

关于 Debug 命令的重要信息

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[警告](#)

[规则](#)

[在调试之前](#)

[获得调试输出](#)

[其他预调试任务](#)

[停止调试](#)

[使用 debug ip packet 命令](#)

[有条件的被触发的调试](#)

[相关信息](#)

简介

此页内容说明了使用Cisco IOS(r)平台上所提供的调试功能的一些基本指导原则，以及正确使用 debug ip packet 命令和有条件调试的示例。

注意： 本文不解释如何使用和解释特定调试指令和输出。有关特定 debug 命令的信息，请参考相应的Cisco debug 命令参考文献。

从调试privileged EXEC的输出发出命令一般来说包括各种各样的网际工程事件与协议状态和网络活动相关的提供诊断信息。

先决条件

要求

Cisco 建议您了解以下主题：

- 连接对路由器使用控制台，辅助和VTY端口
- 一般Cisco IOS配置问题
- 解释Cisco IOS debug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。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

警告

请谨慎使用 debug 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，如果记录启用，接入服务器能间歇地冻结，当控制台端口超载与日志消息。

在您开始debug命令前，总是请考虑此命令将生成，并且时间这能采取的输出。例如，如果有有一基本速率接口(BRI)的一个路由器， debug isdn q931很可能不会危害系统。但是，执行在一AS5800的同样调试与全双工E1配置能非常很可能生成输入可能暂停和停止响应。

在调试前，请查看您的CPU负载用show processes cpu命令。验证您有充足的CPU联机，在您开始调试前。参考[在Cisco路由器的故障排除高CPU利用率](#)关于如何处理高CPU负载的更多信息。例如，如果有有进行然后桥接的ATM接口的一个Cisco 7200路由器，根据配置的子接口数量，重新启动路由器也许使用很多其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。在这样的重要时间内启动调试功能可能会使CPU利用率急剧增加，并导致系统挂起或网络断开。

注意：当调试运行时，您通常看不到路由器提示，特别是当调试是密集的时。但是，在大多数情况下，您能使用No Debug All或Undebug All命令为了终止调试。[有关安全使用调试的更多信息，请参考“获得调试输出”一章。](#)

规则

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

在调试之前

除以上提到的点之外，请确保您了解调试的影响在平台的稳定性。您应该也考虑在路由器的哪个接口您应该连接。此部分有一些指南。

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。下面讨论每种方法的使用说明和注意事项：

控制台端口

如果在控制台连接，在正常配置下，没有额外的工作需要执行。调试输出应自动显示。但是，请确保操作日志控制台级别如期望的一样设置，并且那记录日志未用no logging console命令禁用。参考[使用调试指令](#)欲知更多信息。

警告：对路由器的控制台端口的额外的调试能造成它暂停。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权。若路由器处理发送到控制面板端口的大型调试输出，路由器可能处于挂起状态。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。下面提供这方面的更多信息。

注意：默认情况下，记录在控制台端口启用。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，Cisco建议，在正常的运行条件下，您安排**no logging console**命令一直启用并且使用其他方法获取调试。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。还要验证路由器上 no logging on 命令是否处于禁用状态。

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端口不显示启动顺序输出。连接到控制台端口为了查看启动顺序。

VTY端口

如果连接通过辅助端口或通过telnet，请键入**terminal monitor**命令。并且请验证未使用no logging on命令。

将信息记录到内部缓冲器

默认日志设备是控制台;除非另外说明所有消息在控制台显示。

若要将日志信息记录到内部缓冲器上，则使用日志缓冲路由器配置命令。这是此命令完整语法：

```
logging buffered no logging buffered
```

logging buffered 命令将日志消息复制到内部缓冲器上，而不是将这些消息写在控制面板中。缓冲器具有循环特性，新消息将覆盖旧消息。若要显示缓冲器中所记录的消息，则使用show logging特权EXEC命令。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。

提示：确保足够的内存是可用的在方框在输入缓冲区大小前。请使用show proc mem命令的Cisco IOS为了发现内存联机。

通过 no logging buffered 命令，可禁用缓冲器，并将消息写入到控制面板中(缺省条件下)。

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：

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

通过 logging 命令确定syslog服务器主机以接收日志消息。<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。

通过 no logging 命令，可以删除具有syslogs列表中指定地址的syslog服务器。

[使用调试指令](#)，关于设置系统日志服务器的更多信息，参考。

[其他预调试任务](#)

1. 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并选择相应的选项。欲知更多信息, 参考[捕获从超级终端的文本输出](#)。有关其他终端仿真器软件, 请参考软件文献。
2. 通过以下业务时间标记命令来启用毫秒(msec)时间标记:

```
router(config)#service timestamps  
debug datetime msec router(config)#service timestamps log datetime msec
```

这些命令加时标记对在格式MMM DD HH的调试: MM:SS, 指示日期和时间根据系统时钟。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, 值得注意的是, 在控制面板端口输出大量信息时, 可能不与事件的实际定时发生联系。例如, 如果启用在有200个VC的方框的**debug x25 all**, 并且输出被记录到缓冲区(使用**no logging console**和**logging buffered**命令), 在debug输出中显示的时间戳(在缓冲区内)也许不是确切的时间, 当数据包穿过接口时。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, 则使用 **no debug all** 或 **undebug all** 命令。通过**show debug**命令来验证是否已经关闭调试。

请注意, 通过 **no logging console** 和 **terminal no monitor** 命名, 只能分别使输出不在控制面板、Aux或 vty上发生, 而不能停止调试, 并因此用尽路由器资源。它不终止调试并且用完路由器资源。

[使用 debug ip packet 命令](#)

通过 **debug ip packet** 命令, 可产生有关数据包的信息, 这种数据包未经路由器进行快速转换。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。所以, 如本文所描述的那样, 只在最严格的控制下使用 **debug ip packet**。

限制 **debug ip packet** 输出的最佳办法是 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需要使用**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: 在缺省状态下, 路由器进行快速交换, 或在特定配置情况下进行CEF交换。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。若要使调试显示内容, 则需要通过**no ip route-cache** (适用于单点发送数据包)或**no ip mroute-cache**(适用于组播数据包)来禁止路由器上的快速交换。这种办法适用于传输业务的接口。通过 **show ip route** 命令来验证这种情况。

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。

下面研究一个示例方案:

access-list已配置的在router_122是:

```
access-list 105 permit icmp host 10.10.10.2 host 13.1.1.1 access-list 105 permit icmp host 13.1.1.1 host 10.10.10.2
```

此访问列表允许所有互联网控制消息协议(ICMP)数据包从主机router_121(IP地址为10.10.10.2)传输到主机router_123(IP地址为13.1.1.1), 以及传输到其他方向上。允许数据包传输到任一方向是很重要的, 否则路由器可能会丢弃返回的ICMP数据包。

只取消在一个接口的快速交换在router_122。这意味着您能为该接口是注定的数据包只看到调试, 如被看到从截断数据包的IOS的角度。从调试, 这样数据包显现“d=”。因为您未关闭在另一个接口的快速交换, 返回信息包不是受**调试ip数据包支配**。此输出显示您如何能禁用快速交换:

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

您必须当前激活与access-list及早定义的**调试ip数据包(access-list 105)**。

```
router_122#debug ip packet detail 105 IP packet debugging is on (detailed) for access list 105
router_122# 00:10:01: IP: s=13.1.1.1 (Serial3/0), d=10.10.10.2 (Virtual-Access1), g=10.10.10.2,
len 100, forward 00:10:01: ICMP type=0, code=0 ! -- ICMP packet from 13.1.1.1 to 10.10.10.2. ! -
- This packet is displayed because it matches the ! -- source and destination requirements in
access list 105 00:10:01: IP: s=13.1.1.1 (Serial3/0), d=10.10.10.2 (Virtual-Access1),
g=10.10.10.2, len 100, forward 00:10:01: ICMP type=0, code=0 00:10:01: IP: s=13.1.1.1
(Serial3/0), d=10.10.10.2 (Virtual-Access1), g=10.10.10.2, len 100, forward 00:10:01: ICMP
type=0, code=0
```

下面删除其他接口(router_122)上的快速路由。也就是通过这两个接口上的全部数据包现在是分组交换的(这是 debug ip packet所要求的):

```
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=13.1.1.1
(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 13.1.1.1 00:11:57: IP: s=13.1.1.1 (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 13.1.1.1 to 10.10.10.2 00:11:57: IP: s=10.10.10.2 (Virtual-Access1),
d=13.1.1.1 (Serial3/0), g=172.16.1.6, len 100, forward 00:11:57: ICMP type=8, code=0 00:11:57:
IP: s=13.1.1.1 (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
```

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。关于此步骤的另外的信息, 参考[了解Ping和Traceroute命令](#)。

关于如何构件访问列表的更多信息, 参考[标准IP访问列表登录](#)。

有条件的被触发的调试

当有条件地被触发的调试特点启用时, 路由器生成输入或离开在指定的接口的数据包的调试消息路由器;路由器不生成输入或离开通过一个不同的接口的数据包的调试输出。关于有条件调试的用途的信息, 参考的[有条件地被触发的调试](#)。

查看有条件调试的一个简单实施。考虑此方案: 下面所示的路由器(trabol)有两个接口(两的serial0和的序列3)运行HDLC封装。

您能使用normal debug serial interface命令为了观察在所有接口接收的HDLC Keepalive。您能观察在两个接口的Keepalive。

```
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 09:42:34.855: Serial3: HDLC myseq 26, mineseen 26*, yourseen 27, line
up ! -- HDLC keepalive on interface Serial 3 *Mar 8 09:42:44.851: Serial0: HDLC myseq 29,
mineseen 29*, yourseen 42, line up *Mar 8 09:42:44.855: Serial3: HDLC myseq 27, mineseen 27*,
```

yourseen 28, line up

接口序列的3. Enable (event)有条件调试。这意味着接口序列的3仅调试显示。请使用debug interface <interface_type interface_number>命令。

```
traxbol#debug interface serial 3 Condition 1 set
```

请使用show debug condition命令为了验证conditional调试是活跃的。请注意，串接口3处于激活状态。

```
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：

```
*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undebug接口<interface_type interface_number>命令为了删除有条件调试。推荐您关闭调试(例如，使用undebug all)，在您删除有条件的触发前。通过这种办法，可避免删除条件时出现调试输出泛滥。

```
traxbol#undebug interface serial 3 This condition is the last interface condition set. Removing all conditions may cause a flood of debugging messages to result, unless specific debugging flags are first removed. Proceed with removal? [yes/no]: y Condition 1 has been removed traxbol
```

您能现在遵守两interface serial 0的调试以及序列3显示。

```
*Mar 8 09:43:34.927: Serial3: HDLC myseq 32, mineseen 32*, yourseen 33, line up *Mar 8 09:43:44.923: Serial0: HDLC myseq 35, mineseen 35*, yourseen 48, line up
```

警告：一些调试操作独自自有条件的。示例是ATM调试。使用ATM调试您应该明确地指定调试应该启用而不是启用在所有ATM接口的调试和指定情况的接口。

此部分显示正确方式对一子接口限制调试的ATM信息包：

```
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 debugging is on 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 OUI:0080C2 TYPE:0007 Length:0x278 *Dec 21 10:16:51.891: 0000 FFFF FFFF FFFF 0010 7BB9 BDC4 0800 4500 025C 01FE 0000 FF11 61C8 0A30 *Dec 21 10:16:51.891: 4B9B FFFF FFFF 0044 0043 0248 0000 0101 0600 0015 23B7 0000 8000 0000 0000 *Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0010 7BB9 BDC3 0000 0000 0000 0000 0000 0000 *Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21 10:16:51.895: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21 10:16:51.895: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21 10:16:51.895:
```

当设法启用所有端口上的 atm debugging (通过适用的条件)时，若路由器上存在大量ATM子接口，则路由器可能处于“挂起”状态。下面举例说明了atm 调试的不正确方法。

在这种情况下您能看到情况应用，但是您也看到这没有效果。您能仍然看到从另一个接口的数据包。在此实验室情形中您只有两个接口和非常很少的数据流。如果接口数量高，则所有接口的 debug输出非常高，并且能造成路由器暂停。

```
arielle-nrp2#show debugging condition Condition 1: interface AT0/0/0.1 (1 flags triggered) Flags: AT0/0/0.1 ! -- 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 AT0/0/0.1 VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2 TYPE:000E Length:0x2F *Dec 21 10:22:06.727: 0000 0000 0180 0000 107B B9BD C400 0000 0080 0000 107B B9BD C480 0800 0014 *Dec 21 10:22:06.727: 0002 000F 0000 *Dec 21 10:22:06.727: un a *Dec 21 10:22:08.727: ATM0/0/0.2(O): VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2 TYPE:000E Length:0x2F *Dec 21 10:22:08.727: 0000 0000 0180 0000 107B B9BD C400 0000 0080 0000
```

```
107B B9BD C480 0800 0014 *Dec 21 10:22:08.727: 0002 000F 0000 *Dec 21 10:22:08.727: 11 *Dec 21
10:22:10.727: ATM0/0/0.2(O): VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2
TYPE:000E Length:0x2F *Dec 21 10:22:10.727: 0000 0000 0080 0000 107B B9BD C400 0000 0080 0000
107B B9BD C480 0800 0014 *Dec 21 10:22:10.727: 0002 000F 0000 *Dec 21 10:22:10.727: *Dec 21
10:22:12.727: ATM0/0/0.2(O): VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2
TYPE:000E Length:0x2F *Dec 21 10:22:12.727: 0000 0000 0080 0000 107B B9BD C400 0000 0080 0000
107B B9BD C480 0800 0014 *Dec 21 10:22:12.727: 0002 000F 0000 *Dec 21 10:22:12.727: *Dec 21
10:22:13.931: 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 SAP:AAAA CTL:03 OUI:0080C2 TYPE:0007 Length:0x278 *Dec 21
10:22:13.931: 0000 FFFF FFFF FFFF 0010 7BB9 BDC4 0800 4500 025C 027F 0000 FF11 6147 0A30 *Dec 21
10:22:13.931: 4B9B FFFF FFFF 0044 0043 0248 0000 0101 0600 001A 4481 0000 8000 0000 0000 *Dec 21
10:22:13.931: 0000 0000 0000 0000 0000 0000 0010 7BB9 BDC3 0000 0000 0000 0000 0000 0000 *Dec 21
10:22:13.931: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21
10:22:13.931: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21
10:22:13.931: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21
10:22:13.935: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

相关信息

- [拨号和接入技术支持](#)
- [技术支持和文档 - Cisco Systems](#)