

了解有关Debug命令的重要信息

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[规则](#)

[背景信息](#)

[警告](#)

[在调试之前](#)

[获得调试输出](#)

[控制台端口](#)

[AUX 端口](#)

[VTY端口](#)

[将信息记录到内部缓冲器](#)

[将信息记录到UNIX Syslog服务器](#)

[其他预调试任务](#)

[停止调试](#)

[使用 debug ip packet 命令](#)

[警告](#)

[有条件的被触发的调试](#)

[相关信息](#)

简介

本文档介绍使用 debug 命令(包括Cisco IOS®平台上可用的 debug ip packet 命令)的一般指南。

先决条件

要求

Cisco 建议您了解以下主题：

- [使用控制台、辅助和 VTY 端口连接到路由器](#)
- [Cisco IOS 常见配置问题](#)

•

解释 Cisco IOS 调试输出

使用的组件

本文档不限于特定的软件和硬件版本。

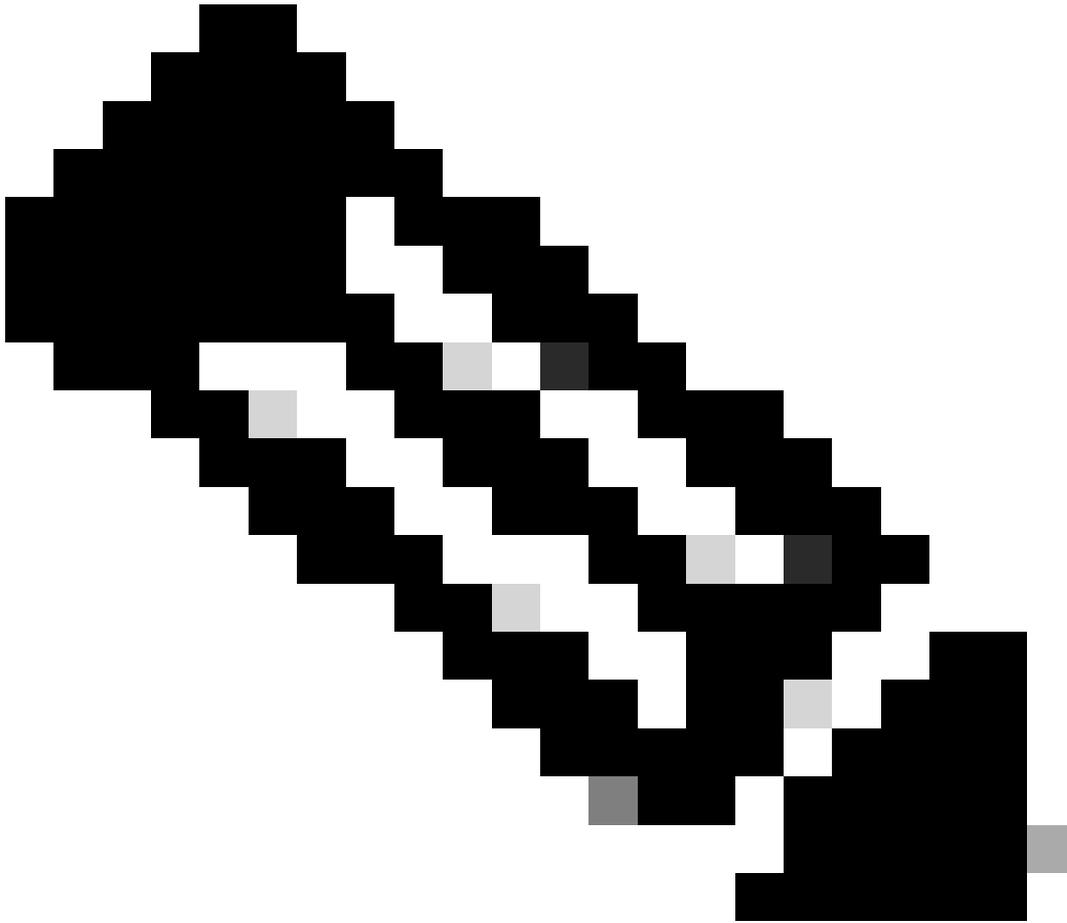
本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

规则

有关文档规则的详细信息，请参阅 Cisco 技术提示规则。

背景信息

本页提供有关使用Cisco IOS平台上提供的调试功能的一些一般指导原则，以及正确使用 **debug ip packet** 命令和条件调试的示例。



注意：本文档不介绍如何使用和解释特定的debug命令和输出。有关特定 debug 命令的信息，请参考相应的Cisco debug 命令参考文献。

特权EXEC命令 **debug** 的输出提供诊断信息，其中包括与协议状态和网络活动相关的各种互联事件。

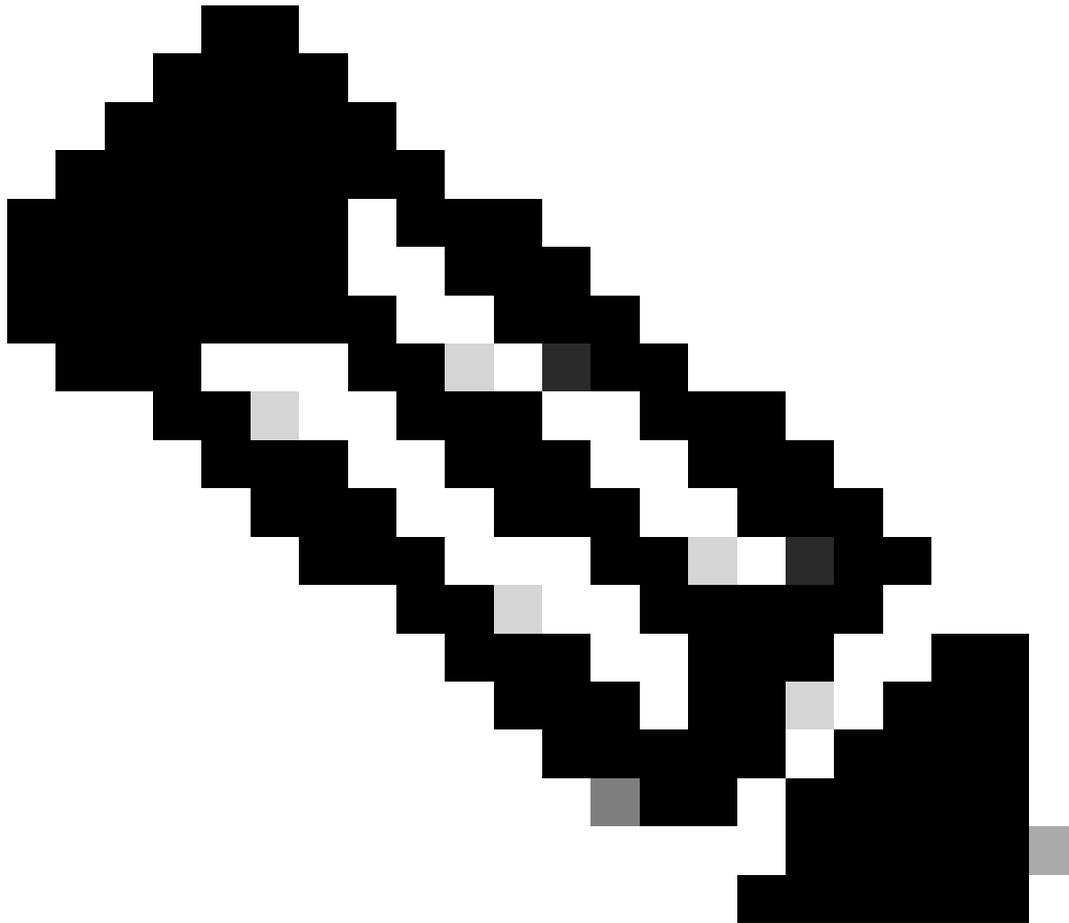
警告

debug 请谨慎使用命令。当解决具体问题时，通常只推荐在路由器技术支持人员提供指导的情况下使用这些命令。

在互联网处于高负载的情况下，启用调试功能会中断路由器的运行。因此，如果已启用日志记录，则在控制台端口中的日志消息出现过载后，接入服务器可能会间歇性地停止响应。

在启动 `debug` 命令之前，请始终考虑此命令可以生成的输出以及它可以花费的时间。例如，如果路由器具有一个基本速率接口 (BRI)，则 `debug isdn q931` 可能不会对系统产生危害。但是，在具有完整E1配置的AS5800上执行相同的调试可能会生成大量输入，以至于可以挂起并停止响应。

在调试之前，请使用 `show processes cpu` 命令查看您的CPU负载。在开始调试之前，请验证是否有足够的 CPU 可用。有关如何处理高 CPU 负载的详细信息，请参阅解决思科路由器上的高 CPU 使用率问题。例如，如果您有一台Cisco 7200路由器与ATM接口进行桥接，则根据配置的子接口数量，重新启动路由器会占用大量CPU。这是因为：对于每个虚电路(VC)，都需要生成网桥协议数据单元 (BPDU)数据包。在如此关键的时间启动调试会导致CPU利用率大幅上升，并导致挂起或网络连接丢失。



注意：当调试运行时，通常看不到路由器提示，尤其是当调试密集时。但是，在大多数情况下，您可以使用 `no debug all` 或 `undebug all` 命令来停止调试。有关安全使用调试的详细信息，请参阅获取调试输出部分。

在调试之前

除了上面提到的几点，请确保您了解调试对平台稳定性的影响。您还必须考虑必须连接路由器上的哪个接口。本部分介绍了一些指导原则。

获得调试输出

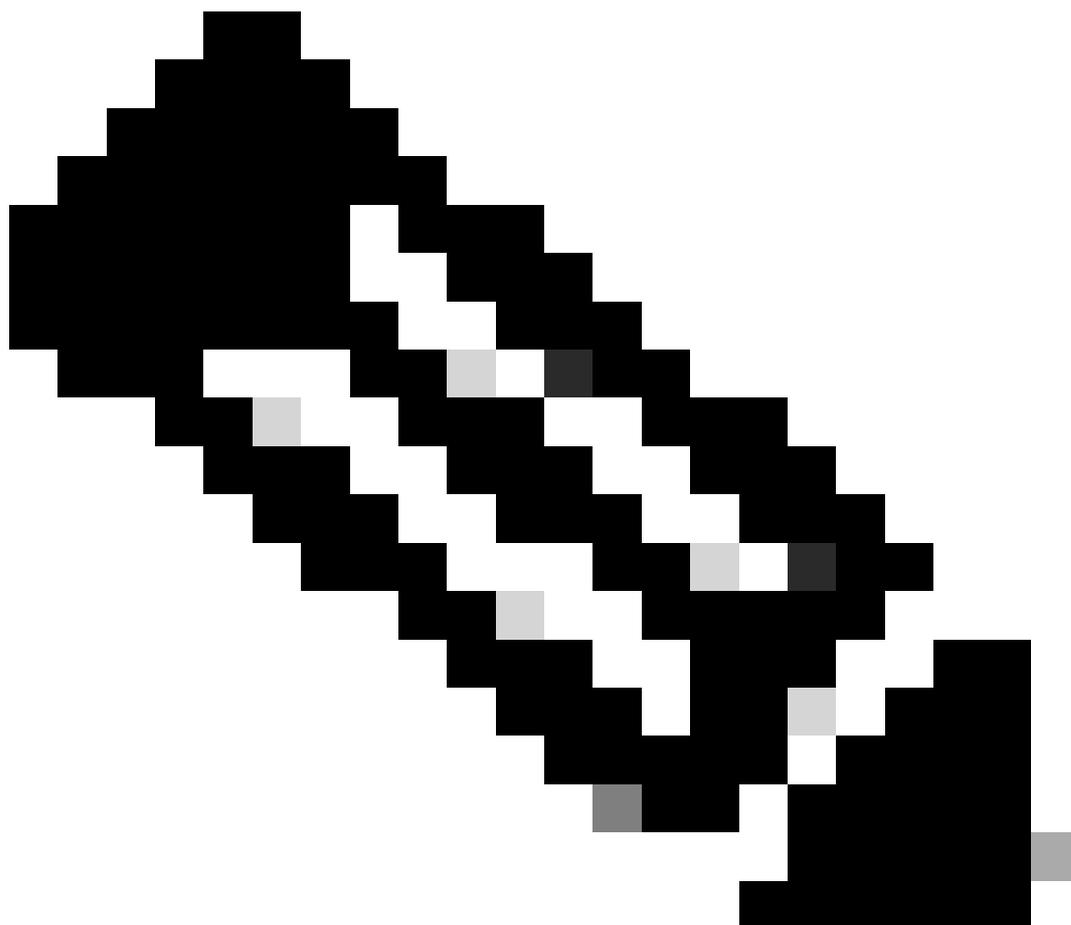
路由器可显示各种接口的调试输出，其中包括控制面板、aux和vty端口。路由器还可将发送到内部缓冲器的日志消息记录到外部unix syslog服务器上。下面将讨论每种方法的说明和注意事项：

控制台端口

如果您在控制台上连接，在正常配置下，无需执行额外操作。调试输出必须自动显示。但是，请确保 **logging console level** 已按要求设置日志且未使用 **no logging console** 命令禁用日志记录。



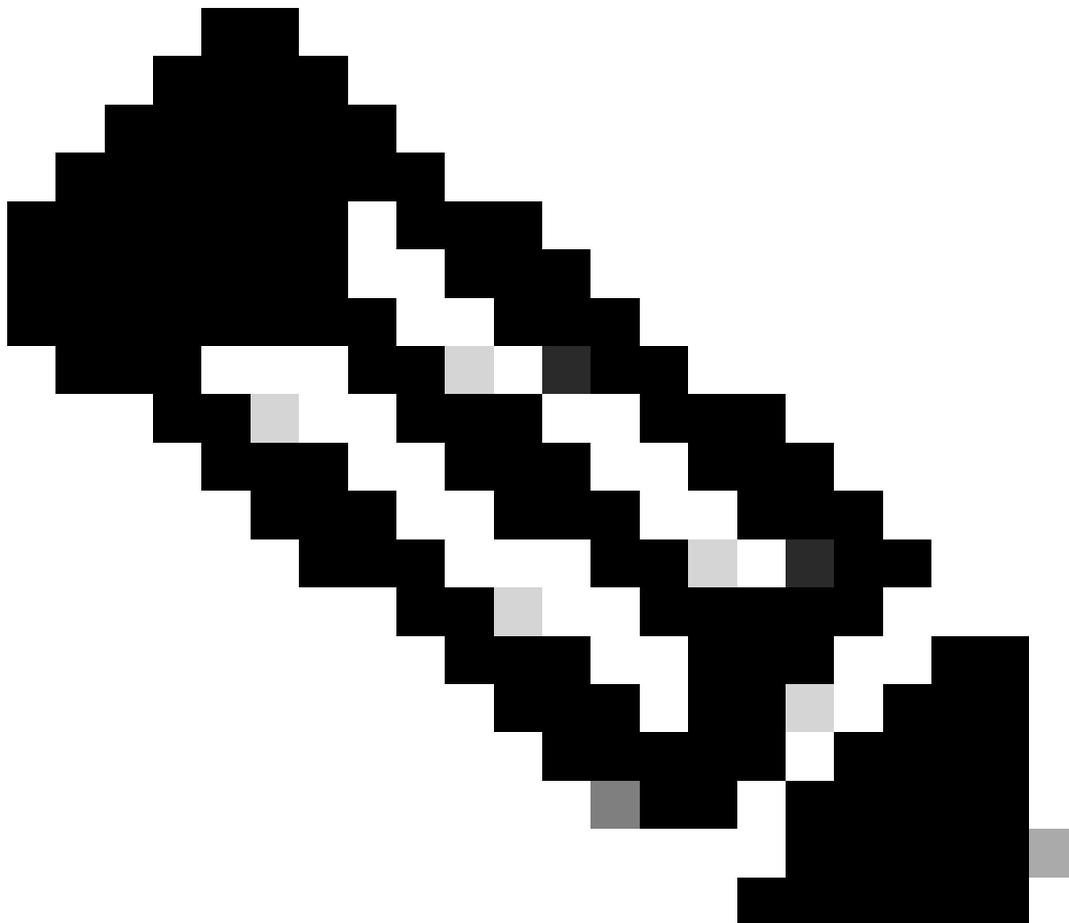
警告：对路由器控制台端口进行的过度调试会导致其挂起。这是因为：在执行路由器其他功能之前，路由器对控制面板输出自动划分优先级。因此，如果路由器正在处理到控制台端口的大量debug输出，它可能会挂起。所以，若出现过多的调试输出，则使用vty (telnet)端口或日志缓冲器来获得调试输出。下面提供了更多信息。



注意：默认情况下，在控制台端口上启用日志记录。即使实际上使用一些其他端口或方法(如：Aux、vty或缓冲器)来捕获输出，控制面板端口通常也要处理调试输出。因此，思科建议在正常运行条件下始终启用 no logging console 命令，并使用其他方法捕获调试。在需要使用控制面板的情况下，暂时将 logging console 设置为启用状态。

AUX 端口

如果通过辅助端口连接，请键入 **terminal monitor** 命令。另请检验该命令 **no logging on** 是否尚未在路由器上激活。



注意：如果使用Aux端口监控路由器，请记住，当路由器重新启动时，Aux端口不会显示引导序列输出。请连接到控制台端口以查看引导顺序。

VTY端口

如果通过辅助端口或telnet连接，请键入 **terminal monitor** 命令。另请检验命令 **no logging on** 是否尚未使用。

将信息记录到内部缓冲器

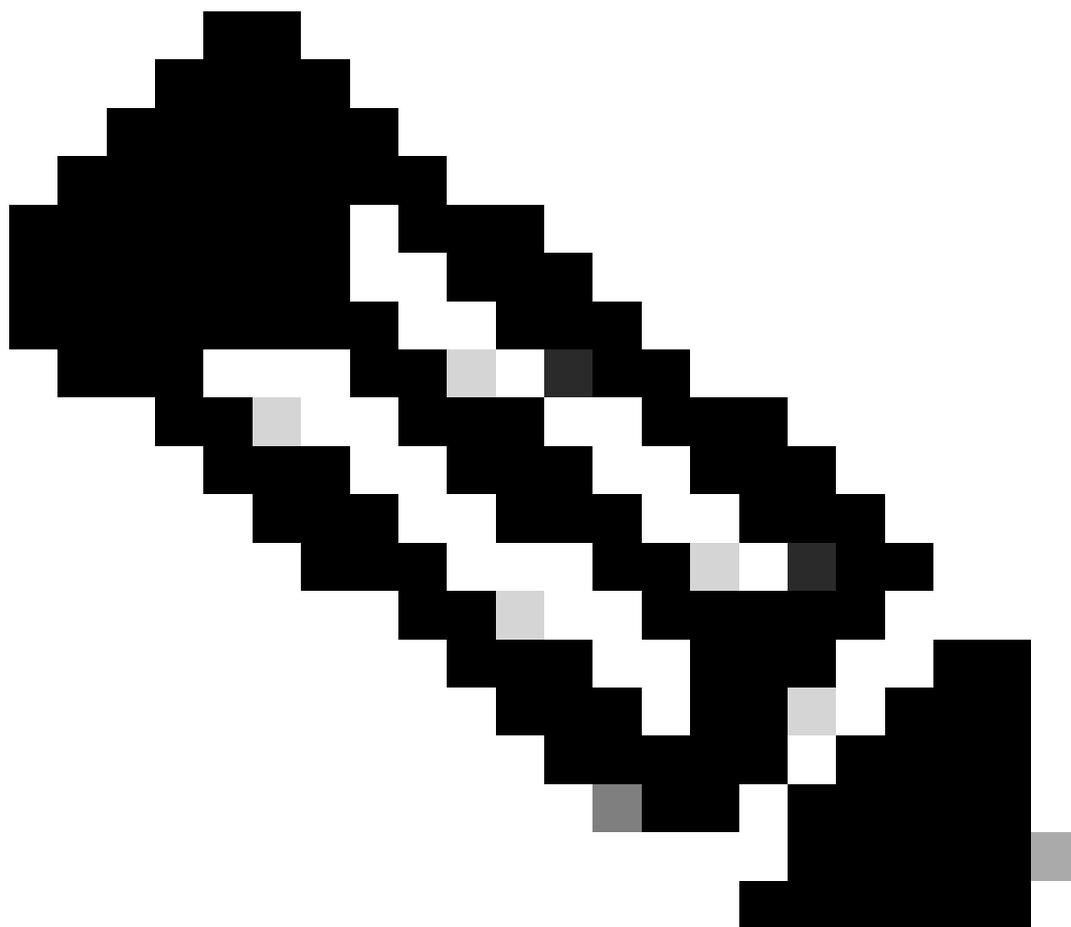
默认日志记录设备是控制台；除非另行指定，否则所有消息都会显示在控制台上。

要将消息记录到内部缓冲区，请使用 `logging buffered` 路由器配置命令。以下是该命令的完整语法：

```
<#root>
```

```
logging buffered no logging buffered
```

`logging buffered` 该命令将日志消息复制到内部缓冲区，而不是将其写入控制台。缓冲器具有循环特性，新消息将覆盖旧消息。要显示缓冲器中所记录的消息，请使用特权EXEC命令 `show logging`。所显示的第1个消息是缓冲器中最旧的消息。可以指定缓冲器的大小，以及要记录的信息的重要级别。



注意：在输入缓冲区大小之前，请确保机箱中有足够的可用内存。使用Cisco IOS show proc mem 命令，查看可用的内存。

no logging buffered 该命令会取消缓冲区的使用并将消息写入控制台（默认）。

将信息记录到UNIX Syslog服务器

若要将日志信息记录到syslog服务器主机上，则使用日志路由器配置命令。此命令的全部句法如下：

```
<#root>
```

```
logging <ip-address> no logging <ip-address>
```

logging该命令可标识要接收日志记录消息的系统日志服务器主机。<ip-address> 参数是主机的IP地址。通过多次发布此命令，可建立接收日志消息的syslog服务器列表。

no logging 该命令从系统日志列表中删除具有指定地址的系统日志服务器。

其他预调试任务

-

安装终端仿真器软件(例如，HyperTerminal) 以便将调试输出捕获到文件上。例如，在HyperTerminal中，点击**Transfer**，然后点击 **Capture Text**，然后选择适当的选项。有关详细信息，请参阅从超级终端捕获文本输出。有关其他终端仿真器软件，请参考软件文献。

-

使用 **service timestamps** 命令启用毫秒(msec)时间戳：

```
<#root>
```

```
router(config)#
```

```
service timestamps debug datetime msec
```

```
router(config)#
```

```
service timestamps log datetime msec
```

这些命令以MMM DD HH : MM : SS格式将时间戳添加到调试中，根据系统时钟指示日期和时间。若未设置系统时钟，则日期和时间之前带有星号(*)以说明日期和时间可能不正确。

通常合理的办法是配置毫秒时间标记，这是因为：在查看调试输出时，通过这种办法可实现更高级别的准确性。对于相互联系的调试事件，毫秒时间标记提供了更好的指示说明。但是，请注意，当控制台端口输出大量消息时，它们无法与事件的实际时间相关联。例如，如果您在有200条VC的盒上启 **debug x25** 用all，并且输出记录到缓冲区(使用 **no logging console logging buffered** theandcommand)，则debug输出（缓冲区内）中显示的时间戳不能是数据包通过接口的确切时间。因此，不要使用msec时间标记来证明性能情况，但可使用这种时间标记来获得事件发生时间的相对信息。

停止调试

若要停止调试，请使用**no debug allundebug all**理论命令。使用命令**show debug**，验证是否已经关闭调试。

请记住，**no logging console terminal no monitor** 命令只阻止输出分别通过控制台、Aux或vty输出。它不会停止调试，因此会耗尽路由器资源。

使用 debug ip packet 命令

debug ip packet 该命令可生成有关数据包的信息，这种数据包未经路由器进行快速交换。但是，由于生成每个数据包的输出，所以输出尺寸扩大，并因此使路由器处于“挂起”状态。因此，仅在最严格的控制下使 **debug ip packet** 用（如本节所述）。

限制输出的最佳方法 **debug ip packet** 是创建与调试链接的访问列表。只有符合访问列表标准的数据包才能使用 **debug ip packet**。不必在所有接口上都应用此访问列表，但非常适用于调试操作。

在使用 **debugging ip packet** 之前，请注意路由器在默认情况下执行快速交换，或者可以执行CEF交换（如果已配置为这样做）。也就是说，如果这些方法可用，则不向处理器提供数据包，因此，调试不显示任何内容。要使用此操作起作用，您需要使用 **no ip route-cache**（对于单播数据包）或 **no ip mroute-cache**（对于多播数据包）在路由器上禁用快速交换。这必须应用到流量流经的接口上。使用 **show ip route** 命令进行检验。

警告

- 对于处理大量数据包的路由器，启用路由器上快速交换可增加CPU利用率，从而使设备处于“挂起”状态，并失去对等连接。

- 对于运行多协议标记 (MPLS) 交换的路由器，不要禁用路由器上的快速交换。MPLS是与CEF一起使用的。因此，禁用接口上的快速交换可导致很严重的后果。

请考虑以下示例情景：



router_122 上配置访问列表为：

```
<#root>
```

```
access-list 105 permit icmp host 10.10.10.2 host 10.1.1.1 access-list 105 permit icmp host 10.1.1.1 host
```

此访问列表允许从主机router_121 (IP地址为10.10.10.2) 到主机router_123 (IP地址为10.1.1.1) 的任何Internet控制消息协议 (ICMP)数据包以及沿另一个方向传输。请务必允许数据包朝任意方向传输，否则路由器会丢弃返回的ICMP数据包。

仅删除 router_122 的一个接口上的快速交换。这意味着，您只能看到发往该接口的数据包的调试，从拦截数据包的Cisco IOS的角度来看。在调试中，此类数据包显示有“d=”。由于您尚未关闭其他接口上的快速交换，返回的数据包将不适用 `debug ip packet` 。此输出显示如何禁用快速交换：

```
<#root>
```

```
router_122(config)#
interface virtual-template 1
  router_122(config-if)#
no ip route-cache
  router_122(config-if)#
end
```

现在必须使用以前定义的访问列表(access-list 105) **debug ip packet** 激活。

<#root>

router_122#

debug ip packet

detail 105 IP packet debugging is on (detailed) for access list 105 router_122# 00:10:01: IP: s=10.1.1.1

现在删除另一个接口 (router_122上的) 上的快速交换。这意味着通过这两个接口的所有数据包现在都进行了数据包交换(这是 **debug ip packet**

<#root>

router_122(config)#

interface serial 3/0

router_122(config-if)#

no ip route-cache

router_122(config-if)#

end

router_122# 00:11:57: IP:

s=10.10.10.2

(Virtual-Access1),

d=10.1.1.1

(Serial3/0), g=172.16.1.6, len 100, forward 00:11:57:

ICMP type=8

, code=0 ! -- ICMP packet (echo) from 10.10.10.2 to 10.1.1.1 00:11:57: IP:

s=10.1.1.1

(Serial3/0),

d=10.10.10.2

(Virtual-Access1), g=10.10.10.2, len 100, forward 00:11:57:

ICMP type=0

, code=0 ! -- ICMP return packet (echo-reply) from 10.1.1.1 to 10.10.10.2 00:11:57: IP: s=10.10.10.2

请注意：debug ip packet 输出不显示任何不符合访问列表标准的数据包。有关此过程的更多信息，请参阅了解 Ping 和 Traceroute 命令。

有关如何构建访问列表的详细信息，请参阅标准 IP 访问列表日志记录。

有条件的被触发的调试

启用有条件触发调试功能时，路由器为进入或离开指定接口上的路由器的数据包生成调试消息；路由器不会为通过其他接口进入或离开路由器的数据包生成调试输出。

下面介绍条件调试的简单实施。考虑此场景：下面显示的路由器(traxbol)有两个接口（serial 0和serial 3），均运行HDLC封装。

您可以使用 **debug serial interface** 正常命令来观察所有接口上收到的HDLC keepalive。您可以观察两个接口上的 keepalive。

<#root>

traxbol#

debug serial interface

Serial network interface debugging is on traxbol# *Mar 8 09:42:34.851:

Serial0: HDLC

myseq 28, mineseen 28*, yourseen 41, line up ! -- HDLC keepalive on interface Serial 0 *Mar 8 09:42:

Serial3: HDLC

myseq 26, mineseen 26*, yourseen 27, line up ! -- HDLC keepalive on interface Serial 3 *Mar 8 09:42:

启用接口 serial 3 的条件调试。这意味着仅显示接口 serial 3 的调试。使用 **debug interface <interface_type interface_number >**命令。

<#root>

traxbol#

debug interface serial 3

Condition 1 set

请使用 **show debug condition** 命令验证有条件调试是否处于活动状态。请注意，串接口3处于激活状态。

<#root>

traxbol#

show debug condition

Condition 1: interface Se3 (1 flags triggered) Flags: Se3 traxbol#

请注意，现在只显示串接口3的调试：

```
<#root>
```

```
*Mar 8 09:43:04.855:
```

```
Serial3: HDLC
```

```
myseq 29, mineseen 29*, yourseen 30, line up *Mar 8 09:43:14.855:
```

```
Serial3: HDLC
```

```
myseq 30, mineseen 30*, yourseen 31, line up
```

请使用 `undebbug interface <interface_type interface_number>` 命令删除条件调试。建议您在删除条件触发之前关闭调试（例如，使用 `undebbug all` 命令）。通过这种办法，可避免删除条件时出现调试输出泛滥。

```
<#root>
```

```
traxbol#
```

```
undebbug interface serial 3
```

```
This condition is the last interface condition set. Removing all conditions can cause a flood of debug
```

```
y
```

```
Condition 1 has been removed traxbol
```

您现在可以观察到，系统显示了接口 serial 0 和 serial 3 的调试。

```
<#root>
```

```
*Mar 8 09:43:34.927:
```

```
Serial3: HDLC
```

```
myseq 32, mineseen 32*, yourseen 33, line up *Mar 8 09:43:44.923:
```

```
Serial10: HDLC
```

```
myseq 35, mineseen 35*, yourseen 48, line up
```



警告：某些调试操作本身是有条件的。ATM 调试就是一个例子。使用ATM调试时，您必须明确指定必须为其启用调试的接口，而不是在所有atm接口上启用调试并指定条件。

本部分介绍了将 ATM 数据包调试限制在一个子接口的正确方法：

```
<#root>
```

```
arielle-nrp2#
```

```
debug atm packet interface atm 0/0/0.1
```

```
!--- Note that you explicitly specify the sub-interface to be used for debugging ATM packets debuggi
```

```
Displaying packets on interface ATM0/0/0.1 only
```

```
arielle-nrp2# *Dec 21 10:16:51.891: ATM0/0/0.1(O): VCD:0x1 VPI:0x1 VCI:0x21 DM:0x100 SAP:AAAA CTL:03 O
```

如果您尝试在所有接口上启用 **atm debugging** 用 (带应用条件) , 则路由器在拥有大量ATM子接口时可能会挂起。下面举例说明了atm调试的不正确方法。

在本例中, 您可以看到应用了一个条件, 但也可以看到该条件没有任何影响。您仍然可以看到来自另一个接口的数据包。在本实验场景中, 您只有两个接口, 流量非常小。如果接口数量较多, 则所有接口的调试输出都会非常高, 这可能会导致路由器挂起。

```
<#root>
```

```
arielle-nrp2#
```

```
show debugging condition
```

```
Condition 1: interface ATM0/0/0.1
```

```
(1 flags triggered) Flags: ATM0/0/0.1 ! -- A condition for a specific interface. arielle-nrp2#  
debug atm packet
```

```
ATM packets debugging is on Displaying all ATM packets arielle-nrp2# *Dec 21 10:22:06.727:
```

```
ATM0/0/0.2
```

```
(O): ! -- You see debugs from interface ATM0/0/0.2, even though the condition ! -- specified ONLY ATM0/0/0.1
```

```
ATM0/0/0.1
```

```
(O): !--- You also see debugs for interface ATM0/0/0.1 as you wanted. VCD:0x1 VPI:0x1 VCI:0x21 DM:0x100
```

相关信息

- [拨号和接入技术支持](#)
- [思科技术支持和下载](#)

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。