了解Ping和Traceroute命令

目录

简介

先决条件

- 要求
- 使用的组件
- 规则

背景信息

ping 命令

排除Ping故障

- 路由器
- 接口关闭
- 访问列表命令
- 地址解析协议 (ARP) 问题
- 延迟
- 正确的源地址
- 输入队列丢包数较高

traceroute 命令

性能

使用 Debug 命令

相关信息

简介

本文档介绍如何在思科路由器上使用 ping 和 traceroute 命令。

先决条件

要求

本文档没有任何特定的要求。

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。

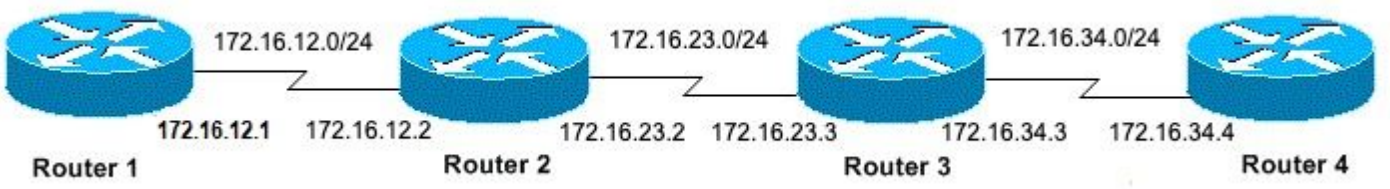
规则

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

> 注意：生产路由器上使用的任何debug命令都可能导致严重问题。在发出 debug 命令之前，请阅读使用 Debug 命令部分。

本文档中的示例使用此基本网络拓扑：



网络拓扑

ping 命令

ping 命令是一种用来对设备可访问性进行故障排除的常用方法。它使用一系列的 Internet 控制消息协议 (ICMP) 回声消息以确定以下内容：

- 远程主机是处于活动还是非活动状态。
- 用于与主机通信的往返延迟。
- 数据包丢失。

ping 命令首先向一个地址发送一个回声请求数据包，然后等待应答。只有在以下情况下，ping 才是成功的：

- 回声请求到达目标，并且
- 目标能够在称为“超时”的预先确定的时间内将回声应答返回到源。在 Cisco 路由器上，此超时默认值为两秒。

无法更改 ping 数据包的 TTL 值。

下一个代码示例显示启用了 debug ip packet detail 命令之后的 ping 命令。

⚠ 警告：在生产路由器上使用debug ip packet detail命令时，可能导致高CPU利用率。这可能会导致性能严重下降或网络中断。

<#root>

```
Router1#

debug ip packet detail

IP packet debugging is on (detailed)

Router1#

ping 172.16.12.2


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms

Router1#
Jan 20 15:54:47.487: IP: s=172.16.12.1 (local), d=172.16.12.2 (Serial0), len 100,
  sending
Jan 20 15:54:47.491:

ICMP type=8

, code=0

!--- This is the ICMP packet 172.16.12.1 sent to 172.16.12.2.
!--- ICMP type=8 corresponds to the echo message.

Jan 20 15:54:47.523: IP: s=172.16.12.2 (Serial0), d=172.16.12.1 (Serial0), len 100,
  rcvd 3
Jan 20 15:54:47.527:

ICMP type=0

, code=0

!--- This is the answer we get from 172.16.12.2. !--- ICMP type=0 corresponds to the echo reply message
```

```
!--- By default, the repeat count is five times, so there will be five
!--- echo requests, and five echo replies.
```

ICMP-type 的可能值：

ICMP 类型	文字
0	回声应答
3	目的地不可达代码 0 = 网络不可达 1 = 主机不可达 2 = 协议不可达 3 = 端口不可达 4 = 需要分段并设置 DF 5 = 源路由失败
4	source-quench
5	重定向代码 0 = 网络的重定向数据报 1 = 主机的重定向数据报 2 = 服务和网络类型的重定向数据报 3 = r服务和主机类型的重定向数据报
6	alternate-address
8	echo
9	路由器公告
10	router-solicitation
11	超时代码 0 = 传输中超出生存时间 1 = 超出片段重组时间
12	parameter-problem
13	timestamp-request
14	timestamp-reply
15	信息请求
16	information-reply
17	mask-request
18	mask-reply
31	conversion-error
32	mobile-redirect

Ping 工具可能输出的字符

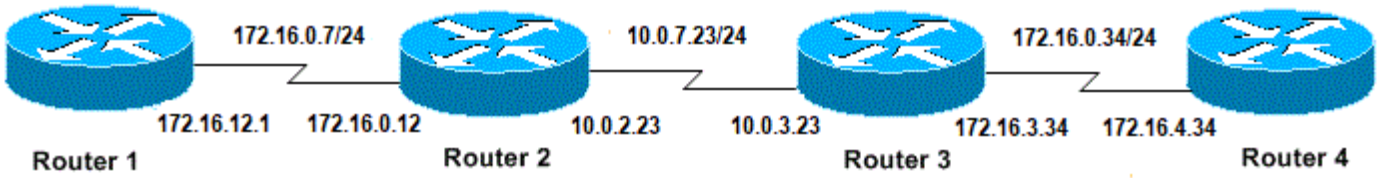
字符	描述
!	每个感叹号表示收到应答。
.	每个句号表示等待应答时网络服务器超时。
U	收到了目标不可达错误 PDU。
问	源抑制（目标太忙）。
M	无法分段。
?	未知数据包类型。
&	超出了数据包的有效期。

排除Ping故障

如果您无法成功 ping 通某个 IP 地址，请考虑此部分中列出的原因。

路由器

以下是未成功的 ping 操作尝试示例，可以根据这些示例确定出问题所在，以及如何解决问题。此示例结合此网络拓扑图显示：



路由器问题

<#root>

Router1#

```
!
interface Serial0
ip address 172.16.12.1 255.255.255.0
no fair-queue
clockrate 64000
!
```

Router2#

```
!
interface Serial0
ip address 172.16.23.2 255.255.255.0
no fair-queue
clockrate 64000
!
interface Serial1
ip address 172.16.12.2 255.255.255.0
!
```

Router3#

```
!
interface Serial0
```

```
ip address 172.16.34.3 255.255.255.0
no fair-queue
!
interface Serial1
ip address 172.16.23.3 255.255.255.0
!
```

```
Router4#
```

```
!
interface Serial0
ip address 172.16.34.4 255.255.255.0
no fair-queue
clockrate 64000
!
```

尝试从 Router1 对 Router4 执行 ping 操作：

<#root>

```
Router1#
```

```
ping 172.16.34.4
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:
.....
```

```
Success rate is 0 percent (0/5)
```

成果：

<#root>

```
Router1#
```

```
debug ip packet
```

```
IP packet debugging is on
```

警告：在生产路由器上使用debug ip packet命令时，可能导致高CPU利用率。这可能会导致

⚠ 性能严重下降或网络中断。

<#root>

```
Router1#

ping 172.16.34.4



Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:

Jan 20 16:00:25.603: IP: s=172.16.12.1 (local), d=172.16.34.4, len 100, unroutable.
Jan 20 16:00:27.599: IP: s=172.16.12.1 (local), d=172.16.34.4, len 100, unroutable.
Jan 20 16:00:29.599: IP: s=172.16.12.1 (local), d=172.16.34.4, len 100, unroutable.
Jan 20 16:00:31.599: IP: s=172.16.12.1 (local), d=172.16.34.4, len 100, unroutable.
Jan 20 16:00:33.599: IP: s=172.16.12.1 (local), d=172.16.34.4, len 100, unroutable.
Success rate is 0 percent (0/5)
```

由于Router1上没有运行任何路由协议，因此它不知道将数据包发送到何处，从而导致出现“不可路由”消息。

向 Router1 添加一个静态路由：

<#root>

```
Router1#

configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#

ip route 0.0.0.0 0.0.0.0 Serial0
```

成果：

<#root>

```
Router1#

debug ip packet detail

IP packet debugging is on (detailed)
```

```
Router1#

ping 172.16.34.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)

Jan 20 16:05:30.659: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
  sending
Jan 20 16:05:30.663:     ICMP type=8, code=0
Jan 20 16:05:30.691: IP: s=172.16.12.2 (Serial0), d=172.16.12.1 (Serial0), len 56,
  rcvd 3
Jan 20 16:05:30.695:     ICMP type=3, code=1
Jan 20 16:05:30.699: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
  sending
Jan 20 16:05:30.703:     ICMP type=8, code=0
Jan 20 16:05:32.699: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
  sending
Jan 20 16:05:32.703:     ICMP type=8, code=0
Jan 20 16:05:32.731: IP: s=172.16.12.2 (Serial0), d=172.16.12.1 (Serial0), len 56,
  rcvd 3
Jan 20 16:05:32.735:     ICMP type=3, code=1
Jan 20 16:05:32.739: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
  sending
Jan 20 16:05:32.743:     ICMP type=8, code=0
```

检查 Router2 上的问题：

<#root>

```
Router2#

debug ip packet detail

IP packet debugging is on (detailed)

Router2#
Jan 20 16:10:41.907: IP: s=172.16.12.1 (Serial1), d=172.16.34.4, len 100, unroutable
Jan 20 16:10:41.911:     ICMP type=8, code=0
Jan 20 16:10:41.915: IP: s=172.16.12.2 (local), d=172.16.12.1 (Serial1), len 56, sending
Jan 20 16:10:41.919:

ICMP type=3, code=1

Jan 20 16:10:41.947: IP: s=172.16.12.1 (Serial1), d=172.16.34.4, len 100, unroutable
Jan 20 16:10:41.951:     ICMP type=8, code=0
Jan 20 16:10:43.943: IP: s=172.16.12.1 (Serial1), d=172.16.34.4, len 100, unroutable
Jan 20 16:10:43.947:     ICMP type=8, code=0
Jan 20 16:10:43.951: IP: s=172.16.12.2 (local), d=172.16.12.1 (Serial1), len 56, sending
Jan 20 16:10:43.955:     ICMP type=3, code=1
Jan 20 16:10:43.983: IP: s=172.16.12.1 (Serial1), d=172.16.34.4, len 100, unroutable
Jan 20 16:10:43.987:     ICMP type=8, code=0
Jan 20 16:10:45.979: IP: s=172.16.12.1 (Serial1), d=172.16.34.4, len 100, unroutable
```

```
Jan 20 16:10:45.983:     ICMP type=8, code=0
Jan 20 16:10:45.987: IP: s=172.16.12.2 (local), d=172.16.12.1 (Serial1), len 56, sending
Jan 20 16:10:45.991:     ICMP type=3, code=1
```

Router1向Router2正确发送了其数据包，但Router2不知道如何访问地址172.16.34.4。Router2向Router1发送回一条“unreachable ICMP”消息。

在Router2和Router3上配置动态路由协议(本例中使用路由信息协议(RIP)):

```
Router2#
router rip
 network 172.16.0.0
 no auto-summary
Router3#
router rip
 network 172.16.0.0
 no auto-summary
```

成果：

<#root>

```
Router1#

debug ip packet

IP packet debugging is on

Router1#

ping 172.16.34.4


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:

Jan 20 16:16:13.367: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending.
Jan 20 16:16:15.363: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending.
Jan 20 16:16:17.363: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending.
Jan 20 16:16:19.363: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending.
Jan 20 16:16:21.363: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending.
Success rate is 0 percent (0/5)
```

Router1 向 Router4 发送数据包，但 Router4 未发回应答。

Router4 上可能存在的问题：

<#root>

```
Router4#

debug ip packet

IP packet debugging is on

Router4#
Jan 20 16:18:45.903: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:45.911: IP: s=172.16.34.4 (local), d=172.16.12.1, len 100,

unroutable

Jan 20 16:18:47.903: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:47.907: IP: s=172.16.34.4 (local), d=172.16.12.1, len 100, unroutable
Jan 20 16:18:49.903: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:49.907: IP: s=172.16.34.4 (local), d=172.16.12.1, len 100, unroutable
Jan 20 16:18:51.903: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:51.907: IP: s=172.16.34.4 (local), d=172.16.12.1, len 100, unroutable
Jan 20 16:18:53.903: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:53.907: IP: s=172.16.34.4 (local), d=172.16.12.1, len 100, unroutable
```

Router4 收到 ICMP 数据包，并尝试应答 172.16.12.1，但因没有通往此网络的路由而失败。

向 Router4 添加一个静态路由：

<#root>

```
Router4(config)#

ip route 0.0.0.0 0.0.0.0 Serial0
```

现在双方都可以互相访问：

<#root>

```
Router1#
```

```
ping 172.16.34.4


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/35/36 ms
```

接口关闭

这是指接口停止并不再工作的情况。下一个示例中尝试从 Router1 对 Router4 执行 ping 操作：

<#root>

```
Router1#

ping 172.16.34.4


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
```

由于路由正确，所以对问题进行分步故障排除。尝试对 Router2 执行 ping 操作：

<#root>

```
Router1#

ping 172.16.12.2



Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

在上一个示例中，问题出在Router2和Router3之间。一种可能是Router3上的串行接口已关闭：

<#root>

```
Router3#
```

```
show ip interface brief

Serial0   172.16.34.3    YES manual up                      up
Serial1   172.16.23.3    YES manual administratively down   down
```

这很容易修复：

```
<#root>

Router3#

configure terminal


Enter configuration commands, one per line.  End with CNTL/Z.
Router3(config)#

interface serial1


Router3(config-if)#

no shutdown

Router3(config-if)#
Jan 20 16:20:53.900: %LINK-3-UPDOWN: Interface Serial1, changed state to up
Jan 20 16:20:53.910: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1,
  changed state to up
```

访问列表命令

在此场景中，仅允许 Telnet 流量通过接口 Serial0 进入 Router4。

```
<#root>

Router4(config)#

access-list 100 permit tcp any any eq telnet

Router4(config)#

interface serial0

Router4(config-if)#

ip access-group 100 in
```

```
Router1#

configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#

access-list 100 permit ip host 172.16.12.1 host 172.16.34.4

Router1(config)#

access-list 100 permit ip host 172.16.34.4 host 172.16.12.1

Router1(config)#

end

Router1#

debug ip packet 100


IP packet debugging is on
Router1#

debug ip icmp

ICMP packet debugging is on
```

尝试对 Router4 执行 ping 操作：

<#root>

```
Router1#

ping 172.16.34.4



Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)

Jan 20 16:34:49.207: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending
Jan 20 16:34:49.287: IP: s=172.16.34.4 (Serial0), d=172.16.12.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:34:49.291: ICMP: dst (172.16.12.1)
```

```
administratively prohibited unreachable

 rcv from 172.16.34.4
Jan 20 16:34:49.295: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending
Jan 20 16:34:51.295: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending
Jan 20 16:34:51.367: IP: s=172.16.34.4 (Serial0), d=172.16.12.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:34:51.371: ICMP: dst (172.16.12.1) administratively prohibited unreachable
 rcv from 172.16.34.4
Jan 20 16:34:51.379: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending
```

access-list 命令末尾始终有一个隐式deny all。这意味着进入Router4上Serial 0接口的ICMP数据包被拒绝，而Router 4会向原始数据包的源发送ICMP“administratively prohibited unreachable”消息，如debug 消息所示。解决方案是在 access-list 命令中添加以下命令行：

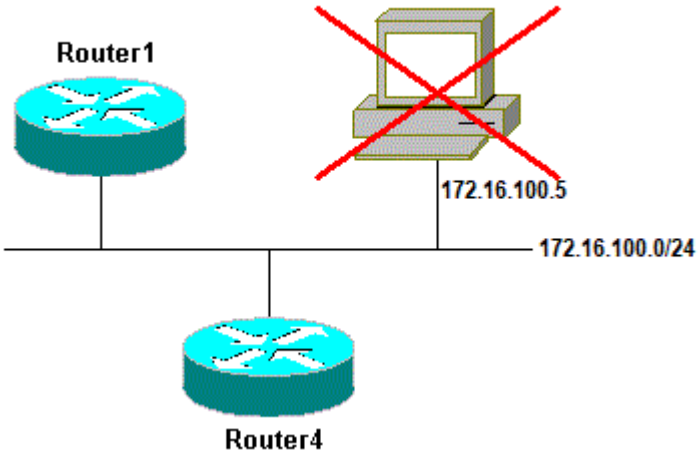
<#root>

```
Router4(config)#

access-list 100 permit icmp any any
```

地址解析协议 (ARP) 问题

在此场景中，以太网连接如下：



地址解析协议问题

<#root>

```
Router4#

ping 172.16.100.5


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.100.5, timeout is 2 seconds:

Jan 20 17:04:05.167: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:05.171: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,


encapsulation failed


.
Jan 20 17:04:07.167: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:07.171: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:09.175: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:09.183: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:11.175: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:11.179: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:13.175: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:13.179: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  encapsulation failed.
Success rate is 0 percent (0/5)
Router4#
```

在本例中，由于出现“encapsulation failed”（封装失败）消息，ping 无法生效。这意味着路由器知道它必须在哪个接口上发送数据包，但不知道如何发送。在这种情况下，您需要了解地址解析协议 (ARP) 的工作原理。

ARP 是用于将第 2 层地址（MAC 地址）映射到第 3 层地址（IP 地址）的协议。 您可以使用 show arp 命令进行检查：

<#root>

```
Router4#

show arp


Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  172.16.100.4               -   0000.0c5d.7a0d  ARPA   Ethernet0
Internet  172.16.100.7              10   0060.5cf4.a955  ARPA   Ethernet0
```

回到“encapsulation failed”问题，但这次启用 debug arp 命令：

<#root>

```
Router4#

debug arp

ARP packet debugging is on

Router4#

ping 172.16.100.5



Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.100.5, timeout is 2 seconds:

Jan 20 17:19:43.843: IP ARP: creating incomplete entry for IP address: 172.16.100.5
  interface Ethernet0
Jan 20 17:19:43.847: IP ARP: sent req src 172.16.100.4 0000.0c5d.7a0d,


dst 172.16.100.5 0000.0000.0000 Ethernet0.

Jan 20 17:19:45.843: IP ARP: sent req src 172.16.100.4 0000.0c5d.7a0d,
                 dst 172.16.100.5 0000.0000.0000 Ethernet0.
Jan 20 17:19:47.843: IP ARP: sent req src 172.16.100.4 0000.0c5d.7a0d,
                 dst 172.16.100.5 0000.0000.0000 Ethernet0.
Jan 20 17:19:49.843: IP ARP: sent req src 172.16.100.4 0000.0c5d.7a0d,
                 dst 172.16.100.5 0000.0000.0000 Ethernet0.
Jan 20 17:19:51.843: IP ARP: sent req src 172.16.100.4 0000.0c5d.7a0d,
                 dst 172.16.100.5 0000.0000.0000 Ethernet0.
Success rate is 0 percent (0/5)
```

上一个输出显示 Router4 广播数据包并将其发送到以太网广播地址 FFFF.FFFF.FFFF。这里，0000.0000.0000表示Router4查找目标172.16.100.5的MAC地址。由于在本示例中请求ARP时它不知道MAC地址，因此它使用0000.000.000作为从接口Ethernet 0发出的广播帧中的占位符，并询问哪个MAC地址对应于172.16.100.5。如果没有答案，则MAC与show arp 输出中的IP地址对应的命令标记为不完整：

<#root>

```
Router4#

show arp

Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  172.16.100.4               -   0000.0c5d.7a0d  ARPA   Ethernet0
```

```
Internet  172.16.100.5              0   Incomplete      ARPA

Internet  172.16.100.7              2   0060.5cf4.a955  ARPA   Ethernet0
```

在一个预先确定的时段之后，将从 ARP 表中清除此不完整条目。只要 MAC 地址不在 ARP 表中，ping 操作就会因“encapsulation failed”而失败。

延迟

默认情况下，如果您在两秒内未从远程端收到应答，则 ping 将失败：

```
<#root>

Router1#

ping 172.16.12.2



Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2,

timeout is 2 seconds:

.....
Success rate is 0 percent (0/5)
```

在具有低速链路或较长延迟的网络上，两秒是不够的。要更改此默认值，您可以使用扩展 ping：

```
<#root>

Router1#

ping

Protocol [ip]:
Target IP address:  172.16.12.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:

30

Extended commands [n]:
Sweep range of sizes [n]:
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 30 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1458/2390/6066 ms
```

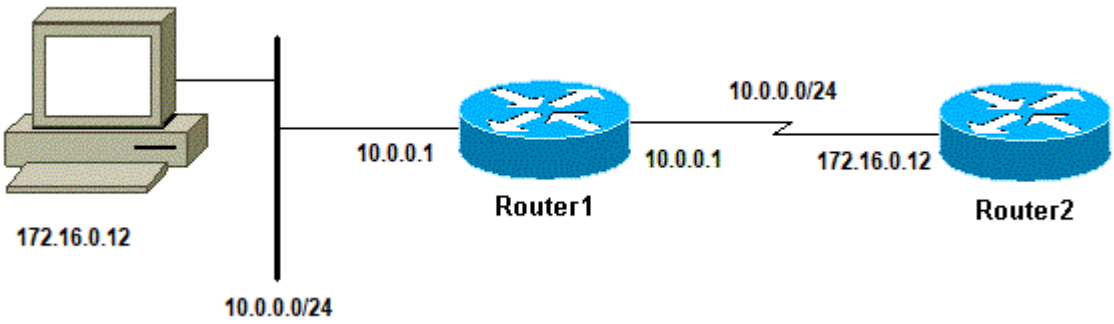
有关扩展 ping 命令的详细信息，请参阅[了解扩展 ping 和扩展 traceroute 命令](#)。

在上一个示例中，当超时增加时，ping 操作成功。

注意：平均往返时间超过两秒。

正确的源地址

此示例是一个常见场景：



正确的源地址

在 Router1 上添加一个 LAN 接口：

<#root>

```
Router1(config)#
```

interface ethernet0

```
Router1(config-if)#
```

ip address 10.0.0.1 255.255.255.0

从LAN上的站点，可以ping通Router1。从Router1可以ping通Router2。但从LAN上的站点，不能ping通Router2。

可以从 Router1 ping 通 Router2，是因为默认情况下，您将使用传出接口的 IP 地址作为 ICMP 数据包中的源地址。Router2 没有关于此新 LAN 的信息。如果必须应答此网络发来的数据包，它不知道该如何处理。

<#root>

```
Router1#

debug ip packet

IP packet debugging is on
```

警告：在生产路由器上使用debug ip packet命令时，可能导致高CPU利用率。这可能会导致性能严重下降或网络中断。

<#root>

```
Router1#

ping 172.16.12.2



Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/7/9 ms
Router1#

Jan 20 16:35:54.227: IP: s=172.16.12.1 (local), d=172.16.12.2 (Serial0), len 100, sending
Jan 20 16:35:54.259: IP: s=172.16.12.2 (Serial0), d=172.16.12.1 (Serial0), len 100, rcvd 3
```

上一个输出示例有效，因为发送的数据包的源地址是172.16.12.1。要模拟来自LAN的数据包，需要使用扩展ping:

<#root>

```
Router1#

ping

Protocol [ip]:
```

```
Target IP address: 172.16.12.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface:
```

10.0.0.1

```
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:

Jan 20 16:40:18.303: IP: s=10.0.0.1 (local), d=172.16.12.2 (Serial0), len 100,
 sending.
Jan 20 16:40:20.303: IP: s=10.0.0.1 (local), d=172.16.12.2 (Serial0), len 100,
 sending.
Jan 20 16:40:22.303: IP: s=10.0.0.1 (local), d=172.16.12.2 (Serial0), len 100,
 sending.
Jan 20 16:40:24.303: IP: s=10.0.0.1 (local), d=172.16.12.2 (Serial0), len 100,
 sending
Jan 20 16:40:26.303: IP: s=10.0.0.1 (local), d=172.16.12.2 (Serial0), len 100,
 sending.
Success rate is 0 percent (0/5)
```

这一次的源地址是 10.0.0.1，结果行不通。数据包已发送，但未收到响应。要解决此问题，请在Router2中添加指向10.0.0.0的路由。基本规则是，被ping的设备还必须知道如何向ping的源发送应答。

输入队列丢包数较高

数据包输入路由器时，路由器尝试在中断级别进行转发。如果无法在相应的缓存表中找到匹配项，数据包将在传入接口的输入队列中排队以等待处理。将始终处理某些数据包，但是需要适当的配置并在稳定的网络中进行，已处理数据流包的速率决不能导致输入队列拥塞。如果输入队列已满，将丢弃数据包。

虽然接口已启用，但由于输入队列丢包数较高，您无法 ping 通设备。您可以使用 show interface 命令检查输入丢包数。

<#root>

```
Router1#
```

show interface Serial0/0/0

```
Serial0/0/0 is up, line protocol is up

  MTU 1500 bytes, BW 1984 Kbit, DLY 20000 usec,
     reliability 255/255, txload 69/255, rxload 43/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:02, output 00:00:00, output hang never
  Last clearing of "show interface" counters 01:28:49


Input queue: 76/75/5553/0


 (size/max/drops/flushes);
     Total output drops: 1760
  Queueing strategy: Class-based queueing
  Output queue: 29/1000/64/1760 (size/max total/threshold/drops)
     Conversations  7/129/256 (active/max active/max total)
     Reserved Conversations 4/4 (allocated/max allocated)
     Available Bandwidth 1289 kilobits/sec


!--- Output supressed
```

从输出中可以看出，输入队列丢包数较高。要排查输入/输出队列丢包情况的故障，请参阅对输入队列丢包和输出队列丢包进行故障排除。

traceroute 命令

traceroute 命令用于搜索数据包在传输到它们的目标时实际采用的路由。设备（例如路由器或 PC）向远程主机的无效端口地址发送用户数据报协议 (UDP) 数据报序列。

发送了三个数据报，每一个数据报都具有一个值设置为 1 的存活时间 (TTL) 字段。TTL 值为 1，会导致数据报到达路径中的第一个路由器后就立即“超时”。然后，该路由器会以一条 ICMP 超时消息 (TEM) 进行响应，指示该数据报已过期。

现在将发送另外三条 UDP 消息，每条 UDP 消息的 TTL 值设置为 2，这导致第二台路由器返回 ICMP TEM。此过程将持续，直到数据包实际到达另一个目标。由于这些数据报尝试访问目标主机上的无效端口，因此会返回ICMP端口不可达消息，并指示不可达端口；此事件会对 Traceroute 程序发出它已完成的信号。

此命令的真正目的是记录每条 ICMP 超时消息的源，以提供对数据包到达目标所采用的路径的跟踪。

<#root>

```
Router1#

traceroute 172.16.34.4

Type escape sequence to abort.
Tracing the route to 172.16.34.4

  1 172.16.12.2 4 msec 4 msec 4 msec
  2 172.16.23.3 20 msec 16 msec 16 msec
  3 172.16.34.4 16 msec *  16 msec

Jan 20 16:42:48.611: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.615:     UDP src=39911, dst=

33434

Jan 20 16:42:48.635: IP: s=172.16.12.2 (Serial0), d=172.16.12.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:42:48.639:

ICMP type=11, code=0

!--- ICMP Time Exceeded Message from Router2.

Jan 20 16:42:48.643: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.647:     UDP src=34237, dst=33435
Jan 20 16:42:48.667: IP: s=172.16.12.2 (Serial0), d=172.16.12.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:42:48.671:     ICMP type=11, code=0
Jan 20 16:42:48.675: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.679:     UDP src=33420, dst=33436
Jan 20 16:42:48.699: IP: s=172.16.12.2 (Serial0), d=172.16.12.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:42:48.703:     ICMP type=11, code=0
```

这是使用TTL=1发送的第一个数据包序列。在本例中，第一个路由器(Router2(172.16.12.2))丢弃数据包，然后向源(172.16.12.1)发回type=11 ICMP消息。这对应于超时消息。

<#root>

```
Jan 20 16:42:48.707: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.711:     UDP src=35734, dst=33437
Jan 20 16:42:48.743: IP: s=

172.16.23.3

 (Serial0), d=172.16.12.1 (Serial0), len 56,
```

```
 rcvd 3
Jan 20 16:42:48.747:
```

ICMP type=11, code=0

!--- ICMP Time Exceeded Message from Router3.

```
Jan 20 16:42:48.751: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.755:     UDP src=36753, dst=33438
Jan 20 16:42:48.787: IP: s=172.16.23.3 (Serial0), d=172.16.12.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:42:48.791:     ICMP type=11, code=0
Jan 20 16:42:48.795: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.799:     UDP src=36561, dst=33439
Jan 20 16:42:48.827: IP: s=172.16.23.3 (Serial0), d=172.16.12.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:42:48.831:     ICMP type=11, code=0
```

对于TTL=2的Router3(172.16.23.3)，会出现相同的过程：

<#root>

```
Jan 20 16:42:48.839: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.843:     UDP src=34327, dst=33440
Jan 20 16:42:48.887: IP: s=
```

172.16.34.4

```
 (Serial0), d=172.16.12.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:42:48.891:
```

ICMP type=3, code=3

!--- Port Unreachable message from Router4.

```
Jan 20 16:42:48.895: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.899:     UDP src=37534, dst=33441
Jan 20 16:42:51.895: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:51.899:     UDP src=37181, dst=33442
Jan 20 16:42:51.943: IP: s=172.16.34.4 (Serial0), d=172.16.12.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:42:51.947:     ICMP type=3, code=3
```

TTL=3 时，最终到达 Router4。这次，由于端口无效，因此 Router4 向 Router1 发送回 type=3

（目标不可达消息）和 code=3（表示端口不可达）的 ICMP 消息。

下表列出了 traceroute 命令输出中可能出现的字符。

IP Traceroute 文本字符

字符	描述
nn msec	对于每个节点，指定数量的探测信号的往返时间，以毫秒为单位
*	探测信号超时
A	管理性禁止（例如访问列表）
问	源抑制（目标太忙）
I	用户中断测试
U	端口不可达
H	主机不可达
n	网络不可达
P	协议不可达
T	超时
?	未知数据包类型

性能

您可以使用 ping 和 traceroute 命令获取往返时间 (RTT)。这是发送响应信息包并获得回复所需的时间。有助于大致了解链路上的延迟情况。但是，这些数字用于性能评估不够精确。

数据包目标是路由器本身时，此数据包必须通过进程进行交换。处理器必须处理此数据包中的信息并发回应答。这不是路由器的主要目标。通过定义，将路由器构建为对数据包进行路由。应答 ping 操作是一种尽力而为的服务。

为了说明这一点，以下是从 Router1 向 Router2 执行 ping 操作的示例：

<#root>

```
Router1#

ping 172.16.12.2



Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

RTT 大约是四毫秒。在 Router2 上启用了一些处理密集的功能之后，尝试从 Router1 ping Router2。

<#root>

```
Router1#

ping 172.16.12.2

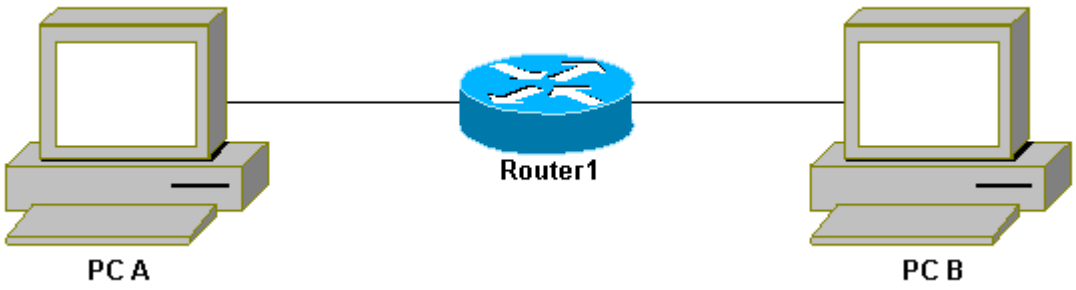


Type

escape sequence


 to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/25/28 ms
```

此时，RTT 已显著增加。Router2 很繁忙，其优先事项不是应答 ping。有一种更好的方法可测试路由器性能：使用通过路由器的流量。



通过路由器的流量

此流量随后将进行快速切换，并由具有最高优先级的路由器来处理。以下基本网络将说明这种情况：



三路由器基本网络

从 Router1 对 Router3 执行 ping 操作：

<#root>

```
Router1#

ping 172.16.23.3



Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
```

网络流量通过 Router2 并进行快速切换。在 Router2 上启用进程密集型功能：

<#root>

```
Router1#

ping 172.16.23.3



Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms
```

几乎没有任何区别。这是因为，在 Router2 上，数据包现在是在中断级别处理的。

使用 Debug 命令

使用 debug 命令之前，请参阅有关 Debug 命令的重要信息。

本文中使用的各种 debug 命令显示了使用 ping 或 traceroute 命令时发生的情况。这些命令可以帮助您排查问题。但是，在生产环境中使用这些 debug 命令时必须十分谨慎。如果您的 CPU 功能不够强，或者您有大量的需要通过进程交换的数据包，则它们可能很容易地使您的设备停止运行。有两个方法可以将 debug 命令对路由器的影响减到最小。一个方法是使用访问列表以减少您希望监控的特定数据流的流量。

例如：

<#root>

```
Router4#

debug ip packet ?

  <1-199>      Access list
  <1300-2699>  Access list (expanded range)
  detail       Print more debugging detail

Router4#

configure terminal

Router4(config)#

access-list 150 permit ip host 172.16.12.1 host 172.16.34.4

Router4(config)#^

Z



Router4#

debug ip packet 150

IP packet debugging is on for access list 150

Router4#

show debug

Generic IP:
  IP packet debugging is on for access list 150

Router4#

show access-list



Extended IP access list 150
    permit ip host 172.16.12.1 host 172.16.34.4 (5 matches)
```

使用此配置时，Router4只打印与访问列表150匹配的调试消息。从Router1执行ping操作会导致显示以下消息：

```
Router4#
Jan 20 16:51:16.911: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
```

```
 rcvd 3
Jan 20 16:51:17.003: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:51:17.095: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:51:17.187: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:51:17.279: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
```

Router4 并不能为此问题提供答案，因为这些数据包与访问列表不匹配。要查看这些信息，请添加：

<#root>

```
Router4(config)#

access-list 150 permit ip host 172.16.12.1 host 172.16.34.4

Router4(config)#

access-list 150 permit ip host 172.16.34.4 host 172.16.12.1
```

成果：

```
Jan 20 16:53:16.527: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:53:16.531: IP: s=172.16.34.4 (local), d=172.16.12.1 (Serial0), len 100,
 sending
Jan 20 16:53:16.627: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:53:16.635: IP: s=172.16.34.4 (local), d=172.16.12.1 (Serial0), len 100,
 sending
Jan 20 16:53:16.727: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:53:16.731: IP: s=172.16.34.4 (local), d=172.16.12.1 (Serial0), len 100,
 sending
Jan 20 16:53:16.823: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:53:16.827: IP: s=172.16.34.4 (local), d=172.16.12.1 (Serial0), len 100,
 sending
Jan 20 16:53:16.919: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:53:16.923: IP: s=172.16.34.4 (local), d=172.16.12.1 (Serial0), len 100,
 sending
```

若要降低 debug 命令的影响，另一种方法是缓冲调试消息，并在 debug 关闭后使用 show log 命令显示这些消息：

<#root>

```
Router4#

configure terminal

Router4(config)#

no logging console

Router4(config)#

logging buffered 5000

Router4(config)#^

Z



Router4#

debug ip packet


IP packet debugging is on
Router4#

ping 172.16.12.1



Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/36/37 ms

Router4#

undebug all

All possible debugging has been turned off

Router4#

show log


Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged
    Buffer logging: level debugging, 61 messages logged
    Trap logging: level informational, 59 message lines logged

Log Buffer (5000 bytes):
```

```
Jan 20 16:55:46.587: IP: s=172.16.34.4 (local), d=172.16.12.1 (Serial0), len 100,
 sending
Jan 20 16:55:46.679: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
```

ping和traceroute命令提供基本的可达性和路径信息，网络工程师通常使用这些信息对网络访问问题进行故障排除。

相关信息

- 了解扩展 ping 和扩展 traceroute 命令
- 技术支持 - Cisco Systems

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）。