN 分配静态接入端口（CLI）

用户可以为一个通过禁用 VTP（处于 VTP 透明模式）阻止 VLAN 配置信息全局传播的 VLAN 分配一个静态接入端口。

如果用户想把集群成员设备上的端口分配给 VLAN，请先使用 **rcommand** 特权 EXEC 命令登录到集群成员交换机上。

如果用户将接口分配给不存在的 VLAN 时，新 VLAN 会被创建。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. switchport mode access

5. switchport access vlan *vlan-id*

6. end

7. show running-config interface *interface-id*

8. show interfaces*interface-id***switchport**

9. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** Device>**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式。
步骤 3	**interface** *interface-id* **示例：** Device(config)# **interface gigabitethernet2/0/1**	输入要添加到 VLAN 的接口。
步骤 4	**switchport mode access** **示例：** Device(config-if)# switchport mode access	为端口（二层接入端口）定义 VLAN 成员资格模式。
步骤 5	**switchport access vlan** *vlan-id* **示例：** Device(config-if)# **switchport access vlan 2**	为 VLAN 分配端口。有效的 VLAN ID 是从 1 到 4094。

步骤 6	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 7	**show running-config interface** *interface-id* **示例：** `Device#` **show running-config interfacegigabitethernet2/0/1**	验证接口的 VLAN 成员资格模式。
步骤 8	**show interfaces** *interface-id* **switchport** **示例：** `Device#` **show interfaces gigabitethernet2/0/1switchport**	在显示的 *Administrative Mode* 和 *Access Mode VLAN* 字段中验证用户配置条目。
步骤 9	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）在配置文件中保存用户配置条目。

如何配置扩展 VLAN

扩展范围的 VLAN 使得服务提供商能为更多客户提供他们的基础设施。只要 **switchport** 命令能用 VLAN ID，该命令就能用扩展范围的 VLAN ID。

对于 VTP 版本 1 或 2，扩展范围 VLAN 的配置不在 VLAN 数据库中存储，但由于 VTP 模式为透明，所以配置信息存储于设备正在运行的配置文件中，用户还可以将配置保存到启动配置文件中。在 VTP 版本 3 中创建的扩展范围的 VLAN 将存储在 VLAN 数据库中。

用户只能更改扩展范围 VLAN 上的 MTU 大小和远程 SPAN 配置状态; 所有其他特性必须保持默认状态。

创建扩展范围的 VLAN（CLI）

总步骤

1. enable

2. configure terminal

3. vlan *vlan-id*

4. remote-span

5. exit

6. end

7. show vlan id *vlan-id*

8. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal**	进入全局配置模式。

	示例： `Device#` **configure terminal**	
步骤 3	**vlan** *vlan-id* **示例：** `Device(config)#` **vlan 2000** `Device(config-vlan)#`	输入扩展范围的 VLAN ID 并输入 VLAN 配置模式。范围从 1006 到 4049。
步骤 4	**remote-span** **示例：** `Device(config-if)# remote-span`	（可选）将 VLAN 配置为 RSPAN VLAN。
步骤 5	**exit** **示例：** `Device(config-vlan)#` **exit** `Device(config)#`	返回配置模式。
步骤 6	**end** **示例：** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 7	**show vlan id** *vlan-id* **示例：** `Device#` **show vlan id 2000**	验证 VLAN 已被创建。
步骤 8	**copy running-config startup-config** **示例：** `Device#` **copy running-config startup-config**	（可选）在配置文件中保存用户配置条目。

监控 VLAN

表格 214：特权 EXEC show 命令

命令	目的
show interfaces [vlan *vlan-id***]**	显示所有接口或设备上配置的指定 VLAN 的特性。
show vlan [access-map *name* **| brief | dot1q { tag native } | filter [access-map | vlan] | group [group-name** *name* **] | id** *vlan-id* **| ifindex | mtu | name** *name* **| remote-span | summary]**	显示所有 VLAN 或设备上指定 VLAN 的参数。可选命令如下： • **access-map**——显示 VLAN 接入映射。 • **brief**——简短显示 VTP VLAN 状态。 • **dol1q**——显示 dot1q 参数。 • **filter**——显示 VLAN 过滤信息。 • **group**——显示 VLAN 组的名称和已连接的可用 VLAN。 • **id**——通过标识编号来显示 VTP VLAN 的状态。 • **ifindex**——显示 SNMP 的 ifIndex 信息。 • **mtu**——显示 VLAN MTU 信息。 • **name**——通过指定的名称显示 VTP

	VLAN 信息。 • **remote-span**——显示远程 SPAN VLAN。 • **summary**——显示 VLAN 信息的总结。

接下来做什么？

在配置 VLAN 之后，用户可以配置以下内容：

- VLAN 组
- VLAN 中继协议（VLAN Trunking Protocol，VTP）
- VLAN 中继
- 语音 VLAN

其他参考资料

相关文档

相关主题	文档标题
有关本章中使用命令的完整语法和使用信息。	*VLAN 命令参考（Inspur6650 交换机）* *二/三层命令参考（Inspur6650 交换机）*
VLAN 接入映射	*安全配置指南（Inspur6650 交换机）* *安全命令参考资源（Inspur6650 交换机）*
VLAN 和移动代理	*移动配置指南，Inspur INOS（Inspur6650 交换机）*
Inspur 灵活网络流量	*Inspur 灵活网络流量配置，Inspur INOS（Inspur 6650 交换机）*
IGMP 侦听	*IP 多播路由命令参考资源（Inspur 6650 交换机）* *IP 多播路由配置指南（Inspur 6650 交换机）*
IPv6	*IPv6 配置指南（Inspur 6650 交换机）* *IPv6 命令参考资源（Inspur 6650 交换机）*
SPAN	*网络管理命令参考资源（Inspur 6650 交换机）* *网络管理配置指南（Inspur 6650 交换机）*
独立平台的配置信息	*基于身份的网络服务配置指南，Inspur INOS（Inspur 6650 交换机）*

错误信息解释

描述	链接
为帮助管理员搜索并解决该版本中的系统错误信息，管理员可使用错误信息解释工具。	http://www.icntnetworks.com

标准和 RFC

标准/RFC	标题
RFC 1573	MIB-II 的接口组的演变
RFC 1757	远程网络监控管理
RFC 2021	使用 SMIv2 的传输控制协议的 SNMPv2 管理信息库

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	http://www.icntnetworks.com

VTP 的历史特性和信息

版本	修订
Inspur INOS 11.3.1	支持 VLAN GUI

配置 VLAN 组

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

VLAN 组的前提

- 设备中应有一个 VLAN，以便能够将其添加到 VLAN 组。
- 为了使 VLAN 组正常工作，除了全局启用 DHCP 侦听功能外，用户还必须保证在所有的 VLAN 中都启用了 DHCP 侦听功能。

VLAN 组的限制

能够映射到 VLAN 组的 VLAN 数量不受 Inspur INOS 软件版本的限制。但是，如果 VLAN 组中的 VLAN 数量超过了建议值 32，则不希望设备有移动行为，并且在 VLAN 组中，某些 VLAN 的二层组播会被中断。因此，管理员的责任是在 VLAN 组中配置可行数量的 VLAN。在将 VLAN 添加到已映射到 WLAN 且已有 32 个 VLAN 的 VLAN 组时，会提示警告。但是当新的 VLAN 组映射到多于 32 个 VLAN 的 WLAN 时，会提示错误。
对于 VLAN 组的预期行为，组中映射的 VLAN 必须在设备中。不支持静态 IP 客户端行为。

如何配置 VLAN 组

创建 VLAN 组（CLI）

总结步骤

1. configure terminal

2. vlan group *WORD* **vlan-list** *vlan-ID*

3. end

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例：** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**vlan group** *WORD* **vlan-list** *vlan-ID* **示例：** `Device(config)#`**vlan groupvlangrp1 vlan-list 91-95**	创建给定组名（vlangrp1）的 VLAN 组，并添加命令中列出的所有 VLAN。VLAN 列表的取值范围为 1 到 4096，建议组中的 VLAN 数量为 32。
步骤 3	**end** **示例：** `Device(config)#` **end**	退出全局配置模式并返回特权 EXEC 模式。或者，按 **CTRL-Z** 退出全局模式。

移除 VLAN 组（CLI）

总步骤

1. configure terminal

2. vlan group *WORD* **vlan-list** *vlan-ID*

3. no vlan group *WORD* **vlan-list** *vlan-ID*

4. end

具体步骤

步骤 1	**configure terminal** **示例：** `Device#`**configure terminal**

	进入全局命令模式。
步骤 2	**vlan group** *WORD* **vlan-list** *vlan-ID* **示例：** `Device(config)#`**vlan group vlangrp1 vlan-list 91-95** 创建给定组名（vlangrp1）的 VLAN 组，并添加命令中列出的所有 VLAN。VLAN 列表的取值范围为 1 到 4096，建议组中的 VLAN 数量为 32。
步骤 3	**no vlan group WORD vlan-list vlan-ID** **示例：** `Device(config)#`**no vlan group vlangrp1 vlan-list 91-95** 移除给定组名的 VLAN 组。
步骤 4	**示例：** `Device(config)#` **end** 退出全局配置模式并返回特权 EXEC 模式。或者，按 **CTRL-Z** 退出全局模式。

查看 VLAN 组中的 VLAN

命令	描述
show vlan group	显示 VLAN 组名列表和可用 VLAN 列表。
show vlan group group-name **<group_name>**	显示指定 VLAN 组的具体信息。

接下来做什么？

在配置 VLAN 组后，用户可以配置以下内容：

- VLAN
- VLAN 中继协议（VLAN Trunking Protocol，VTP）
- VLAN 中继
- 语音 VLAN

其他参考资料

相关文档

相关主题	**文档标题**
有关本章中使用命令的完整语法和使用信息。	*VLAN 命令参考（Inspur6650 交换机）* *二/三层命令参考（Inspur6650 交换机）*
VLAN 接入映射	*安全配置指南（Inspur6650 交换机）* *安全命令参考资源（Inspur6650 交换机）*
VLAN 和移动代理	*移动配置指南，Inspur INOS（Inspur6650 交换机）*
Inspur 灵活网络流量	*Inspur 灵活网络流量配置，Inspur INOS（Inspur 6650 交换机）*
IGMP 侦听	*IP 多播路由命令参考资源（Inspur 6650 交换机）* *IP 多播路由配置指南（Inspur 6650 交换机）*
IPv6	*IPv6 配置指南（Inspur 6650 交换机）*

	IPv6 命令参考资源（Inspur 6650 交换机）
SPAN	*网络管理命令参考资源（Inspur 6650 交换机）* *网络管理配置指南（Inspur 6650 交换机）*
独立平台的配置信息	*基于身份的网络服务配置指南，Inspur INOS（Inspur 6650 交换机）*

错误信息解释

描述	链接
为帮助管理员搜索并解决该版本中的系统错误信息，管理员可使用错误信息解释工具。	http://www.icntnetworks.com

标准和 RFC

标准/RFC	标题
RFC 1573	MIB-II 的接口组的演变
RFC 1757	远程网络监控管理
RFC 2021	使用 SMIv2 的传输控制协议的 SNMPv2 管理信息库

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	http://www.icntnetworks.com

VTP 的历史特性和信息

版本	修订
Inspur INOS 11.3.1	支持 VLAN GUI

配置 VLAN 中继

查询特征信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用浪潮特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问浪潮特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

VLAN 中继的前提

IEEE 802.1Q trunks 在网络的中继策略中增加了如下限制：

- 在通过 IEEE 802.1Qtrunk 连接的 Inspur 设备网络中,每个 VLAN 维护一个生成树实例。非 Inspur 设备的网络中所有 VLAN 维护一个生成树实例。
  - 当通过 IEEE 802.1Q trunk 将 Inspur 设备连接到非 Inspur 设备时，Inspur 设备会将中继的 VLAN 的生成树实例与非 Inspur IEEE 802.1Q 设备的生成树实例合并。但是，每个 VLAN 的生成树信息由 Inspur 设备维护，这些设备由一组非 Inspur IEEE 802.1Q 设备隔开。分离 Inspur 设备的非 InspurIEEE 802.1Q 的云被视为设备之间的单条 trunk 链路。
- 确保 IEEE 802.1Q trunk 的本征 VLAN 在中继链路的两端是相同的。如果中继的一端的本征 VLAN 与另一端的本征 VLAN 不同，则可能会导致生成树环路。
- 在 IEEE 802.1Qtrunk 的本征 VLAN 上禁用生成树，而不禁用网络中每个 VLAN 上的生成树可能会导致生成树环路。建议在 IEEE 802.1Qtrunk 的本征 VLAN 上启用生成树，或在网络中的每个 VLAN 上禁用生成树。在禁用生成树之前，请确保网络无环路。

VLAN 中继的限制条件

以下是 VLAN 的限制条件：

- 中继端口不能是安全端口。
- 中继端口可以分组在 EtherChanne 端口组中，但组中的所有中继必须具有相同的配置。当首次创建组时，所有端口都遵循添加到组中的第一个端口的参数。如果更改其中一个参数的配置，设备会将更改的设置传播到组中的所有端口：
  - 允许 VLAN 列表。
  - 每个 VLAN 的 STP 端口优先级。
  - STP 端口快速设置。
  - 中继状态：

    如果端口组中的一个端口不再是中继，所有端口都不再是中继。
- 建议在每个 VLAN 生成树（PVST）模式下配置不超过 24 个中继端口，并且在多生成树

（MST）模式下配置不超过 40 个中继端口。

- 如果用户尝试在中继端口上启用 IEEE 802.1x，则会显示错误消息，并且不启用 IEEE 802.1x。如果用户尝试将启用 IEEE 802.1x 的端口模式更改为中继，则端口模式不会随其他端口改变。
- 动态模式下的接口可以与其邻接接口协商成为中继端口。如果用户尝试在动态端口上启用 IEEE 802.1x，则会显示错误消息，并且不启用 IEEE 802.1x。如果用户尝试将启用 IEEE 802.1x 的端口模式更改为动态，则端口模式不会更改。
- 隧道端口不支持动态中继协议（Dynamic Trunking Protocol，DTP）。
- 设备不支持三层中继；不能配置子接口或在第 3 层接口使用 **encapsulation** 关键字。设备支持第 2 层中继和第 3 层 VLAN 接口，为他们提供等效功能。
- 不能在交换机堆栈中混合使用 Inspur 3850 和 Inspur 6650 交换机。

关于 VLAN 中继的信息

中继概述

中继是一个或多个以太网设备接口与另一个网络设备（如路由器或设备）之间的点对点链路。以太网中继通过单个链路承载多个 VLAN 的流量，用户可以在整个网络上扩展 VLAN。
以下中继封装在所有以太网接口上可用：

- IEEE 802.1Q-中继封装的行业标准。

中继模式

以太网中继接口支持不同的中继模式。可以将接口设置为中继或非中继，或与邻居接口协商中继。自动协商中继，接口必须在同一个 VTP 域中。
中继协商由动态中继协议（DTP）管理，DTP 是点对点协议（Point-to-Point Protocol，PPP）。但是，一些网络互连设备可能不正确地转发 DTP 帧，可能会导致配置错误。

二层接口模式

表 215：二层接口模式

模式	目的
switchport mode access	将接口（接入端口）置于永久非中继模式，并协商将链路转换为非中继链路。无论相邻接口是否是中继接口，接口都将成为非中继接口。
switchport mode dynamic auto	使接口能够将链路转换为中继链路。如果相邻接口设置为 **trunk** 或 **desirable** 模式，则接口变为中继接口。所有以太网接口的默认交换机端口模

	式是 **dynamic auto** 模式。
switchport mode dynamic desirable	使接口主动尝试将链路转化为中继链路。如果邻接口被设置为 **trunk** 或 **desirable**，或 **auto** 模式，则接口变为中继接口。
switchport mode trunk	使接口进入永久中继模式，并协商将相邻链路转化为中继链路。即使邻接接口不是中继接口，此接口也变为中继接口。
switchportnonegotiate	防止接口生成 DTP 帧。只有当接口 **switchport** 模式为 **access** 或 **trunk** 时，才能使用此命令。必须手动配置邻居接口作为中继接口来建立中继链路。

中继允许 VLAN

缺省情况下，中继端口向所有 VLAN 发送和接收流量。每个中继上允许所有 VLAN ID（1 到 4094）。可以从允许列表中删除 VLAN，阻止来自这些 VLAN 的流量通过中继。

为了降低生成树环路或风暴的风险，可以通过从允许的列表中删除 VLAN 1 来禁用任何单个 VLAN 中继端口上的 VLAN 1。当从 Trunk 端口删除 VLAN 1 时，接口继续发送和接收 VLAN1 中的管理流量，例如，Inspur Discovery Protocol (CDP)，端口聚合协议（PAgP），链路聚合控制协议（Link Aggregation Control Protocol），DTP 以及 VTP。

如果禁止 VLAN 1 的中继端口转换为非中继端口，它会被添加到接入 VLAN。如果接入 VLAN 设置为 1，则端口将添加到 VLAN 1，而不考虑 **switchport trunkallowed** 设置。对于已在端口上禁用的任何 VLAN 也是如此。

如果 VLAN 已启用，VTP 知道此 VLAN，并且 VLAN 位于该端口的允许列表中，则中继端口可以成为该 VLAN 的成员。当 VTP 检测到新启用的 VLAN 并且该 VLAN 在中继端口的允许列表中，中继端口自动成为启用的 VLAN 的成员。当 VTP 检测到新的 VLAN，并且该 VLAN 不在中继端口的允许列表中，中继端口不成为新的 VLAN 的成员。

中继端口的负载均衡

负载均衡会分配设备之间并行中继提供的带宽。为了避免环路，STP 通常仅允许设备之间并行链路中的一个发送流量。使用负载均衡，可以根据流量所属的 VLAN 在链路之间分配流量。可以通过使用 STP 端口优先级或 STP 路径开销在中继端口上配置负载均衡。对于使用 STP 端口优先级的负载均衡，两个负载均衡链路必须连接到同一设备。对于使用 STP 路径成本的负载均衡，每个负载均衡链路可以连接到同一设备或两个不同的设备。

使用 STP 优先级进行网络负载均衡

当同一设备上的两个端口形成环路时，设备使用 STP 端口优先级来决定哪个端口启用，哪个端口阻塞。可以在并行中继端口上设置优先级，以便该端口承载给定 VLAN 的所有流量。对于同一 VLAN，具有较高优先级（较低值）的中继端口转发该 VLAN 流量。对于同一 VLAN，具有较低优先级（较高值）中继端口对于该 VLAN 仍保持阻塞状态。同一个中继端口发送或接收 VLAN 的所有流量。

使用 STP 路径开销进行网络负载均衡

您可以在中继上设置不同的路径开销，把路径开销与不同的 VLAN 关联，为不同 VLAN 阻塞不同端口，以此来配置并行中继以分配 VLAN 流量。VLAN 保持流量分离，并在链路丢失的情况下保持冗余。

特性交互

中继通过以下方式和其他功能交互：

- 中继端口不能是安全端口。
- 中继端口可以分组在 EtherChanne 端口组中，但组中的所有中继必须具有相同的配置。当首次创建组时，所有端口的设置都遵循被添加到组中的第一个端口的参数。如果更改了其中一个参数的配置，设备会将输入的设置传播到组中的所有端口：
  - 允许 VLAN 列表
  - 每个 VLAN 的 STP 端口优先级。
  - STP 端口快速设置。
  - 中继状态：

    如果端口组中的一个端口不再是 trunk，所有端口都不再是 trunk。
- 建议在每个 VLAN 生成树（PVST）模式下配置不超过 24 个 trunk 端口，并且在多生成树（MST）模式下配置不超过 40 个 trunk 端口。
- 如果尝试在中继端口上启用 IEEE 802.1x，则会显示错误消息，并且不启用 IEEE 802.1x。如果尝试将启用 IEEE 802.1x 的端口的模式更改为中继，则端口模式不会更改。
- 动态模式下的端口可以与其邻居协商成为中继端口。如果尝试在动态端口上启用 IEEE 802.1x，则会显示错误消息，并且不启用 IEEE 802.1x。如果尝试将启用 IEEE 802.1x 的端口的模式更改为动态，则端口模式不会更改。

如何配置 VLAN 中继

为了避免中继配置错误，配置连接到设备且不支持 DTP 的接口不转发 DTP 帧，即关闭 DTP。

- 如果不打算将链路作为中继，在接口配置命令 **switchport mode access** 禁用中继。
- 要让不支持 DTP 的设备进行中继，使用接口配置命令 **switchport mode trunk** 和 **switchportnonegotiate** 来配置接口成为中继，但不产生 DTP 帧。

将以太网接口配置为中继端口

配置中继端口（CLI）

因为中继端口发送和接收 VTP 通告，要使用 VTP，必须确保在设备上至少配置了一个中继端口，并且此中继端口连接到另一个设备的中继端口。否则，该设备接收不到任何 VTP 通告。

在开始前

缺省情况下，接口处于二层模式。二层接口的默认模式是 **switchport modedynamic auto**。如果相邻接口支持中继并且配置为允许中继，则链路是二层中继链路；如果接口处于三层模式，则在输入 **switchport** 接口配置命令时，它将成为二层中继。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. switchport mode {dynamic {auto | desirable} | trunk}

5. switchport access vlan*vlan-id*

6. switchport trunk native vlan*vlan-id*

7. end

8. show interfaces *interface-id***switchport**

9. show interfaces *interface-id* **trunk**

10. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>enable`	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 3	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet1/0/2`	将指定接口配置为中继，并进入接口配置模式。
步骤 4	**switchport mode {dynamic {auto | desirable} |** **trunk}** **示例：** `Device(config-if)#switchport mode dynamicdesirable`	将接口配置为二层中继（仅当接口为二层接入端口或隧道端口，或指定中继模式时才需要使用）。 • **dynamic auto**——如果相邻接口是 trunk 模式或 desirable 模式，则接口变为中继链路。此模式为默认模式。 • **dynamic desirable**——如果相邻接口为 trunk，desirable 或 auto 模式，则将接口设置为中继链路。 • **trunk**——将接口设置为永久中继模式，即使相邻接口不是中继接口，也协商要将链路转换为中继链路。
步骤 5	**switchport access vlan***vlan-id* **示例：** `Device(config-if)# switchport access vlan 200`	（可选）指定默认 VLAN，如果接口停止中继，则使用该 VLAN。
步骤 6	**switchport trunk native vlan***vlan-id*	指定 IEEE 802.1Q 中继的本征 VLAN。

	示例: Device(config-if)#**switchport trunk native** **vlan 200**	
步骤 7	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式。
步骤 8	**show interfaces** *interface-id***switchport** **示例：** Device# **show interfaces gigabitethernet1/0/2switchport**	在 *AdministrativeMode* 和 *Administrative Trunking Encapsulation* 字段中显示接口的交换机端口配置。
步骤 9	**show interfaces***interface-id***trunk** **示例：** Device# **show interfaces gigabitethernet1/0/2trunk**	显示接口的中继配置
步骤 10	**copy running-config startup-config** **示例：** Device# **copy running-config startup-config**	（可选）把配置保存在配置文件中。

在中继上定义允许的 VLAN（CLI）

VLAN 1 是所有 Inspur 设备中所有中继端口上的默认 VLAN，以前要求在每个中继链路上始终启用 VLAN 1。可以使用 VLAN 1 最小化特性来禁用任何单个 VLAN 中继链路上的 VLAN 1，以便在 VLAN 1 上不发送或接收用户流量（包括生成树通告）。

总步骤：

1. enable

2. configure terminal

3. interface *interface-id*

4. switchport mode trunk

5. switchport trunk allowed vlan{ *word* **| add | all | except | none | remove}** *vlan-list*

6. end

7. show interfaces *interface-id* **switchport**

8. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** Device>**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式。
步骤 3	**interface** *interface-id* **示例：** Device(config)# **interface gigabitethernet1/0/2**	将指定接口配置为中继，并进入接口配置模式。

步骤 4	**switchport mode trunk** **示例：** `Device(config-if)#switchport mode trunk`	将接口配置为 VLAN 中继端口。
步骤 5	**switchport trunk allowed vlan {** *word* **| add | all |except | none | remove}** *vlan-list* **Example:** `Device(config-if)#switchport trunk allowedvlan remove 2`	（可选）配置中继上允许的 VLAN 列表。 *vlan-list* 参数是从 1 到 4094 的单个 VLAN 号，或由两个 VLAN 号描述的 VLAN 范围，较低的一个在前，用连字符分隔。不要在逗号分隔的 VLAN 参数之间或连字符指定的范围内输入任何空格。 默认情况下允许所有 VLAN
步骤 6	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式。
步骤 7	**show interfaces** *interface-id***switchport** **示例：** `Device# show interfaces gigabitethernet1/0/2switchport`	在显示的 *Trunking VLANs Enabled* 字段中验证条目。
步骤 8	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把配置保存在配置文件中。

更改可修剪列表（CLI）

可修剪列表仅适用于中继端口。每个中继端口都有自己的资格列表。须启用 VTP 修剪才能使此过程生效。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. switchport trunk pruning vlan {add | except | none | remove}*vlan-list [,vlan [,vlan [,,,]]*

5. end

6. show interfaces *interface-id* **switchport**

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>enable`	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 3	**interface** *interface-id*	将指定接口配置为中继，并进入接口

	示例： `Device(config)# interface gigabitethernet1/0/2`	配置模式。
步骤 4	**switchport trunk pruning vlan {add | except | none | remove}***vlan-list [,vlan [,vlan [,,,]]*	配置允许从中继中裁剪的 VLAN 列表。有关使用 **add**，**except**，**none** 和 **remove** 关键字的说明，请参阅此版本的命令参考。 使用逗号分隔不连续的 VLAN ID，不含空格；用连字符指定 ID 范围。有效 ID 为 2 到 1001。 扩展范围 VLAN（VLAN ID 从 1006 到 4094）不能 修剪。 不可修剪的 VLAN 会接收泛洪流量。 允许修剪 VLAN 的默认列表包含 VLAN 2 到 1001。
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式。
步骤 6	**show interfaces** *interface-id***switchport** **示例：** `Device# show interfaces gigabitethernet1/0/2switchport`	在显示的 *Trunking VLANs Enabled* 字段中验证条目。
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把配置保存在配置文件中。

为未标记流量配置本征 VLAN（CLI）

配置了 IEEE 802.1Q 标记的中继端口可以接收已标记和未标记的流量。默认情况下，设备在为端口配置的本征 VLAN 中转发未标记的流量。默认情况下，本征 VLAN 为 VLAN 1。

本征 VLAN 可以分配使用任何 VLAN ID。

如果数据包具有与输出端口本征 VLAN ID 相同的 VLAN ID，则数据包以无标记的方式发送；否则，设备发送带有标记的数据包。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. switchport trunk native vlan*vlan-id*

5. end

6. show interfaces *interface-id***switchport**

7. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入

	示例： `Device>enable`	密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 3	**interface** *interface-id* **示例：** `Device(config)# interface gigabitethernet1/0/2`	指定配置为IEEE 802.1Q中继的接口，并进入接口配置模式。
步骤 4	**switchport trunk native vlan***vlan-id* **Example:** `Device(config-if)#switchport trunk native vlan12`	配置在中继端口上发送和接收未标记流量的VLAN。 对于 *vlan-id*，取值范围为 1 到 4094。
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式。
步骤 6	**show interfaces** *interface-id***switchport** **示例：** `Device# show interfaces gigabitethernet1/0/2switchport`	在显示的 *Trunking VLANs Enabled* 字段中验证条目。
步骤 7	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）把配置保存在配置文件中。

配置中继端口的负载均衡

使用 STP 端口优先级配置负载均衡（CLI）

如果设备是设备堆栈的成员，则必须使用 **spanning-tree[vlan***vlan-id***] cost***cost* interface 配置命令，而不能使用 **spanning-tree[vlan***vlan-id***] port-priority** *priority* 接口配置命令来选择一个接口进入转发状态。将较低的开销分配给希望首先选择的接口，给希望最后选择的接口分配较高的开销。

这些步骤描述如何使用 STP 端口优先级配置具有负载均衡的网络。

总步骤

1. enable

2. configure terminal

3. vtp domain *domain-name*

4. vtp mode server

5. end

6. show vtp status

7. show vlan

8. configure terminal

9. interface *interface-id*

10. switchport mode trunk

11. end

12. show interfaces *interface-id***switchport**

13. 在设备 A 或设备堆中的第二个端口重复上述步骤。

14.在设备 B 上重复上述步骤，配置连接到设备 A 上的中继端口。

15. show vlan

16. configure terminal

17. interface *interface-id*

18. spanning-tree vlan*vlan-range* **port-priority** *priority-value*

19. exit

20. interface *interface-id*

21. spanning-tree vlan*vlan-range* **port-priority** *priority-value*

22. end

23. show running-config

24. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例：** `Device>enable`	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：** `Device# configure terminal`	进入设备 A 的全局配置模式。
步骤 3	**vtp domain** *domain-name* **示例：** `Device(config)#vtp domain workdomain`	配置 VTP 管理域。域名长度为 1 至 32 个字符。
步骤 4	**vtp mode server** **示例：** `Device(config)#vtp mode server`	将设备 A 配置为 VTP 服务器。
步骤 5	**end** **示例：** `Device(config)# end`	返回特权 EXEC 模式。
步骤 6	**show vtp status** **示例：** `Device# show vtp status`	验证设备 A 和设备 B 上的 VTP 配置。在显示中，检查 *VTP Operating Mode* 和 *VTP Domain Name* 字段。
步骤 7	**show vlan** **示例：** `Device# show vlan`	验证设备 A 的数据库上是否存在 VLAN。
步骤 8	**configure terminal** **示例：** `Device# configure terminal`	进入全局配置模式。
步骤 9	**interface** *interface-id* **示例：**	定义要配置为中继的接口，并进入接口配置模式。

	Device(config)# **interface gigabitethernet1/0/1**	
步骤 10	**switchport mode trunk** **示例：** Device(config-if)#**switchport mode trunk**	将端口配置为中继端口
步骤 11	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式。
步骤 12	**show interfaces** *interface-id***switchport** **示例：** Device# **show interfaces gigabitethernet1/0/1 switchport**	验证 VLAN 的配置
步骤 13	在设备A或设备堆中的第二个端口重复上述步骤。	
步骤 14	在设备B上重复上述步骤，配置连接到设备A上的中继端口。	
步骤 15	**show vlan** **示例：** Device# **show vlan**	当中继链路启用时，VTP 会把 VTP 和 VLAN 信息传递给设备 B。该命令验证设备 B 是否已经学到了 VLAN 配置。
步骤 16	**configure terminal** **示例：** Device# **configure terminal**	进入设备 A 的全局配置模式
步骤 17	**interface** *interface-id* **示例：** Device(config)# **interface gigabitethernet1/0/1**	设置接口 STP 端口优先级，进入接口配置模式。
步骤 18	**spanning-tree vlan***vlan-range* **port-priority** *priority-value* **示例：** Device(config-if)# **spanning-tree vlan 8-10 port-priority 16**	为指定的 VLAN 范围分配端口优先级。 输入端口优先级值从 0 到 240。端口优先级值按 16 递增。
步骤 19	**exit** **示例：** Device(config-if)# **exit**	返回全局配置模式。
步骤 20	**interface** *interface-id* **示例：** Device(config)# **interface gigabitethernet1/0/2**	设置接口 STP 端口优先级，进入接口配置模式。
步骤 21	**spanning-tree vlan***vlan-range*	为指定的 VLAN 范围分配端口优先

	port-priority *priority-value* **示例：** `Device(config-if)# spanning-tree vlan 3-6 port-priority 16`	级。 输入从 0 到 240 的端口优先级值。端口优先级值按 16 递增。
步骤 22	**end** **示例：** `Device(config-if)# end`	返回特权 EXEC 模式。
步骤 23	**show running-config** **示例：** `Device# show running-config`	验证输入
步骤 24	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）将输入保存到配置文件中。

使用 STP 路径开销配置网络负载均衡（CLI）

这些步骤描述如何使用 STP 路径开销配置具有负载均衡的网络。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. switchport mode trunk

5. exit

6. 在设备 A 或设备 A 堆的第二个接口上重复步骤 2 到 4。

7. end

8. show running-config

9. show vlan

10. configure terminal

11. interface *interface-id*

12. spanning-tree vlan*vlan-range* **cost** *cost-value*

13. end

14.重复步骤 9-13，配置设备 A 上另外的中继接口，并设置 VLAN8、9、10 的生成树路径开销为 30。

15. exit

16. show running-config

17. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例：** `Device>enable`	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例：**	进入全局配置模式。

	Device# **configure terminal**	
步骤 3	**interface** *interface-id* **示例：** Device(config)# **interface gigabitethernet1/0/1**	将接口配置为中继，进入接口配置模式。
步骤 4	**switchport mode trunk** **示例：** Device(config-if)# **switchport mode trunk**	将端口配置为中继端口。
步骤 5	**exit** **示例：** Device(config-if)# **exit**	返回全局配置模式
步骤 6	在设备A或设备 A堆的第二个接口上重复步骤2到4。	
步骤 7	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式
步骤 8	**show running-config** **示例：** Device# **show running-config**	验证输入。在显示中确认接口已配置为中继端口。
步骤 9	**show vlan** **示例：** Device# **show vlan**	当中继链路启用时，设备 A 从其他设备接收 VTP 信息。此命令验证设备 A 已获知 VLAN 配置。
步骤 10	**configure terminal** **示例：** Device# **configure terminal**	进入全局配置模式。
步骤 11	**interface** *interface-id* **示例：** Device(config)# **interface gigabitethernet1/0/1**	定义要设置 STP 开销的接口，并进入接口配置模式。
步骤 12	**spanning-tree vlan***vlan-range* **cost** *cost-value* **Example:** Device(config-if)# **spanning-tree vlan 2-4 cost 30**	设置 VLAN2-4 的生成树路径开销为 30。
步骤 13	**end** **示例：** Device(config)# **end**	返回特权 EXEC 模式。
步骤 14	重复步骤9-13，配置设备A上另外的中继接口，并设置VLAN8、9、10的生成树路径开销为30。	
步骤 15	**exit** **示例：** Device(config)# **exit**	返回特权 EXEC 模式

步骤 16	**show running-config** **示例：** `Device# show running-config`	验证输入。在显示中验证两个中继接口的路径开销设置是否正确。
步骤 17	**copy running-config startup-config** **示例：** `Device# copy running-config startup-config`	（可选）将输入保存到配置文件中。

接下来做什么？

配置 VLAN 中继后，可以接着配置如下内容：

- VLAN
- VLAN 组
- 语音 VLAN

其他参考资料

相关

相关主题	文档标题
有关本章中使用的命令的完整语法和使用信息。	*VLAN Command Reference (Inspur 6650 Switches)* *ayer 2/3 Command Reference (Inspur 6650 Switches)*
生成树（STP）	*Network ManagementCommand Reference (Inspur6650 Switches)* *NetworkManagementConfigurationGuide(Inspur6650Switchs)*

错误信息解释

描述	链接
为帮助管理员搜索并解决该版本中的系统错误信息，管理员可使用错误信息解释工具。	http://www.icntnetworks.com

标准和 RFC

标准/RFC	标题
RFC 1573	*Evolution of the Interfaces Group of MIB-II*
RFC 1757	*Remote Network Monitoring Management*
RFC 2021	*SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2*

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现	http://www.icntnetworks.com

场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	

VLAN 的特征历史与信息

版本	修改

配置语音 VLAN

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用 Inspur 特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问 Inspur 特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

语音 VLAN 的前提

以下是语音 VLAN 的前提：

- 仅在设备接入端口上支持语音 VLAN 配置；中继端口不支持语音 VLAN 配置。

注释： 中继端口可以承载任意数量的语音 VLAN，类似于常规 VLAN。中继端口不支持语音 VLAN 的配置。

- 在启用语音 VLAN 之前，请输入 **trust device inspur-phone** 接口配置命令在设备上启用 QoS。如果使用自动 QoS 功能，则会自动配置这些设置。
- 您必须在连接到 Inspur IP 电话的设备端口上启用 CDP 才能将配置发送到电话（CDP 默认全局在所有设备接口上启用）。

语音 VLAN 的限制条件

无法在语音 VLAN 中配置静态安全 MAC 地址。

关于语音 VLAN 的信息

语音 VLAN

语音 VLAN 特性使接入端口能够承载来自 IP 电话的 IP 语音流量。当设备连接到 Inspur7960 IP 电话时，电话会发送具有三层 IP 优先级和二层服务等级（class of service，CoS）值的语音流量，默认情况下均设置为 5。因为如果数据不均匀地发送，IP 电话呼叫的话音质量可能变差，所以设备支持基于 IEEE 802.1p CoS 的服务质量（quality of service，QoS）。QoS 使用分类和调度，通过可预测的方式从设备发送网络流量。

Inspur IP 电话语音流量

管理员可以配置一个连接了 Inspur IP 电话的接入端口，将一个 VLAN 用于语音流量，另一个 VLAN 用于来自连接到该电话的设备的数据流量。管理员可配置设备上的接入端口发送 Inspur 发现协议（Inspur Discovery Protocol ， CDP）数据包，指示连接的电话通过以下任何方式向设备发送语音流量：

- 在标记有二层 CoS 优先级值的语音 VLAN 中
- 在标记有二层 CoS 优先级值的接入 VLAN 中
- 在接入 VLAN 中，无标记（没有二层 CoS 优先级值）

注释：在所有配置中，语音流量都携带三层 IP 优先级值（对于语音流量，默认值为 5；对于语音控制流量，默认值为 3）。

Inspur IP 电话数据流量

设备还可以处理带标记的数据流量（IEEE 802.1Q 或 IEEE 802.1p 帧类型的流量），该数据流量来自连接到 Inspur IP 电话上的接入端口的设备。管理员可配置设备上的二层接入端口发送 CDP 数据包，指示连接的电话按照下列模式之一配置电话接入端口：

- 在可信模式下，通过 Inspur IP 电话上的接入端口接收的所有流量流经电话时都不会改变。
- 在不可信模式下，通过 Inspur IP 电话上的接入端口接收的 IEEE 802.1Q 或 IEEE 802.1p 帧中的所有流量都会接收配置的二层 CoS 值。默认的二层 CoS 值为 0。不可信模式是默认模式。

注释： 不管电话上接入端口的可信状态如何，来自连接到 Inspur IP 电话的设备的未标记流量流经电话时不会改变。

语音 VLAN 配置指南

- 因为 Inspur 7960 IP 电话还支持与 PC 或其他设备相连，所以连接到 Inspur IP 电话的设备端口可以携带混合流量。管理员可以配置端口，决定 Inspur IP 电话如何传输语音流量和数据流量。
- 语音 VLAN 应该在设备上存在并处于活动状态，以便 IP 电话在语音 VLAN 上正常通信。使用 **show vlan** 特权 EXEC 命令查看是否存在语音 VLAN（在显示输出中列出）。如果未列出 VLAN，请创建语音 VLAN。
- 如果 Inspur 预标准和 IEEE 802.3af 兼容的供电设备没有交流电源供电，以太网供电（Power over Ethernet，PoE）设备能够自动向这些设备提供电源。
- 当配置了语音 VLAN 时，Port Fast 特性会自动启用。当禁用语音 VLAN 时，Port Fast 特性不会自动禁用。
- 如果 Inspur IP 电话和连接到电话的设备在同一个 VLAN 中，它们必须在同一个 IP 子网中。这些条件表明它们在同一 VLAN 中：
  - 它们都使用 IEEE 802.1p 或无标记数据帧。
  - Inspur IP 电话使用 IEEE 802.1p 帧，设备使用无标记帧。
  - Inspur IP 电话使用无标记帧，设备使用 IEEE 802.1p 帧。
  - Inspur IP 电话使用 IEEE 802.1Q 帧，语音 VLAN 与接入 VLAN 相同。
- 因为在同一个子网的流量不会进行路由，所以当 Inspur IP 电话和连接到电话的设备在同一个 VLAN 和子网但使用不同的帧类型时，它们不能通信（路由将消除帧类型的差异）。
- 语音 VLAN 端口也会是以下端口类型：
  - 动态接入端口。
  - IEEE 802.1x 认证端口。

  注释： 如果在配置了语音 VLAN 并且连接了 Inspur IP 电话的接入端口上启用 IEEE 802.1x，则该电话会与设备的失去连接达 30 秒。

  - 受保护端口。
  - SPAN 或 RSPAN 会话的源或目标端口。
  - 安全端口。

注释： 在配置了语音 VLAN 的接口上启用端口安全时，必须将端口上允许的最大安全地址数设置为接入 VLAN 上允许的最大安全地址数加 2。当端口连接到 Inspur IP 电话时，电话最多需要两个 MAC 地址。电话地址会在语音 VLAN 上学习到，也可能在接入 VLAN 上学习到。将 PC 连接到电话需要额外的 MAC 地址。

如何配置语音 VLAN

配置 Inspur IP 电话语音流量（CLI）

管理员可以配置一个连接到 Inspur IP 电话的端口将 CDP 包发送到电话，以配置电话发送语音流量的方式。电话可以在 IEEE 802.1Q 帧中携带具有二层 CoS 值的指定语音 VLAN 的语音流量。它可以使用 IEEE 802.1p 优先级标记为语音流量提供更高的优先级，并通过本征（接

入）VLAN 转发所有语音流量。Inspur IP 电话还可以发送无标记的语音流量或使用自己的配置在接入 VLAN 中发送语音流量。在所有配置中，语音流量都携带三层 IP 优先级值（默认值为 5）。

总步骤

1. configure terminal

2. interface *interface-id*

3. trust device inspur-phone

4. switchport voice vlan {*vlan-id* | **dot1p** | **none** | **untagged**}

5. end

6. 使用以下命令之一：

- **show interfaces** *interface-id* **switchport**
- **show running-config interface** *interface-id*

7. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 2	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet1/0/1**	指定连接到电话的接口，并进入接口配置模式。
步骤 3	**trust device inspur-phone** **示例:** `Device(config-if)#` **trust-device inspurphone**	配置接口信任 Inspur IP 电话的入向流量包。
步骤 4	**switchport voice vlan** {*vlan-id* | **dot1p** | **none** | **untagged**} **示例:** `Device(config-if)#` **switchport voice vlan dot1p**	配置语音 VLAN。 • *vlan-id*——配置电话通过指定的 VLAN 转发所有语音流量。默认情况下，Inspur IP 电话以 IEEE 802.1Q 优先级 5 转发语音流量。有效的 VLAN ID 为 1 到 4094。 • **dot1p——**配置设备接受标记为 VLAN ID 0（本地 VLAN）的语音和数据 IEEE 802.1p 优先级帧。默认情况下，设备会丢弃所有标记为 VLAN 0 的语音和数据流量。如果配置为 802.1p，Inspur IP 电话将使用 IEEE 802.1p 优先级 5 转发流量。

		• **none——**允许电话使用自己的配置发送无标记的语音流量。 • **untagged——**配置电话以发送无标记语音流量。
步骤 5	**end** **示例:** `Device(config-if)#` **end**	返回特权 EXEC 模式。
步骤 6	使用以下命令之一： • **show interfaces** *interface-id* **switchport** • **show running-config interface** *interface-id* **示例:** `Device#` **show interfaces gigabitethernet1/0/1 switchport** 或 `Device#` **show running-config interface gigabitethernet1/0/1**	验证语音 VLAN 条目或 QoS 和语音 VLAN 条目。
步骤 7	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）将条目保存在设备启动配置文件中。

配置入向数据帧的优先级（CLI）

管理员可以将 PC 或其他数据设备连接到 Inspur IP 电话端口。要处理带标记的数据流量（在 IEEE 802.1Q 或 IEEE 802.1p 帧中），可以设备配置发送 CDP 数据包，以指示电话如何发送连接到 Inspur IP 电话接入端口的设备的数据包。PC 可以生成带有指定 CoS 值的数据包。可以配置电话不更改（信任）或覆盖（不信任）从连接的设备到达电话端口的帧的优先级。

按照以下步骤设置从 Inspur IP 电话上的非语音端口接收的数据流量的优先级：

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. switchport priority extend {**cos** *value* | **trust**}

5. end

6. show interfaces *interface-id* **switchport**

7. copy running-config startup-config

具体步骤

	命令或操作	**目的**
步骤 1	**enable**	进入特权 EXEC 模式。在提示时输入密码。

	示例: `Device>`**enable**	
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet1/0/1**	指定连接到 Inspur IP 电话的接口，并进入接口配置模式。
步骤 4	**switchport priority extend** {**cos** *value* | **trust**} **示例:** `Device(config-if)#` **switchport priority extend trust**	设置从 Inspur IP 电话接入端口接收的数据流量优先级： • **cos** *value*——配置电话以覆盖从 PC 或具有指定 CoS 值的连接设备接收的优先级。该值为 0 到 7 之间的数字，7 为最高优先级。默认优先级为 **cos**0。 • **trust**——配置电话接入端口以信任从 PC 或连接的设备接收的优先级。
步骤 5	**end** **示例:** `Device(config-if)#` **end**	返回特权 EXEC 模式。
步骤 6	**show interfaces** *interface-id* **switchport** **示例:** `Device#` **show interfaces gigabitethernet1/0/1 switchport**	验证条目。
步骤 7	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）将条目保存在设备启动配置文件中。

监控语音 VLAN

要显示接口的语音 VLAN 配置，请使用 **show interfaces** *interface-id* **switchport** 特权 EXEC 命令。

接下来做什么？

配置了语音 VLAN 后可以做以下配置：

- VLAN
- VLAN 组（VLANgroups）
- VLAN 中继（VLAN Trunking）

- VTP

其他参考资料

相关文档

相关主题	文档题目
有关本章中使用的命令的完整语法和使用信息。	*VLAN 命令参考（Inspur 6650 交换机）* *二/三层命令参考（Inspur 6650 交换机）*
其他配置命令及过程。	*Inspur INOS 的 LAN 交换配置指南（Inspur 6650 交换机）* *二/三层配置指南（Inspur 6650 交换机）*
平台无关的配置信息。	*Inspur INOS 的基于身份的网络服务配置指南（Inspur 6650 交换机）*

错误消息解码器

描述	链接
为了帮助您在本版本中研究和解决系统错误消息，请使用错误消息解码器工具（Error Message Decoder tool）。	http://www.icntnetworks.com

标准和 RFC

标准/RFC	题目
RFC 1573	MIB-II 接口组的演进
RFC 1757	远程网络监控管理
RFC 2021	使用 SMIv2 传输控制协议的 SNMPv2 管理信息库

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	http://www.icntnetworks.com

语音 VLAN 的特性历史与信息

版本	修改
Inspur INOS 11.3.1	引入了此功能。

配置私有 VLAN

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看错误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如需查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用 Inspur 特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访问 Inspur 特性导航（Inspur Feature Navigator），可以访问 http://www.icntnetworks.com。用户不需要在 icntnetworks.com 注册账户就可以使用这个导航系统。

私有 VLAN 的前提

VTP 1、2 和 3 的透明模式支持使用私有 VLAN。VTP 3 的服务器模式也支持私有 VLAN。
在设备上配置私有 VLAN 时，请始终使用默认的交换数据库管理（Switch Database Management，SDM）模板来平衡单播路由和二层条目之间的系统资源。如果配置了另一个 SDM 模板，请使用 **sdm prefer default** 全局命令设置默认模板。

私有 VLAN 的限制

注释：在某些情况下，虽然配置被接受且没有错误消息，但命令没有效果。

- 不要在具有私有 VLAN 的设备上配置回退桥接。
- 不要将远程 SPAN（RSPAN）VLAN 配置为私有 VLAN 的主 VLAN 或辅助 VLAN。
- 不要在配置了这些其他特性的接口上配置私有 VLAN 的端口：
  - 动态接入端口 VLAN 成员
  - 动态中继协议（Dynamic Trunking Protocol，DTP）
  - IPv6 安全组（Security Group，SG）
  - 端口聚合协议（Port Aggregation Protocol，PAgP）
  - 链路聚合控制协议（Link Aggregation Control Protocol，LACP）
  - 组播 VLAN 注册（Multicast VLAN Registration，MVR）
  - 语音 VLAN
  - Web 缓存通信协议（Web Cache Communication Protocol，WCCP）
- 您可以在私有 VLAN 的端口上配置基于 IEEE 802.1x 端口的身份验证，但不要在私有 VLAN 端口上配置使用端口安全、语音 VLAN 或基于用户 ACL 的 802.1x。
- 私有 VLAN 主机或混杂端口不能是 SPAN 目标端口。如果把 SPAN 目标端口配置为私有 VLAN 的端口，则该端口将变为非活跃状态。
- 如果在主 VLAN 中的混杂端口上配置静态 MAC 地址，则无需向所有关联的辅助 VLAN 添

加相同的静态地址。类似地，如果在辅助 VLAN 中的主机端口上配置静态 MAC 地址，则无需向关联的主 VLAN 添加相同的静态 MAC 地址。此外，从私有 VLAN 的端口删除静态 MAC 地址时，不必从私有 VLAN 删除所有配置的 MAC 地址实例。

注释：在私有 VLAN 的辅助 VLAN 中学习的动态 MAC 地址将复制到主 VLAN。所有 MAC 条目都是在辅助 VLAN 上学习的，即使从主 VLAN 进入的流量也一样。如果在主 VLAN 中动态学习了一个 MAC 地址，它将不会复制到相关的辅助 VLAN 中。

- 只能为主 VLAN 配置三层 VLAN 接口（SVI）。

关于私有 VLAN 的信息

私有 VLAN 域

私有 VLAN 特性解决了服务提供商在使用 VLAN 时面临的两个问题：

- 运行 IP Base 或 IP Services 镜像时，设备最多支持 4094 个活跃 VLAN。如果服务提供商为每个客户分配一个 VLAN，这将限制服务提供商可以支持的客户数。
- 要启用 IP 路由，每个 VLAN 都会被分配一个子网地址空间或一个地址块，这可能会浪费未使用的 IP 地址，并造成 IP 地址管理问题。

使用私有 VLAN 解决了可扩展性问题并为服务提供商提供 IP 地址管理优势，还为客户提供二层安全性。私有 VLAN 将常规 VLAN 域划分为子域。一个子域由一对 VLAN 表示：*主（primary）* VLAN 和*辅助（secondary）*VLAN。私有 VLAN 可以有多个 VLAN 对，每个子域有一对。私有 VLAN 中的所有 VLAN 对共享相同的主 VLAN。辅助 VLAN ID 区分一个子域与另一个子域。

图 140：私有 VLAN 域

Private VLAN domain	私有 VLAN 域
Primary VLAN	主 VLAN
Subdomain	子域
Secondary community VLAN	辅助团体 VLAN
Secondary isolated VLAN	辅助隔离 VLAN

辅助 VLAN

辅助 VLAN 包含两种类型：

- 隔离 VLAN——隔离 VLAN 内的端口不能在二层级互相通信。
- 团体 VLAN——团体 VLAN 内的端口可以相互通信，但不能与其他团体中的端口在二层级进行通信。

私有 VLAN 端口

私有 VLAN 在同一私有 VLAN 内的端口之间提供二层隔离。私有 VLAN 端口是以下访问端口类型之一：

- 混杂——混杂端口属于主 VLAN，并且可以和所有接口通信，包括属于与主 VLAN 关联的辅助 VLAN 的团体和隔离的主机端口。

- 隔离——隔离端口是属于隔离辅助 VLAN 的主机端口。它与同一私有 VLAN 中的其他端口具有完全的二层隔离，但混杂端口除外。私有 VLAN 阻止所有流量到隔离端口，除了来自混杂端口的流量。从隔离端口接收的流量仅转发到混杂端口。
- 团体——团体端口是属于团体辅助 VLAN 的主机端口。团体端口与同一个团体 VLAN 中的其他端口以及混杂端口通信。这些接口在二层与其他团体中的所有其他接口和其私有 VLAN 内的隔离端口隔离。

注释：中继端口承载来自常规 VLAN 以及主 VLAN、隔离 VALN 和团体 VLAN 的流量。

主 VLAN 和辅助 VLAN 有以下特征：

- 主 VLAN——一个私有 VLAN 仅有一个主 VALN。私有 VLAN 中的每个端口都属于主 VLAN。主 VLAN 承载从混杂端口到（隔离和团体）主机端口和其他混杂端口的单向流量。
- 隔离 VLAN——私有 VLAN 只有一个隔离 VLAN。隔离 VLAN 是辅助 VLAN，承载从主机到混杂端口和网关的上行单向流量。
- 团体 VLAN——团体 VLAN 是辅助 VLAN，承载从团体端口到混杂端口网关以及同一团体中其他主机端口的上行流量。可以在私有 VLAN 中配置多个团体 VLAN。

混杂端口只能服务于一个主 VLAN、一个隔离 VLAN 和多个团体 VLAN。三层网关通常通过混杂端口连接到设备。使用混杂端口，您可以连接各种设备作为到私有 VLAN 的接入点。例如，您可以使用混杂端口从管理工作站监视或备份所有私有 VLAN 服务器。

网络中的私有 VLAN

在交换环境中，可以为单独终端站或公共终端站组的分配单个私有 VLAN 和相关的 IP 子网。要和私有 VLAN 外部通信，终端站需要只与默认网关通信。

您可以使用私有 VLAN 通过以下方式控制对终端站的访问：

- 将连接到终端站的所选接口配置为隔离端口，以防止在二层进行任何通信。例如，如果终端站是服务器，则此配置会阻止服务器之间的二层通信。
- 将连接到默认网关和选定终端站（例如备份服务器）的接口配置为混杂端口，以允许所有终端站访问默认网关。

您可以通过将主 VLAN、隔离 VLAN 和团体 VLAN 中继到其他支持私有 VLAN 的设备，以在多个设备上扩展私有 VLAN。为了维护私有 VLAN 配置的安全性，并避免把配置的私有 VLAN 用作其他用途，请在所有中间设备（包括没有私有 VLAN 端口的设备）上配置私有 VLAN。

私有 VLAN 的 IP 编址方案

为每个客户分配一个单独的 VLAN 会创建一个低效的 IP 编址方案：

- 为客户 VLAN 分配一块地址可能会导致 IP 地址闲置。
- 如果 VLAN 中的设备数量增加，则已分配的地址数量可能不足以容纳它们。

通过使用私有 VLAN 减少了这些问题，其中私有 VLAN 中的所有成员共享分配给主 VLAN 的公共地址空间。主机连接到辅助 VLAN，DHCP 服务器从分配给主 VLAN 的地址块中给他们分配 IP 地址。在同一主 VLAN 中，后续 IP 地址可以分配给不同辅助 VLAN 中的客户设备。添加新设备时，DHCP 服务器会从大型子网地址池中为其分配下一个可用地址。

多设备上的私有 VLAN

与常规 VLAN 一样，私有 VLAN 可以跨越多个设备。中继端口将主 VLAN 和辅助 VLAN 传送到相邻设备。中继端口把私有 VLAN 当作任何其他的 VLAN 一样对待。

在多个设备上的私有 VLAN 的一个特性是来自设备 A 隔离端口的流量不能到达设备 B 上的隔离端口。

图 141：多台交换机上的私有 VLAN

Trunk ports	中继端口
Switch A	交换机 A
Switch B	交换机 B
Carries VLAN 100,201,and 202 traffic	承载 VLAN 100、201 和 202 的流量
VLAN 100	VLAN 100
VLAN 201	VLAN 201
VLAN 202	VLAN 202
VLAN 100 = Primary VLAN	VLAN 100 = 主 VLAN
VLAN 201 = Secondary isolated VLAN	VLAN 201 = 辅助隔离 VLAN
VLAN 202 = Secondary community VLAN	VLAN 202 = 辅助团体 VLAN

VTP 1、2 和 3 的透明模式支持私有 VLAN。VTP 3 的服务器模式也支持私有 VLAN。如果使用 VTP 3 设置服务器客户端，则在服务器上配置的私有 VLAN 应反映在客户端上。

私有 VLAN 与其他特性的相互作用

私有 VLAN 与单播、广播和组播流量

在常规 VLAN 中，同一 VLAN 内的设备可以在二层互相通信，但连接到不同 VLAN 内接口的设备必须在三层进行通信。在私有 VLAN 中，混杂端口是主 VLAN 的成员，而主机端口则属于辅助 VLAN。由于辅助 VLAN 与主 VLAN 关联，因此这些 VLAN 的成员可以在二层上相互通信。

在常规 VLAN 中，广播被转发到该 VLAN 中的所有端口。私有 VLAN 广播的转发取决于发该送广播的端口：

- 隔离端口仅向混杂端口或中继端口发送广播。
- 团体端口向所有混杂端口、中继端口和同一个团体 VLAN 中的端口发送广播。
- 混杂端口向私有 VLAN 中的所有端口（其他混杂端口、中继端口、隔离端口和团体端口）发送广播。

组播流量在私有 VLAN 的边界和单个团体 VLAN 内路由或桥接。

组播流量不在同一隔离 VLAN 中的端口之间或不同辅助 VLAN 中的端口之间转发。

私有 VLAN 组播转发支持如下功能：

- 发送方可以在 VLAN 域外，接收方可以在 VLAN 域内。
- 发送方可以在 VLAN 域内，接收方可以在 VLAN 域外。
- 发送方和接收方可以都在同一团体 VLAN 中。

私有 VLAN 及 SVI

在三层设备中，一个设备虚拟接口（SVI）代表 VLAN 的三层接口。三层设备只能通过主 VLAN 而非辅助 VLAN 与私有 VLAN 通信。只应给主 VLAN 配置三层 VLAN 接口（SVI）。不能为辅助

VLAN 配置三层 VLAN 接口。当 VLAN 配置为辅助 VLAN 时，辅助 VLAN 的 SVI 是无效的。

- 如果尝试把含有活跃 SVI 的 VLAN 配置为辅助 VLAN，在禁用 SVI 之前不允许进行此配置。
- 如果尝试在配置为辅助 VLAN 的 VLAN 上创建 SVI，并且辅助 VLAN 已在三层映射，则不会创建 SVI，且会返回错误。如果 SVI 未映射到第 3 层，则能创建 SVI，但它会自动关闭。

当主 VLAN 与辅助 VLAN 关联并映射到辅助 VLAN 时，主 VLAN 上的任何配置都会传送到辅助 VLAN 的 SVI。例如，如果将一个 IP 子网分配给主 VLAN 的 SVI，则此子网是私有 VLAN 的整个 IP 子网地址。

私有 VLAN 和设备堆栈

私有 VLAN 可以在设备堆栈内运行，私有 VLAN 的端口可以位于不同的堆栈成员上。但是，以下对堆栈的更改可能会影响私有 VLAN 的操作：

- 如果堆栈只包含一个私有 VLAN 的混杂端口，并且包含该端口的堆栈成员已从堆栈中删除，该私有 VLAN 中的主机端口会在私有 VLAN 之外失去的连通性。
- 如果一个包含该堆栈中唯一私有 VLAN 混杂端口的堆栈 master 出现故障或离开该堆栈，并且选举出了新的堆栈 master，则在旧堆栈 master 上具有混杂端口的私有 VLAN 中的主机端口会失去私有 VLAN 之外的连通性。
- 如果两个堆栈合并，优胜堆栈上的私有 VLAN 不受影响，但失败设备上的私有 VLAN 配置会在设备重新启动时会丢失。

具有动态 MAC 地址的私有 VLAN

在辅助 VLAN 中学习的 MAC 地址会被复制到主 VLAN，反之则不然。这节省了硬件二层 CAM 空间。主 VLAN 始终被用来进行两个方向上的转发查找。

如果需要，在私有 VLAN 的主 VLAN 中学习的动态 MAC 地址将复制到辅助 VLAN 中。例如，如果在辅助 VLAN 上动态接收一个 MAC 地址，则该 MAC 地址将被当作主 VLAN 的一部分。在隔离 VLAN 的情况下，同一个 MAC 的阻塞条目会在 MAC 地址表中添加给辅助 VLAN。因此，在辅助域中的主机端口上学习的 MAC 将作为阻塞类型条目安装。即使流量从主 VLAN 进入，所有 MAC 条目都在辅助 VLAN 上学习。

然而，如果 MAC 地址是在主 VLAN 中动态学习的，该 MAC 地址将不会被复制到相关联的辅助 VLAN 中。

具有静态 MAC 地址的私有 VLAN

与传统模型相比，用户无需复制私有 VLAN 主机的静态 MAC 地址 CLI。

示例：

- 在传统模型中，如果用户配置了静态 MAC 地址，则也需要在关联 VLAN 中添加相同的静态 MAC 地址。例如，用户在 VLAN 101 的 1/0/1 端口上配置了 MAC 地址 A（其中 VLAN 101 是辅助 VLAN，VLAN 100 是主 VLAN），则用户必须进行如下配置：

  mac-address static A vlan 101 interface G1/0/1

  mac-address static A vlan 100 interface G1/0/1
- 在这个设备中，用户无需将 MAC 地址复制到相相关联的 VLAN 中。对于上面的示例，用户只需要进行如下配置：

  mac-address static A vlan 101 interface G1/0/1

私有 VLAN 与 VACL / QOS 的相互作用

当与其他平台中的“单向”VLAN 相比，私有 VLAN 在此设备的情况下是双向的。

在二层转发查找后，正确的出向 VLAN 映射产生，所有基于出向 VLAN 的特性处理都在出向 VLAN 环境中进行。

当二层中的数据帧帧在私有 VLAN 内转发时，VLAN 映射会被应用在在入向侧和出向侧。当数据帧从私有 VLAN 内部被路由到外部端口时，在入向侧应用私有 VLAN 的映射。类似地，

当帧从外部端口被路由到私有 VLAN 时，在出向侧应用私有 VLAN。这适用于桥接和被路由流量。

桥接：

- 对于从辅助 VLAN 到主 VLAN 的上行流量，辅助 VLAN 的 MAP 应用在入向侧，主 VLAN 的 MAP 应用在出向侧。
- 对于从主 VLAN 到辅助 VLAN 的下行流量，主 VLAN 的 MAP 应用在入向，辅助 VLAN 的 MAP 应用在出方向。

路由：

如果有两个私有 VLAN 域——PV1（sec1，prim1）和 PV2（sec2，prim2）。当帧从 PV1 路由到 PV2：

- 在入端口应用 sec1 的 MAP 和 prim1 的 L3 ACL。
- 在出端口应用 sec2 的 MAP 和 prim2 的 L3 ACL。

对于从独立主机端口到混杂端口的上行或下行的数据包，在入方向应用隔离 VLAN 的 VACL，在出方向应用主 VLAN 的 VACL。这允许用户在同一主 VLAN 域中为不同的辅助 VLAN 配置不同的 VACL。

注释： 不需要使用双向团体 VLAN，因为此设备上的私有 VLAN 始终是双向的。

私有 VLAN 以及 HA 支持

PVLAN 会和高可用性（High Availability，HA）特性无缝协作。切换之前 master 上存在的私有 VLAN 在切换后应该相同（新 master 在 INOS 和 FED 上具有与旧 master 类似的 PVLAN 配置）。

私有 VLAN 配置指南

私有 VLAN 的默认配置

无私有 VLAN 配置。

辅助 VLAN 及主 VLAN 的配置

请按照以下指南配置私有 VLAN：

- VTP 1、2 和 3 的透明模式支持私有 VLAN。如果设备运行的是版本 1 或 2 的 VTP，则必须将 VTP 设置为透明模式。配置私有 VLAN 后，不应把 VTP 模式更改为客户端或服务器。VTP 版本 3 在所有模式下都支持私有 VLAN。
- 使用 VTP 版本 1 或 2 时，在配置私有 VLAN 后，请使用 **copy running-configstartup config** 特权 EXEC 命令在设备启动配置文件中保存 VTP 透明模式配置和私有 VLAN 配置。否则，如果设备重置，它默认会成为 VTP 服务器模式，不支持私有 VLAN 配置。 VTP 版本 3 支持私有 VLAN。
- 版本 1 和 2 的 VTP 不传播私有 VLAN 的配置。必须在每个要设置私有 VLAN 端口的设备上配置私有 VLAN，除非设备运行可传播私有 VLAN 的 VTP 版本 3。
- 不能将 VLAN 1 或 VLAN 1002 至 1005 配置为主 VLAN 或辅助 VLAN。扩展 VLAN（VLAN 的 ID 为 1006 到 4094）可以属于私有 VLAN。
- 主 VLAN 可以有一个隔离的 VLAN 和与其关联的多个团体 VLAN。隔离或团体 VLAN 只能有一个与其关联的主 VLAN。
- 虽然私有 VLAN 包含多个 VLAN，但整个私有 VLAN 只运行一个生成树协议（Spanning Tree Protocol，STP）实例。当辅助 VLAN 与主 VLAN 关联时，主 VLAN 的 STP 参数会传播到辅助 VLAN。
- 从 TFTP 服务器复制 PVLAN 配置并将其应用于运行配置时，将不会建立 PVLAN 关联。需

要检查并确保主 VLAN 与所有辅助 VLAN 相关联。

也可以用 **configure replace flash:config_file force**，而不使用 **copy flash:config_filerunning-config**.。

- 您可以在私有 VLAN 上启用 DHCP 侦听。当在主 VLAN 上启用 DHCP 侦听时，它会传播侦听信息到辅助 VLAN。如果在辅助 VLAN 上配置 DHCP 侦听，且已配置了主 VLAN，则配置不会生效。
- 在私有 VLAN 端口上启用 IP 源地址防护时，必须在主 VLAN 上启用 DHCP 侦听功能。
- 建议裁剪在设备的中继上不承载流量的私有 VLAN。
- 可以对主 VLAN、隔离 VLAN 和团体 VLAN 应用不同的服务质量（quality of service，QoS）配置。
- 注意粘性 ARP 的以下事项：
  - 粘性 ARP 条目是在 SVI 和三层接口上学习的。这些条目不会过期。
  - **ip sticky-arp** 全局配置命令仅在属于私有 VLAN 的 SVI 上支持。
  - **ip sticky-arp** 接口配置命令仅支持：
    - 三层接口
    - 属于常规 VLAN 的 SVI
    - 属于私有 VLAN 的 SVI

    有关使用 **ip sticky-arp** 全局配置和 **ip sticky-arpinterface** 接口配置命令的更多信息，请参阅此版本的命令参考。
- 您可以在主 VLAN 和辅助 VLAN 上配置 VLAN 映射。然而，建议在私有 VLAN 的主 VLAN 和辅助 VLAN 上配置相同的 VLAN 映射。
- PVLAN 是双向的。它们可以应用在入向和出向。

  当二层中的数据帧在私有 VLAN 内转发时，VLAN 映射会应用在在入向和出向端。当数据帧从私有 VLAN 内部路由到外部端口时，私有 VLAN 映射会应用在在入向侧。类似地，当数据帧从外部端口路由到私有 VLAN 时，私有 VLAN 映射会应用在在出向侧。

桥接：

- 对于从辅助 VLAN 到主 VLAN 的上行流量，在入向侧应用辅助 VLAN 的 MAP，在出向侧应用主 VLAN 的 MAP。
- 对于从主 VLAN 到辅助 VLAN 的下行流量，在入方向应用主 VLAN 的 MAP，在出方向应用辅助 VLAN 的 MAP。

路由：

如果有两个私有 VLAN 域——PV1（sec1，prim1）和 PV2（sec2，prim2）。当数据帧从 PV1 路由到 PV2：

- 在入端口应用 sec1 的 MAP 和 prim1 的 L3 ACL。
- 在出端口应用 sec1 的 MAP 和 prim2 的 L3 ACL。
- 对于从独立主机端口到混杂端口的上行或下行的分组，在入方向应用隔离 VLAN 的 VACL，在出方向应用主 VLAN 的 VACL。这允许用户在同一主 VLAN 域中为不同的辅助 VLAN 配置不同的 VACL。

要过滤私有 VLAN 的特定 IP 流量，应该将 VLAN 映射同时应用于主 VLAN 和辅助 VLAN。

- 可以仅在主 VLAN 的 SVI 上应用路由器 ACL。该 ACL 应用于主 VALN 和辅助 VLAN 的三层流量。
- 虽然私有 VLAN 在二层提供主机隔离，但主机可以在三层相互通信。
- 私有 VLAN 支持交换端口分析器（Switched Port Analyzer，PAN）的如下特性：
  - 可以将私有 VLAN 端口配置为 SPAN 源端口。

- 可以在主 VLAN、隔离 VLAN 和团体 VLAN 上使用基于 VLAN 的 SPAN（VSPAN），或者仅使用一个 VLAN 上的 SPAN 来分别监视出向或入向的流量。

私有 VLAN 端口配置

请按照以下指南配置私有 VLAN 端口：

- 仅使用私有 VLAN 配置命令将端口分配给主 VLAN、隔离 VLAN 或团体 VLAN。当某一 VLAN 是私有 VLAN 配置的一部分时，分配给配置为主 VLAN、隔离 VLAN 或团体 VLAN 的二层接入端口是非活动状态。二层中继端口保持在 STP 转发状态。
- 不要将属于 PAgP 或 LACP EtherChannel 的端口配置为私有 VLAN 端口。虽然端口是私有 VLAN 配置的一部分，但它的任何 EtherChannel 配置都是非活动状态的。
- 在隔离和团体主机端口上启用 Port Fast 和 BPDU 防护，以防止由于配置错误导致的 STP 环路，并加速 STP 收敛。当启用时，STP 将 BPDU 防护功能应用于所有配置 Port Fast 的二层 LAN 端口。不要在混杂端口上启用 Port Fast 和 BPDU 防护。
- 如果删除在私有 VLAN 配置中使用的 VLAN，则与该 VLAN 关联的私有 VLAN 端口将变为非活动状态。
- 如果设备是中继连接的，并且主 VLAN 和辅助 VLAN 没有从中继删除，则私有 VLAN 端口可以在不同的网络设备上。

如何配置私有 VLAN

配置私有 VLAN

配置私有 VLAN，请执行以下步骤：

注释： VTP 1、2 和 3 的透明模式支持私有 VLAN。VTP 3 的服务器模式也支持私有 VLAN。

总步骤

1. 将 VTP 设置为 **transparent** 模式
2. 创建主 VLAN 和辅助 VLAN，并将它们关联起来。
3. 将接口配置为隔离或团体主机端口，并将 VLAN 的全体成员分配给主机端口。
4. 将接口配置为混杂端口，并将混杂端口映射到主、辅助 VLAN 对。
5. 如果使用 VLAN 间路由，则配置主 SVI，并将辅助 VLAN 映射到主 VLAN。
6. 验证私有 VLAN 配置。

具体步骤

步骤 1	将 VTP 设置为 **transparent** 模式 **注释：**对于 VTP3，您可以将模式设置为服务器或透明模式。
步骤 2	创建主 VLAN 和辅助 VLAN，并将他们相连。 请见在私有 VLAN 中配置和连接 VLAN **注释：**如果尚未创建 VLAN，则私有 VLAN 配置过程将创建该 VLAN。
步骤 3	将接口配置为隔离或团体主机端口，并将 VLAN 的全体成员分配给主机端口。 请见配置二层接口作为私有 VLAN 主机接口
步骤 4	将接口配置为混杂端口，并将混杂端口映射到主、辅助 VLAN 对。 请见配置二层接口作为私有 VLAN 混杂接口
步骤 5	如果使用 VLAN 间路由，则配置主 SVI，并将辅助 VLAN 映射到主 VLAN。 请见将辅助 VLAN 映射到主 VLAN 的三层 VLAN 接口
步骤 6	验证私有 VLAN 配置。

在私有 VLAN 中配置和关联 VLAN

在退出 VLAN 配置模式之前，**private-vlan** 命令不会生效。
在私有 VLAN 中配置和关联 VLAN，请执行以下步骤：
总步骤
1. enable
2. configure terminal
3. vtp mode transparent
4. vlan *vlan-id*
5. private-vlan primary
6. exit
7. vlan *vlan-id*
8. private-vlan isolated
9. exit
10. vlan *vlan-id*
11. private-vlan community
12. exit
13. vlan *vlan-id*
14. private-vlan community
15. exit
16. vlan *vlan-id*
17. private-vlan association [**add** | **remove**] *secondary_vlan_list*
18. end
19. show vlan private-vlan [**type**] or **show interfaces status**
20. copy running-config startup config
具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**vtp mode transparent** **示例:** `Device(config)#` **vtp mode transport**	将 VTP 设置为透明模式（禁用 VTP）。 **注释：**对于 VTP 3，可设置为服务器或透明模式。
步骤 4	**vlan** *vlan-id* **示例:** `Device(config)#` **vlan 20**	进入 VLAN 配置模式，并指定或创建一个 VLAN 作为主 VLAN。该 VLAN 的 ID 范围为 2 到 1001 以及 1006 到 4094。

步骤 5	**private-vlan primary** **示例:** `Device(config-vlan)#` **private-vlan primary**	将该 VLAN 指定为主 VLAN。
步骤 6	**exit** **示例:** `Device(config-vlan)#` **exit**	返回全局配置模式。
步骤 7	**vlan** *vlan-id* **示例:** `Device(config)#` **vlan 501**	（可选）进入 VLAN 配置模式，并指定或创建一个 VLAN 作为隔离 VLAN。该 VLAN 的 ID 范围为 2 到 1001 以及 1006 到 4094。
步骤 8	**private-vlan isolated** **示例:** `Device(config-vlan)#` **private-vlan isolated**	将该 VLAN 指定为隔离 VLAN。
步骤 9	**exit** **示例:** `Device(config-vlan)#` **exit**	返回全局配置模式。
步骤 10	**vlan** *vlan-id* **示例:** `Device(config)#` **vlan 502**	（可选）进入 VLAN 配置模式，并指定或创建一个 VLAN 作为团体 VLAN。该 VLAN 的 ID 范围为 2 到 1001 以及 1006 到 4094。
步骤 11	**private-vlan community** **示例:** `Device(config-vlan)#` **private-vlan community**	将该 VLAN 指定为团体 VLAN。
步骤 12	**exit** **示例:** `Device(config-vlan)#` **exit**	返回全局配置模式。
步骤 13	**vlan** *vlan-id* **示例:** `Device(config)#` **vlan 503**	（可选）进入 VLAN 配置模式，并指定或创建一个 VLAN 作为团体 VLAN。该 VLAN 的 ID 范围为 2 到 1001 以及 1006 到 4094。
步骤 14	**private-vlan community** **示例:** `Device(config-vlan)#` **private-vlan community**	将该 VLAN 指定为团体 VLAN。

步骤 15	**exit** **示例:** `Device(config-vlan)#` **exit**	返回全局配置模式。
步骤 16	**vlan** *vlan-id* **示例:** `Device(config)#` **vlan 20**	进入步骤 4 中指定的主 VLAN 的 VLAN 配置模式。
步骤 17	**private-vlan association** [**add** | **remove**] *secondary_vlan_list* **示例:** `Device(config-vlan)#` **private-vlan association 501-503**	将辅助 VLAN 与主 VLAN 关联。参数可以是单个私有 VLAN ID 或用连字符连接的私有 VLAN 的 ID 范围。 • *secondary_vlan_list* 参数不能包含空格。它可以包含多个逗号分隔的项目。每个项目可以是单个私有 VLAN 的 ID 或用连字符连接的私有 VLAN 的 ID 范围。 • *secondary_vlan_list* 参数可以包含多个团体 VLAN 的 ID，但只能包含一个隔离的 VLAN 的 ID。 • 输入 *secondary_vlan_list* 或使用带有 *secondary_vlan_list* 的 **add** 关键字将辅助 VLAN 与主 VLAN 关联。 • 使用 **remove** 关键字及 *secondary_vlan_list* 可以清除辅助 VLAN 和主 VLAN 之间的关联。 • 在退出 VLAN 配置模式之前，命令不会生效。
步骤 18	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 19	**show vlan private-vlan** [**type**] or **show interfaces status** **示例:** `Device#` **show vlan private-vlan**	验证配置。
步骤 20	**copy running-config startup config** **示例:** `Device#` **copy running-config startup-config**	将配置的条目保存在设备启动配置文件中。

将二层接口配置为私有 VLAN 主机端口

用户可以按照以下步骤将二层接口配置为私有 VLAN 主机端口，并将其与主 VLAN 和辅助 VLAN 关联：

注释：隔离 VLAN 和团体 VLAN 都是辅助 VLAN。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. switchport mode private-vlan host

5. switchport private-vlan host-association *primary_vlan_id secondary_vlan_id*

6. end

7. show interfaces [*interface-id*] **switchport**

8. copy running-config startup-config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**interface** *interface-id* **示例:** `Device(config)` **interface gigabitethernet1/0/22**	进入配置二层接口的接口配置模式。
步骤 4	**switchport mode private-vlan host** **示例:** `Device(config-if)#` **switchport mode private-vlanhost**	将二层端口配置为私有 VLAN 主机端口。
步骤 5	**switchport private-vlan host-association** *primary_vlan_idsecondary_vlan_id* **示例:** `Device(config-if)#` **switchport private-vlan host-association 20 501**	将二层端口与私有 VLAN 关联。 **注释：**这是将 PVLAN 与第 2 层接口相关联所必需的步骤。
步骤 6	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。

步骤 7	**show interfaces** [*interface-id*] **switchport** **示例:** `Device#` **show interfaces gigabitethernet1/0/22 switchport**	验证配置。
步骤 8	**copy running-config startup-config** **示例:** `Device#` **copy running-config startup-config**	（可选）将配置的条目保存在设备启动配置文件中。

将二层接口配置为私有 VLAN 混杂端口

用户可以按照以下步骤将二层接口配置为私有 VLAN 混杂端口，并将其与主 VLAN 和辅助 VLAN 进行映射：

注释：隔离 VLAN 和团体 VLAN 都是辅助 VLAN。

总步骤

1. enable

2. configure terminal

3. interface *interface-id*

4. switchport mode private-vlan promiscuous

5. switchport private-vlan mapping *primary_vlan_id* {add | remove} *secondary_vlan_list*

6. end

7. show interfaces [*interface-id*] switchport

8. copy running-config startup config

具体步骤

	命令或操作	**目的**
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**interface** *interface-id* **示例:** `Device(config)#` **interface gigabitethernet1/0/2**	进入配置二层接口的接口配置模式。
步骤 4	**switchport mode private-vlan promiscuous** **示例:**	将二层端口配置为私有 VLAN 混杂端口。

	Device(config-if)# **switchport mode private-vlan promiscuous**	
步骤 5	**switchportprivate-vlanmapping** *primary_vlan_id* {**add** | **remove**} *secondary_vlan_list* **示例:** Device(config-if)# **switchport private-vlan mapping 20 add 501-503**	将私有 VLAN 混杂端口映射到主 VLAN 和所选的辅助 VLAN。 • *secondary_vlan_list* 参数不能包含空格。它可以包含多个逗号分隔的项目。每个项目可以是单个私有 VLAN 的 ID 或用连字符连接的私有 VLAN 的 ID 范围。 • 输入 *secondary_vlan_list* 或使用带有 *secondary_vlan_list* 的 **add** 关键字将辅助 VLAN 映射到私有 VLAN 混杂端口。 • 使用 **remove** 关键字及 *secondary_vlan_list* 可以清除辅助 VLAN 和私有 VLAN 混杂端口之间的映射。
步骤 6	**end** **示例:** Device(config)# **end**	返回特权 EXEC 模式。
步骤 7	**show interfaces** [*interface-id*] **switchport** **示例:** Device# **show interfaces gigabitethernet1/0/2 switchport**	验证配置。
步骤 8	**copy running-config startup config** **示例:** Device# **copy running-config startup-config**	将配置的条目保存在设备启动配置文件中。

将辅助 VLAN 映射到主 VLAN 三层接口

如果私有 VLAN 会用于 VLAN 间路由，则为主 VLAN 配置 SVI，并将辅助 VLAN 映射到 SVI。

注释：隔离 VLAN 和团体 VLAN 都是辅助 VLAN。

用户可以按照以下步骤将辅助 VLAN 映射到主 VLAN 的 SVI，以允许对私有 VLAN 流量进行三层交换：

总步骤

1. enable

2. configure terminal

3. interface vlan *primary_vlan_id*

4. private-vlan mapping [add | remove] ***secondary_vlan_list***

5. end

6. show interface private-vlan mapping

7. copy running-config startup config

具体步骤

	命令或操作	目的
步骤 1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码。
步骤 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步骤 3	**interface vlan** *primary_vlan_id* **示例:** `Device(config)#` **interface vlan 20**	进入主 VLAN 的接口配置模式，并将 VLAN 配置为 SVI。VLAN 的 ID 范围为 2 到 1001 以及 1006 到 4094。
步骤 4	**private-vlan mapping** [**add** | **remove**] *secondary_vlan_list* **示例:** `Device(config-if)#` **private-vlan mapping 501-503**	将辅助 VLAN 映射到主 VLAN 的三层 VLAN 接口，以允许三层交换私有 VLAN 入口流量。 **注释：private-vlan mapping** 接口配置命令只影响三层交换的私有 VLAN 流量。 • *secondary_vlan_list* 参数不能包含空格。它可以包含多个逗号分隔的项目。每个项目可以是单个私有 VLAN 的 ID 或用连字符连接的私有 VLAN 的 ID 范围。 • 输入 *secondary_vlan_list* 或使用带有 *secondary_vlan_list* 的 **add** 关键字将辅助 VLAN 映射到主 VLAN。 • 使用 **remove** 关键字及 *secondary_vlan_list* 可以清除辅助 VLAN 和主 VLAN 之间的映射。
步骤 5	**end** **示例:** `Device(config)#` **end**	返回特权 EXEC 模式。
步骤 6	**show interface private-vlanmapping** **示例:** `Device#` **show interfaces private-vlan mapping**	验证配置。
步骤 7	**copy running-config startup config**	将配置的条目保存在设备启动配置文件中。

	示例: `Device#` **copy running-config startup-config**	

监控私有 VLAN

下表展示了用于监控私有 VLAN 的命令。

表 216：私有 VLAN 的监控命令

命令	目的
show interfaces status	显示接口的状态，包括它们所属的 VLAN。
show vlan private-vlan [type]	显示设备或设备堆栈的私有 VLAN 信息。
show interface switchport	显示接口上的私有 VLAN 配置。
show interface private-vlanmapping	显示私有 VLAN 映射 VLAN SVI 的有关信息。
show platform vlan pvlan	显示 FED 侧的 PVLAN 信息。
show platform vlan pvlan hardware	显示 FAD 侧的 PVLAN 所拥有的所有硬件资源。

私有 VLAN 的配置示例

示例：在私有 VLAN 中配置和关联 VLAN

此示例显示如何将 VLAN 20 配置为主 VLAN，将 VLAN 501 配置为隔离 VLAN，将 VLAN 502 和 503 配置为团体 VLAN，以将它们关联到私有 VLAN，并验证配置：

```
Device# configure terminal
Device(config)# vlan 20
Device(config-vlan)# private-vlan primary
Device(config-vlan)# exit
Device(config)# vlan 501
Device(config-vlan)# private-vlan isolated
Device(config-vlan)# exit
Device(config)# vlan 502
Device(config-vlan)# private-vlan community
Device(config-vlan)# exit
Device(config)# vlan 503
Device(config-vlan)# private-vlan community
Device(config-vlan)# exit
Device(config)# vlan 20
Device(config-vlan)# private-vlan association 501-503
Device(config-vlan)# end
Device# show vlan private-vlan
Primary Secondary Type
--------- -------------- -----------------
```

```
20 501 isolated
20 502 community
20 503 community
```

示例：将接口配置为主机端口

此示例展示如何将接口配置为私有 VLAN 主机端口，将其与私有 VLAN 对关联，并验证配置：

```
Device# configure terminal
Device(config)# interface gigabitethernet1/0/22
Device(config-if)# switchport mode private-vlan host
Device(config-if)# switchport private-vlan host-association 20 501
Device(config-if)# end
Device# show interfaces gigabitethernet1/0/22 switchport
Name: Gi1/0/22
Switchport: Enabled
Administrative Mode: private-vlan host
Operational Mode: private-vlan host
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: 20 501
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan:
20 501
<output truncated>
```

示例：将接口配置为私有 VLAN 混杂端口

此示例展示如何将接口配置为私有 VLAN 混杂端口，并将其映射到私有 VLAN。该接口是主 VLAN 20 的成员，且其映射了辅助 VLAN 501 至 503。

```
Device# configure terminal
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# switchport mode private-vlan promiscuous
Device(config-if)# switchport private-vlan mapping 20 add 501-503
```

```
Device(config-if)# end
```

使用 **show vlan private-vlan** 或 **show interface status** 特权 EXEC 命令显示设备上的主 VLAN、辅助 VLAN 以及私有 VLAN 端口。

示例：将辅助 VLAN 映射到主 VLAN 接口

此示例展示如何将 VLAN 501 和 502 的接口映射到主 VLAN 10，以允许进行从私有 VLAN 501 到 502 的辅助 VLAN 入向流量的路由：

```
Device# configure terminal
Device(config)# interface vlan 20
Device(config-if)# private-vlan mapping 501-503
Device(config-if)# end
Device# show interfaces private-vlan mapping
Interface Secondary VLAN Type
--------- -------------- -----------------
vlan20 501 isolated
vlan20 502 community
vlan20 503 community
```

示例：监控私有 VLAN

以下示例展示了 **show vlan private-vlan** 命令的输出：

```
Device# show vlan private-vlan
Primary Secondary Type Ports
------- --------- ----------------- ------------------------------------------
20 501 isolated Gi1/0/22, Gi1/0/2
20 502 community Gi1/0/2
20 503 community Gi1/0/2
```

接下来做什么？

可进行以下配置：

- VTP
- VLANs
- VLAN 中继
- VLAN 成员策略服务器（VLAN Membership Policy Server，VMPS）
- 语音 VLAN

其他参考资料

相关文档

相关主题	文档题目
CLI 命令	LAN 交换命令参考，InspurINOS 版本

标准和 RFC

标准/RFC	题目
RFC 1573	
RFC 1757	
RFC 2021	

技术助手

描述	链接
Inspur 支持网站为排错和解决 Inspur 产品的技术问题提供了大量的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订阅多种服务，如产品报警工具（通过现场通知访问），Inspur 技术服务简讯以及 RSS 源。 访问 Inspur 支持网站的大部分工具都需要提供 icntnetworks.com 的用户 ID 及密码。	http://www.icntnetworks.com

第 17 部分 YANG 数据模型

配置 YANG 数据模型

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看 误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如 查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。
用户可以使用 Inspur 特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访 Inspur 特性导航（Inspur Feature Navigator），可以访 http://www.icntnetworks.com。用户不 要在 icntnetworks.com 注册账户就可以使用这个导航系统。

数据模型简介——程序化的基于标准的配置

管理网络设备的传统方式是使用命令行界 （Command Line Interface，CLI）管理配置（配置命令）和运行数据（显示命令）。对于网络管理，简单网络管理协议（Simple Network Management Protocol，SNMP）被广泛使用，特别是用于在各种网络设备之一交换管理信息。虽然 CLI 和 SNMP 被大 使用，但它们有几个 制。CLI 是 度专有的，并且 要人为介入来理解和解释其基于文本的规范。SNMP 不区分配置和运行数据。
解决方案是采用一种程序化的基于标准的方式为任何网络设备编写配置，以此取代手动配置的过程。在 InpsurINOS 11.3.1 上运行的网络设备支持使用数据模型对网络上的多个设备进行自动化配置。数据模型以标准的行业定义语言开发，可以定义网络的配置和状态信息。

InpsurINOS 11.3.1 支持 YANG（Yet Another Next Generation）数据建模语言。 YANG 可以与网络配置协议（Network Configuration Protocol，NETCONF）一起用于提供所　的自动化和可编程网络操作的解决方案。 NETCONF（RFC 6241）是一种基于 XML 的协议，客户端应用程序可以使用该协议从设备请求信息和对设备进行配置更改。YANG 主要用于对 NETCONF 操作使用的配置和状态数据进行建模。
在 InspurINOS 11.3.1 中，基于模型的接口可以与现有设备 CLI，Syslog 和 SNMP 接口进行互操作。网络设备可选地向上提供这些接口。YANG 基于 RFC 6020 对每个协议建模。

NETCONF

NETCONF 提供了一种更简单的机制，用于安装、操作和删　网络设备的配置。
对于配置数据以及协议消息，它使用基于可扩展标记语言（Extensible Markup Language，XML）的数据编码方式。
NETCONF 使用简单的基于远程过程调用（Remote Procedure Call，RPC）的机制来进行客户端和服务器之一的通信。客户端可以是作为网络管理器的一部分运行的脚本或应用。服务器通常是网络设备（交换机或路由器）。它使用安全外壳（Secure Shell，SSH）作为网络设备之一的传输层。
NETCONF 还支持性能发现和模型下载，它使用 *ietf-netconf-monitoring* 模型发现支持的模型。每个模型的修订日期显示在性能响应中。数据模型可以使用 *get-schema* rpc 从设备上下载。可以使用这些 YANG 模型来了解或导出数据模型。
更多细节请参　RFC 6241。

配置 NETCONF

在开始前
必代按如下所示配置 NETCONF-YANG。
总步
1. enable
2. configure terminal
3. netconf-yang
4. exit

	命令或操作	目的
步　1	**enable** **示例:** `Device>`**enable**	进入特权 EXEC 模式。在提示时输入密码
步　2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步　3	**netconf-yang**	在网络设备上启用 NETCONF 接口。

	示例: `Device (config)#` **netconf-yang**	**注释：**在通过 CLI 初始启用之后，可以通过基于模型的接口管理网络设备。完全激活基于模型的接口进程可能 要达 90 秒。
步 4	**exit** **示例:** `Device (config)#` **exit**	退出全局配置模式。

配置 NETCONF 选

配置 SNMP

启用 INOS 中的 SNMP 服务器，使 NETCONF 能够使用从支持的 MIB 生成的 YANG 模型访 SNMP MIB 数据，并启用 INOS 中支持的 SNMP ，从支持的 中接收 NETCONF 通知。执行以下步 ：

总步

1.在 INOS 中启用 SNMP 功能。

2.在 NETCONF-YANG 启动后，通过向 NETCONF-YANG 端口发送以下 RPC <edit-config>消息来启用 SNMP 支持。

3.发送以下 RPC 消息到 NETCONF-YANG 端口，将运行配置保存到启动配置。

具体步

步 1	在 INOS 中启用 SNMP 功能。
	
	示例：
	`configure terminal`
	`logging history debugging`
	`logging snmp-trap emergencies`
	`logging snmp-trap alerts`
	`logging snmp-trap critical`
	`logging snmp-trap errors`
	`logging snmp-trap warnings`
	`logging snmp-trap notifications`
	`logging snmp-trap informational`
	`logging snmp-trap debugging`
	`!`
	`snmp-server community public RW`
	`snmp-server trap link ietf`
	`snmp-server enable traps snmp authentication linkdown linkup snmp-server enable traps syslog`
	`snmp-server manager`
	`exit`

步 2	在 NETCONF-YANG 启动后，通过向 NETCONF-YANG 端口发送以下 RPC <edit-config> 消息来启用 SNMP 支持。 **示例：** <?xml version="1.0" encoding="utf-8"?> <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id=""> <edit-config> <target> <running/> </target> <config> <netconf-yang xmlns="http://icntnetworks.com/yang/inspurself-mgmt"><inspur-ia xmlns="http://icntnetworks.com/yang/ins<psunrm-pi-at"r>a p-control> <trap-list> <trap-oid>1.3.6.1.4.1.9.9.41.2.0.1</trap-oid> </trap-list> <trap-list> <trap-oid>1.3.6.1.6.3.1.1.5.3</trap-oid> </trap-list> <trap-list> <trap-oid>1.3.6.1.6.3.1.1.5.4</trap-oid> </trap-list> </snmp-trap-control> </inspur-ia> </netconf-yang> </config> </edit-config> </rpc>
步 3	发送以下 RPC 消息到 NETCONF-YANG 端口，将运行配置保存到启动配置。 **示例：** <?xml version="1.0" encoding="utf-8"?> <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id=""> <inspur-ia:save-config xmlns:inspur-ia="http://icntnetworks.com/yang/inspur-ia"/> </rpc>

其他参考资料

相关文档

组件	**RFC**

YANG	6020
NETCONF	6241

可编程性：网络引导加载程序

InspurINOS 11.3.1 支持以下可编程特性：

- 网络引导加载程序
- 即插即用（Plug-N-Play，PnP）代理启动
- 查询特性信息，
- 关于可编程性的信息
- 如何配置可编程性：网络引导加载程序
- 可编程性配置示例：网络引导加载程序
- 可编程性的其他参考资料：网络引导加载程序
- 可编程性的特性信息：网络引导加载程序

查询特性信息

用户的软件版本有可能无法支持这部分文档所提到的全部特性。要想查询最新的警示信息和特性信息，可以查看 误搜索工具（Bug Search Tool），也可以查看自己使用的平台及软件版本的版本信息。用户如 查找本文档中所提到的特性，并且了解都有哪些系统版本支持这个特性，可以查看文档最后的特性信息表。

用户可以使用 Inspur 特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访 Inspur 特性导航（Inspur Feature Navigator），可以访 http://www.icntnetworks.com。用户不 要在 icntnetworks.com 注册账户就可以使用这个导航系统。

关于可编程性的信息

网络引导加载程序概述

网络引导加载程序支持从基于网络的源进行启动。引导加载程序引导位于 HTTP、FTP 和 TFTP 服务器上的映像。网络引导源使用类似 iPXE 的方法进行自动检测。

iPXE 是 启动执行环境（Preboot eXecution Environment，PXE）的开源实现。它为离线设备启用网络引导。以下是三种类型的 iPXE 引导模式：

- iPXE 超时——使用 IPXE_TIMEOUT rommon 变　配置 iPXE 网络引导的超时（以秒为单位）。当超时过期时，交换机将恢复为设备引导。
- iPXE 启动——通过 iPXE 网络引导启动。当配置 **boot ipxe forever** 命令时，交换机持续发送动态主机控制协议（Dynamic Host Control Protocol，DHCP）请求。这是仅通过 iPXE 引导的方式。
- 设备——用 **boot ipxe** 命令配置设备引导。配置设备引导时，将忽略配置的 IPXE_TIMEOUT rommon 变　。

以下部分介绍 iPXE 引导加载程序的工作原理：

图 145：iPXE 引导加载程序的工作流程

Power ON	打开
DHCP discover	DHCP 发现
DHCP offer	DHCP 提议
DHCP request	DHCP 请求
DHCP Ack	DHCP 确认
DHCP Server	DHCP 服务器
Switch downloads the image and Boots up.	交换机下载映像并启动
Here's my UID. Give me an IP. What image can I boot? How do I reach the remote server?	这是我的 UID。给我一个 IP。 我可以启动什么映像？ 如何连接远程服务器？
Your (client) IP address Next server IP address Boot file name	你的（客户端）IP 地址 下一个服务器 IP 地址 引导文件名
HTTP ''Get''	HTTP“GET”
HTTP 200 Ok	HTTP 200 Ok
TCP Transter	TCP 传输
Web Server	Web 服务器
Here's the image	此处为该映像

1 引导加载程序发送一个 DHCP 请求。

2 DHCP 回应 IP 地址和引导文件名。引导文件名称表示从 TFTP 服务器（tftp://server/filename），FTP 服务器（ftp://userid:password@server/filename）或 HTTP 服务器（http://server/filename）检索引导映像。由于当前的 iPXE 仅通过管理端口（GigabitEthernet0/0）工作，因此不支持通过前　板端口发送的 DHCP 请求。

3 引导加载程序从网络源下载和引导映像。

4 如果未收到 DHCP 响应，则引导加载程序将根据引导加载程序配置，一直发送 DHCP 请求或者发送一定的时一。当超时发生时，引导加载程序将恢复为基于设备的引导。只有配置的引导模式为 **ipxe-forever** 时，交换机才会一直发送 DHCP 请求。如果配置了 **ipxe-timeout** 引导模式命令，则会在指定的时一内发送 DHCP 请求，超时时一过后，交换机将恢复为设备引导。

当禁用手动引导时，引导加载程序将根据 iPXE ROMMON 变　的配置值确定执行设备引导还是执行网络引导。无论手动引导是启用还是禁用，引导加载程序都使用 BOOTMODE 变　来确定是执行设备引导还是网络引导。手动启动意味着用户必代手动　入 **boot manual switch** 命令以启动引导过程。当禁用手动引导以及当交换机重新加载时，引导过程将自动启动。

当 iPXE 被禁用时，设备会按照现有 BOOT 变　的内容确定如何引导。BOOT 变　可能包含一个基于网络的统一资源标识符（network-based uniform resource identifier，URI）（例如，http://、

ftp://、tftp://）；然而 DHCP 不用于获取网络映像路径。设备 IP 地址取自 IP_ADDR 变　。BOOT 变　还可能包含基于设备的路径，在这种情况下，启动基于设备的引导。

要在远程 DHCP 服务器上获取 UUT 以进行引导，请使用机箱序列号（DHCP 选　61 中提供），产品 ID（Product ID，PID）（DHCP 选　60 中提供）或交换机 MAC 地址。使用 **show inventory** 和 **show switch** 命令也会在交换机上显示这些数值。

以下是 show inventory 命令的样例输出：

```
Switch# show inventory
NAME:"c38xx Stack", DESCR:"c38xx Stack"
PID:WS-3850-12X-48U-L, VID:V01 , SN: F0C1911V01A
NAME:"Switch 1", DESCR:"WS-C3850-12X48U-L"
PID:WS-C3850-12X48U-L, VID:V01 , SN:F0C1911V01A
NAME:"Switch1 -Power Supply B", DESCR:"Switch1 -Power Supply B"
PID:PWR-C1-1100WAC, VID:V01, SN:LIT1847146Q
```

为 iPXE 配置以下 rommon 变　：

- BOOTMODE = ipxe-forever | ipxe-timeout | device
- IPXE_TIMEOUT = seconds

即插即用代理概述

Inspur 即插即用（Plug-N-Play，PnP）代理是平台的引导代理程序。基于设备的引导代理与已识别的引导服务器进行交互操作。

平台的引导代理/ PnP 代理能满足以下常见要求：

- 机房内带外引导——在管理端口上使用 DHCP；
- 机房外带外引导——在管理端口上使用基于云的连接，例如，使用 DNS 和 InspurPnP 协议；
- 机房外带内引导——在数据端口上使用基于云的连接，例如，使用 DNS 和 InspurPnP 协议。

如何配置可编程性：网络引导加载程序

配置引导加载程序

总步

1. enable

2. configure terminal

3. boot ipxe {**forever** | **timeout** *seconds*} *switch number*

4. boot system {**flash:** | **ftp:** | **http:** | **switch:**| **tftp:**}

5. end

具体步

	命令或操作	**目的**
步　1	**enable**	进入特权 EXEC 模式。在提示时输入密

	示例: `Switch> enable`	码
步 2	**configure terminal** **示例:** `Device#` **configure terminal**	进入全局配置模式。
步 3	**boot ipxe** {**forever** | **timeout** *seconds*} *switch number* **示例:** `Switch(config)# boot ipxe forever switch 2`	将 BOOTMODE rommon 变 配置为 IPXE-FOREVER。 ● 超时关 字以秒为单位配置 rommon 变 IPXE_TIMEOUT 的值
步 4	**boot system** {**flash:** | **ftp:** | **http:** | **switch:**| **tftp:**} **示例:** `Switch(config)# boot system http:`	配置引导路径。
步 5	**end** **示例:** `Switch(config)# end`	退出全局配置模式并返回到特权 EXEC 模式。

可编程性配置示例：网络引导加载程序

示例：配置引导加载程序

以下示例演示引导加载程序持续发送 DHCP 请求，直到交换机通过映像启动：

```
Switch# configure terminal
Switch(config)# boot ipxe forever switch 2
Switch(config)# boot system http: image-filename
Switch(config)# end
```

以下示例显示为引导加载程序配置超时配置。当超时发生时，引导加载程序将恢复为基于设备的引导。

```
Switch# configure terminal
Switch(config)# boot ipxe timeout 200 switch 2
Switch(config)# boot system ftp: image-filename
Switch(config)#end
```

可编程性的其他参考资料：网络引导加载程序

相关文档

相关主	**文档　目**
YANG Data Models	*Configuring YANG Data Models*

标准和 RFC

标准/RFC	**　目**
RFC 3986	*Uniform Resource Identifier (URI): Generic Syntax*

技术助手

描述	接
Inspur 支持网站为排　和解决 Inspur 产品的技术　　提供了大　的在线资源，包括文档及工具。 为了接收产品的安全及技术信息，管理员可以订　多种服务，如产品报警工具（通过现场通知访　），Inspur 技术服务简讯以及 RSS 源。 访　Inspur 支持网站的大部分工具都　要提供 icntnetworks.com 的用户 ID 及密码。	http://www.icntnetworks.com

可编程性的特性信息：网络引导加载程序

下表提供了有关本模块中描述的特性的版本信息。此表仅列出了对给定软件版本系列中的给定特性引入支持的软件版本。　　另有说明，该软件版本系列的后续版本也支持该特性。用户可以使用 Inspur 特性导航（Inspur Feature Navigator）来查询各个平台及不同 Inspur 软件版本所支持的信息。要访　Inspur 特性导航（Inspur Feature Navigator），可以访 http://www.icntnetworks.com。用户不　要在 icntnetworks.com 注册账户就可以使用这个导航系统。

表 217：可编程性的特性信息：网络引导加载程序

特性名称	版本	特性信息
可编程性：网络引导加载程序	InspurINOS 11.3.1	网络引导加载程序支持从基于设备或基于网络的源进行引导。必代能使用类似 iPXE 的方法检测到网络引导源。
可编程性：即插即用代理	InspurINOS 11.3.1	PnP 代理作为平台引导代理工作