使用操作系统的 traceroute 命令

目录

简介

使用 **traceroute 命令可以返回数据包所经过的跃点序列，从而确定数据包从给定源到达目的地所通过的路径。**您的主机操作系统（例如 Linux 或 Microsoft (MS) Windows）以及 Cisco IOS® 软件都附带此实用程序。

先决条件

要求

本文读者应具备以下某个操作系统的基础知识：

- Cisco IOS 软件
- Linux
- Microsoft Windows

使用的组件

本文档中的信息适用于以下软件和硬件版本：

- 运行 Cisco IOS 软件版本 12.2(27) 的 Cisco 路由器

- 运行 Red Hat Linux 版本 9 的 PC
- 运行 MS Windows 2000 的 PC

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。

规则

有关文件规则的更多信息请参见“ Cisco技术提示规则”。

一般操作

如果在源设备（例如主机或作为主机的路由器）上执行 **traceroute ip-address 命令，它会将 IP 数据包连同一个递增至最大指定跃点计数的存活时间 (TTL) 值发往目的地。**默认为 30。通常，在转发这些数据包时，通往目的地的路径中的每个路由器都会将 TTL 字段递减一个单位。当路径中间的一个路由器发现 TTL = 1 的数据包时，会向源发出一条 Internet Control Message Protocol (ICMP)“time exceeded”消息进行响应。通过此消息，源可以知道数据包将该特定路由器作为一个跃点。

在本文档讨论的各种操作系统中，**traceroute 命令的实现方式有所不同。**

Cisco IOS 和 Linux

初始 User Datagram Protocol (UDP) 数据报探报的 TTL 设置为 1（或为用户在 extended traceroute 命令中指定的最小 TTL）。初始数据报探报的目的地 UDP 端口设置为 33434（或为在扩展 traceroute 命令输出中指定的端口）。 扩展 traceroute 命令是普通 **traceroute 命令的变体，该命令允许对 traceroute 操作（如 TTL）使用的参数默认值以及目的地端口号进行修改。**有关如何使用扩展 traceroute 命令的更多信息，请参阅使用扩展 ping 和扩展 traceroute 命令。初始数据报探报的源 UDP 端口随机使用，并且与 0x8000 执行逻辑 OR 运算（从而确保源端口最小为 0x8000）。 以下步骤说明在启动 UDP 数据报时发生的情况：

注意：参数是可配置的。此示例从 n = 1 开始，到 n = 3 结束。

1. UDP 数据报在调度时 TTL = 1，目的地 UDP 端口 = 33434，源端口随机使用。
2. UDP 目的地端口递增，源 UDP 端口随机使用，调度第二个数据报。
3. 对最多三个探报重复步骤 2（或执行在 traceroute 命令输出中要求的次数）。 对于发送的每个探报，您都会收到“TTL exceeded”消息，该消息用于生成到目的地主机的逐步路径。
4. 如果收到 ICMP“time exceeded”消息，则 TTL 会递增，并使用递增的目的地端口号重复此循环。您也会收到以下消息之一：ICMP 类型 3，编码 3（“destination unreachable”、“port unreachable”）消息，表明到达主机。“host unreachable”、“net unreachable”、“maximum TTL exceeded”或“timeout”类型的消息，表示重新发送了探报。

Cisco 路由器采用随机源端口和递增目的地端口（用以区分不同的探报）发送 UDP 探报数据包。Cisco 路由器将 ICMP 消息“time exceeded”发送回收到 UDP/ICMP 数据包的源。

Linux traceroute 命令类似于 Cisco 路由器实现。不过，它使用固定源端口。**traceroute 命令中的 -n 选项用于避免对名称服务器的请求。**

Microsoft Windows

MS Windows **tracert 命令使用 ICMP Echo 请求数据报而不是作为探报的 UDP 数据报。**ICMP Echo 请求使用递增 TTL 启动，Cisco IOS 和 Linux 中也会发生同样的操作。使用 ICMP Echo 请求数据

报的意义在于，最后一跳不依赖于来自目的地主机的 ICMP“unreachable”消息响应，而是依赖于 ICMP Echo 回复消息。

命令语法为：

```
tracert [-d] [-h maximum_hops] [-j computer-list] [-w timeout] target_name
```

下表介绍命令参数：

参数	描述
-d	指定不将地址解析为计算机名称。
-h maximum_hops	指定搜索目标的最大跃点数。
-j computer-list	指定沿计算机列表的松散源路由。
-w timeout	对于每个答复等待的超时毫秒数。
target_name	目标计算机的名称。

ICMP 不可达消息速率限制

在 Cisco 路由器中，ICMP 不可达限制为每 500 ms 一个数据包（作为针对“拒绝服务”(DoS) 攻击的保护措施）。在 Cisco IOS 软件版本 12.1 和更高版本中，此速率值是可配置的。引入的命令是：

```
ip icmp rate-limit unreachable  [DF] <1-4294967295 millisecond>

no ip icmp rate-limit unreachable [DF] (DF limits rate for code=4)
```

有关详细信息，请参阅 Cisco bug ID CSCdp28161（仅限注册用户）。

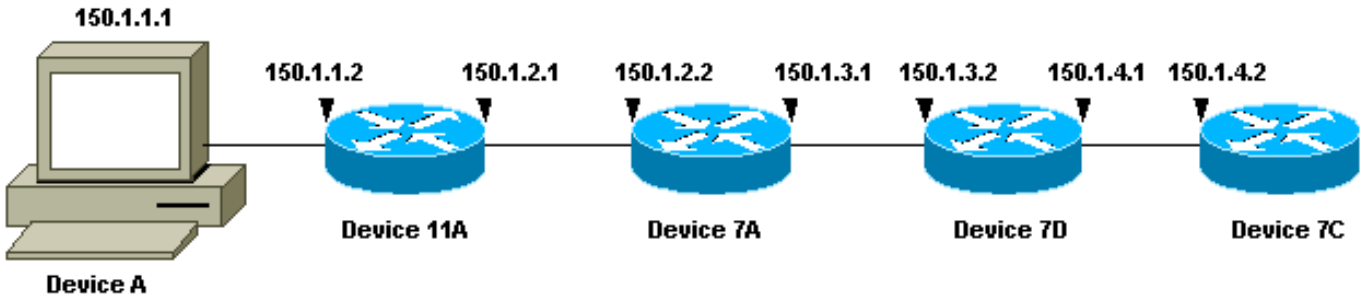
此限制是所有 ICMP 不可达的总速率，如以下输出所示。有关更多信息，请参阅 RFC 792 。

```
type = 3, code
0 = net unreachable;
1 = host unreachable;
2 = protocol unreachable;
3 = port unreachable;
4 = fragmentation needed and DF set;
5 = source route failed.
```

此限制不影响其他数据包（如 ICMP Echo 请求或 ICMP“time exceeded”消息）。

Examples

示例中使用以下网络拓扑：



在三个示例中，每个示例分别使用一个不同的 Device A。从 Device A 开始，**traceroute 150.1.4.2 命令执行到 Device 7C。**

在每个示例中，都在 Device 11A 上运行 **debug ip packet detail 命令。**

使用 Cisco IOS 软件的 Cisco 路由器

此扩展 traceroute 命令示例显示从 Cisco 路由器执行 **traceroute 命令时可以更改的选项。**在本例中，一切均采用默认设置：

```
rp-10c-2611#traceroute
Protocol [ip]:
Target IP address: 150.1.4.2
Source address: 150.1.1.1
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Type escape sequence to abort.
Tracing the route to 150.1.4.2

1 150.1.1.2 4 msec 0 msec 4 msec
2 150.1.2.2 4 msec 4 msec 0 msec
3 150.1.3.2 0 msec 0 msec 4 msec
4 150.1.4.2 4 msec * 0 msec


rp-11a-7204#
*Dec 29 13:13:57.060: IP: s=150.1.1.2 (local), d=150.1.1.1 (Ethernet4/0),
 len 56, sending
*Dec 29 13:13:57.060: ICMP type=11, code=0
*Dec 29 13:13:57.064: IP: s=150.1.1.2 (local), d=150.1.1.1 (Ethernet4/0),
 len 56, sending
*Dec 29 13:13:57.064: ICMP type=11, code=0
*Dec 29 13:13:57.064: IP: s=150.1.1.2 (local), d=150.1.1.1 (Ethernet4/0),
 len 56, sending
*Dec 29 13:13:57.068: ICMP type=11, code=0
```

在此调试输出中，Device 11A 向探报源 (150.1.1.1) 发送 ICMP“time exceeded”消息。 这些ICMP消息是对TTL=1的初始探测的响应。设备11A将TTL递减到零，并以“time exceeded”消息响应。

注意： 出于两个原因，您在此调试输出中看不到 UDP 探报：

- Device 11A 不是 UDP 探报的目的地。
- TTL 递减到零，并且数据包从未路由。所以，调试操作从未识别该数据包。

```
*Dec 29 13:13:57.068: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 28, forward
*Dec 29 13:13:57.068: UDP src=40309, dst=33437
*Dec 29 13:13:57.068: IP: s=150.1.2.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 13:13:57.068: ICMP type=11, code=0
*Dec 29 13:13:57.072: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 28, forward
*Dec 29 13:13:57.072: UDP src=37277, dst=33438
*Dec 29 13:13:57.072: IP: s=150.1.2.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 13:13:57.072: ICMP type=11, code=0
*Dec 29 13:13:57.076: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 28, forward
*Dec 29 13:13:57.076: UDP src=36884, dst=33439
*Dec 29 13:13:57.076: IP: s=150.1.2.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 13:13:57.076: ICMP type=11, code=0
```

此调试输出显示 UDP 探报源为 150.1.1.1，目的地为 150.1.4.2。

注意： 在这些探报中，TTL=2（使用调试看不到）。 Device 11A 将 TTL 递减到 1 并将 UDP 数据包转发到 Device 7A。Device 7A 将 TTL 递减到零，并以 ICMP“time exceeded”消息进行响应。

```
*Dec 29 13:13:57.080: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 28, forward
*Dec 29 13:13:57.080: UDP src=37479, dst=33440
*Dec 29 13:13:57.080: IP: s=150.1.3.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 13:13:57.080: ICMP type=11, code=0
*Dec 29 13:13:57.084: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 28, forward
*Dec 29 13:13:57.084: UDP src=40631, dst=33441
*Dec 29 13:13:57.084: IP: s=150.1.3.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 13:13:57.084: ICMP type=11, code=0
*Dec 29 13:13:57.084: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 28, forward
*Dec 29 13:13:57.088: UDP src=39881, dst=33442
*Dec 29 13:13:57.088: IP: s=150.1.3.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 13:13:57.088: ICMP type=11, code=0
```

在此调试输出中可以看到接下来的三个 UDP 探报。这些探报的 TTL 是 3。Device 11A 将 TTL 递减到 2 并将它们转发到 Device 7A。Device 7A 将 TTL 递减到 1 并将数据包转发到 Device 7B，Device 7B 将 TTL 递减到零并以 ICMP“time exceeded”消息进行响应。

```
*Dec 29 13:13:57.088: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 28, forward
*Dec 29 13:13:57.088: UDP src=39217, dst=33443
*Dec 29 13:13:57.092: IP: s=150.1.4.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 13:13:57.092: ICMP type=3, code=3
*Dec 29 13:13:57.092: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 28, forward
*Dec 29 13:13:57.096: UDP src=34357, dst=33444
*Dec 29 13:14:00.092: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 28, forward
```

```
*Dec 29 13:14:00.092: UDP src=39587, dst=33445
*Dec 29 13:14:00.092: IP: s=150.1.4.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 13:14:00.092: ICMP type=3, code=3
```

在此调试输出中可以看到最后三个 UDP 探报。这些探测器的原始TTL为4。TTL由设备11A递减到3，然后由设备7A递减到2，然后由设备7B递减到1。Device 7C 以 ICMP“port unreachable”消息进行响应，因为它是探报的目的地。

注意：由于速率限制，设备7C只发送两条ICMP“port unreachable”消息。

使用 Linux 的 PC

```
[root#linux-pc]#traceroute -n 150.1.4.2
traceroute to 150.1.4.2 (150.1.4.2), 30 hops max, 40 byte packets
1. 150.1.1.2 1.140 ms 0.793 ms 0.778 ms
2. 150.1.2.2 2.213 ms 2.105 ms 3.491 ms
1. 150.1.3.2 3.146 ms 2.314 ms 2.347 ms
1. 150.1.4.2 3.579 ms * 2.954 ms
```

```
rp-11a-7204#
*Jan 2 07:17:27.894: IP: s=150.1.1.2 (local), d=150.1.1.1 (Ethernet4/0),
len 56, sending
*Jan 2 07:17:27.894: ICMP type=11, code=0
*Jan 2 07:17:27.894: IP: s=150.1.1.2 (local), d=150.1.1.1 (Ethernet4/0),
len 56, sending
*Jan 2 07:17:27.894: ICMP type=11, code=0
*Jan 2 07:17:27.894: IP: s=150.1.1.2 (local), d=150.1.1.1 (Ethernet4/0),
len 56, sending
*Jan 2 07:17:27.894: ICMP type=11, code=0
```

在此调试输出中，Device 11A 向探报源 (150.1.1.1) 发送 ICMP“time exceeded”消息。 这些ICMP消息是对TTL=1的初始探测的响应。设备11A将TTL递减到零，并以“time exceeded”消息响应。

注意：您在此调试输出中看不到UDP探测，原因有二：

- Device 11A 不是 UDP 探报的目的地。
- TTL 递减到零，并且数据包从未路由。所以，调试操作从未识别该数据包。

```
*Jan 2 07:17:27.894: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2(FastEthernet0/0),
 g=150.1.2.2, len 40, forward
*Jan 2 07:17:27.894: UDP src=33302, dst=33438
*Jan 2 07:17:27.898: IP: s=150.1.2.2 (FastEthernet0/0), d=150.1.1.1(Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Jan 2 07:17:27.898: ICMP type=11, code=0
*Jan 2 07:17:27.898: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2(FastEthernet0/0),
 g=150.1.2.2, len 40, forward
*Jan 2 07:17:27.898: UDP src=33302, dst=33439
*Jan 2 07:17:27.898: IP: s=150.1.2.2 (FastEthernet0/0), d=150.1.1.1(Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Jan 2 07:17:27.898: ICMP type=11, code=0
*Jan 2 07:17:27.898: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2(FastEthernet0/0),
 g=150.1.2.2, len 40, forward
*Jan 2 07:17:27.898: UDP src=33302, dst=33440
*Jan 2 07:17:27.902: IP: s=150.1.2.2 (FastEthernet0/0), d=150.1.1.1(Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Jan 2 07:17:27.902: ICMP type=11, code=0
```

注意：在此调试输出中，您现在看到从源150.1.1.1发往150.1.4.2的UDP探测。

注意：在这些探测中，TTL=2（这在调试中无法看到）。 Device 11A 将 TTL 递减到 1 并将 UDP 数据包转发到 Device 7A。Device 7A 将 TTL 递减到零，并以 ICMP“time exceeded”消息进行响应。

```
*Jan 2 07:17:27.902: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2(FastEthernet0/0),
 g=150.1.2.2, len 40, forward
*Jan 2 07:17:27.902: UDP src=33302, dst=33441
*Jan 2 07:17:27.906: IP: s=150.1.3.2 (FastEthernet0/0), d=150.1.1.1(Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Jan 2 07:17:27.906: ICMP type=11, code=0
*Jan 2 07:17:27.906: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2(FastEthernet0/0),
 g=150.1.2.2, len 40, forward
*Jan 2 07:17:27.906: UDP src=33302, dst=33442
*Jan 2 07:17:27.910: IP: s=150.1.3.2 (FastEthernet0/0), d=150.1.1.1(Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Jan 2 07:17:27.910: ICMP type=11, code=0
*Jan 2 07:17:27.910: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2(FastEthernet0/0),
 g=150.1.2.2, len 40, forward
*Jan 2 07:17:27.910: UDP src=33302, dst=33443
*Jan 2 07:17:27.910: IP: s=150.1.3.2 (FastEthernet0/0), d=150.1.1.1(Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Jan 2 07:17:27.910: ICMP type=11, code=0
```

现在，在此调试输出中可以看到接下来的三个 UDP 探报。这些探报的 TTL 是 3。Device 11A 将 TTL 递减到 2 并将它们转发到 Device 7A。Device 7A 将 TTL 递减到 1 并将数据包转发到 Device 7B，Device 7B 将 TTL 递减到零并以 ICMP“time exceeded”消息进行响应。

```
*Jan 2 07:17:27.910: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2(FastEthernet0/0),
 g=150.1.2.2, len 40, forward
*Jan 2 07:17:27.910: UDP src=33302, dst=33444
*Jan 2 07:17:27.914: IP: s=150.1.4.2 (FastEthernet0/0), d=150.1.1.1(Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Jan 2 07:17:27.914: ICMP type=3, code=3
*Jan 2 07:17:27.914: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2(FastEthernet0/0),
 g=150.1.2.2, len 40, forward
*Jan 2 07:17:27.914: UDP src=33302, dst=33445
*Jan 2 07:17:32.910: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2(FastEthernet0/0),
 g=150.1.2.2, len 40, forward
*Jan 2 07:17:32.910: UDP src=33302, dst=33446
*Jan 2 07:17:32.914: IP: s=150.1.4.2 (FastEthernet0/0), d=150.1.1.1(Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Jan 2 07:17:32.914: ICMP type=3, code=3
```

此调试输出显示最后三个 UDP 探报。这些探测器的原始TTL为4。TTL由设备11A递减到3，然后由设备7A递减到2，然后由设备7B递减到1。然后，Device 7C 以 ICMP“port unreachable”消息进行响应，因为它是探报的目的地。

注意：由于速率限制，设备7C只发送两条ICMP“port unreachable”消息。

使用 MS Windows 的 PC

```
C:\>tracert 150.1.4.2

1 <10 ms <10 ms <10 ms 10.1.1.2
1 <10 ms <10 ms <10 ms 10.1.2.2
1 <10 ms <10 ms <10 ms 10.1.3.2
```

```
1 <10 ms 10 ms 10 ms 10.1.4.2

Trace complete

rp-11a-7204#
*Dec 29 14:02:22.236: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 78, forward
*Dec 29 14:02:22.236: UDP src=137, dst=137
*Dec 29 14:02:22.240: IP: s=150.1.4.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 14:02:22.240: ICMP type=3, code=3
*Dec 29 14:02:23.732: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 78, forward
*Dec 29 14:02:23.732: UDP src=137, dst=137
*Dec 29 14:02:23.736: IP: s=150.1.4.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 14:02:23.736: ICMP type=3, code=3
*Dec 29 14:02:25.236: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 78, forward
*Dec 29 14:02:25.236: UDP src=137, dst=137
*Dec 29 14:02:25.236: IP: s=150.1.4.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 14:02:25.240: ICMP type=3, code=3
*Dec 29 14:02:26.748: IP: s=150.1.1.2 (local), d=150.1.1.1 (Ethernet4/0),
 len 56, sending
*Dec 29 14:02:26.748: ICMP type=11, code=0
*Dec 29 14:02:26.752: IP: s=150.1.1.2 (local), d=150.1.1.1 (Ethernet4/0),
 len 56, sending
*Dec 29 14:02:26.752: ICMP type=11, code=0
*Dec 29 14:02:26.752: IP: s=150.1.1.2 (local), d=150.1.1.1 (Ethernet4/0),
 len 56, sending
*Dec 29 14:02:26.752: ICMP type=11, code=0
```

在此调试输出中，Device 11A 向探报源 (150.1.1.1) 发送 ICMP“time exceeded”消息。 这些ICMP消息是对初始探测的响应，初始探测是TTL=1的ICMP回应请求数据包。设备11A将TTL递减到零，并以ICMP消息响应。

注意：在顶部您会看到NETBIOS名称请求。这些请求被视为源端口和目的端口为137的UDP数据包。出于明确原因，NETBIOS数据包会从调试输出的其余部分删除。可以在 **tracert 命令中使用** *-d 选项禁用 NETBIOS 行为。*

注意：您在此调试输出中看不到ICMP探**测**，原因有二：

- Device 11A 不是 ICMP 探报的目的地。
- TTL 递减到零，并且数据包从未路由。所以，调试操作从未识别该数据包。

```
*Dec 29 14:02:32.256: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 92, forward
*Dec 29 14:02:32.256: ICMP type=8, code=0
*Dec 29 14:02:32.260: IP: s=150.1.2.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 14:02:32.260: ICMP type=11, code=0
*Dec 29 14:02:32.260: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 92, forward
*Dec 29 14:02:32.260: ICMP type=8, code=0
*Dec 29 14:02:32.260: IP: s=150.1.2.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 14:02:32.260: ICMP type=11, code=0
*Dec 29 14:02:32.264: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 92, forward
*Dec 29 14:02:32.264: ICMP type=8, code=0
*Dec 29 14:02:32.264: IP: s=150.1.2.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
```

```
     g=150.1.1.1, len 56, forward
    *Dec 29 14:02:32.264: ICMP type=11, code=0
```

在此调试输出中，现在可以看到 ICMP 探报源为 150.1.1.1，目的地为 150.1.4.2。

注意：在这些探测中，TTL=2（这在调试中无法看到）。 设备11A将TTL递减到1，并将UDP数据包转发到设备7A。Device 7A 将 TTL 递减到零，并以 ICMP“time exceeded”消息进行响应。

```
*Dec 29 14:02:37.776: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 92, forward
*Dec 29 14:02:37.776: ICMP type=8, code=0
*Dec 29 14:02:37.776: IP: s=150.1.3.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 14:02:37.776: ICMP type=11, code=0
*Dec 29 14:02:37.780: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 92, forward
*Dec 29 14:02:37.780: ICMP type=8, code=0
*Dec 29 14:02:37.780: IP: s=150.1.3.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 14:02:37.780: ICMP type=11, code=0
*Dec 29 14:02:37.780: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 92, forward
*Dec 29 14:02:37.780: ICMP type=8, code=0
*Dec 29 14:02:37.784: IP: s=150.1.3.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 56, forward
*Dec 29 14:02:37.784: ICMP type=11, code=0
```

在此调试输出中可以看到接下来的三个 ICMP 探报。这些探报的 TTL 是 3。Device 11A 将 TTL 递减到 2 并将它们转发到 Device 7A。Device 7A 将 TTL 递减到 1 并将数据包转发到 Device 7B，Device 7B 将 TTL 递减到零并以 ICMP“time exceeded”消息进行响应。

```
*Dec 29 14:02:43.292: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 92, forward
*Dec 29 14:02:43.292: ICMP type=8, code=0
*Dec 29 14:02:43.296: IP: s=150.1.4.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 92, forward
*Dec 29 14:02:43.296: ICMP type=0, code=0
*Dec 29 14:02:43.296: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 92, forward
*Dec 29 14:02:43.296: ICMP type=8, code=0
*Dec 29 14:02:43.300: IP: s=150.1.4.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 92, forward
*Dec 29 14:02:43.300: ICMP type=0, code=0
*Dec 29 14:02:43.300: IP: s=150.1.1.1 (Ethernet4/0), d=150.1.4.2 (FastEthernet0/0),
 g=150.1.2.2, len 92, forward
*Dec 29 14:02:43.300: ICMP type=8, code=0
*Dec 29 14:02:43.304: IP: s=150.1.4.2 (FastEthernet0/0), d=150.1.1.1 (Ethernet4/0),
 g=150.1.1.1, len 92, forward
*Dec 29 14:02:43.304: ICMP type=0, code=0
```

此调试输出显示最后三个 ICMP 探报。这些探测器的原始TTL为4。TTL由设备11A递减到3，然后由设备7A递减到2，然后由设备7B递减到1。然后，Device 7C 以 ICMP Echo 回复消息 (type=0, code=0) 进行响应，因为它是探报的目的地。

注意：与ICMP“端口不可达”消息相比，ICMP回应应答消息的速率没有限制。在本例中，可以看到全部三条 ICMP Echo 回复消息均已发送。

其他说明

在 Cisco 路由器中，**traceroute 命令回复的代码如下：**

```
! -- success
* -- time out
N -- network unreachable
H -- host unreachable
P -- protocol unreachable
A -- admin denied
Q -- source quench received (congestion)
? -- unknown (any other ICMP message)
```

如果从 UNIX 运行 **traceroute 命令，请注意以下几点：**

- 您可以接收“traceroute:icmp socket:Permission denied”消息。
- **traceroute 程序依赖于网络接口开关 (NIT) 探听网络。**此设备只能由根访问。必须将该程序作为根运行，或者设置根的用户 ID。

摘要

本文档演示了如何使用 UDP 和 ICMP 数据包，通过 **traceroute 命令确定数据包从给定源到给定目的地的路径。**输出中可能存在以下 ICMP 消息类型：

- 如果 TTL 在中转中超时，type=11、code=0，那么，只要探报数据包的 TTL 在数据包到达目的地之前到期，中转路由器就会退回数据包。
- 如果端口不可达，type=3、code=3，那么，当 UDP 探报数据包到达目的地时（UDP 应用程序未定义），会在对 UDP 探报数据包的响应中退还数据包。 这些数据包限制为每 500 ms 一个数据包。这解释了在平稳的响应中来自目的地的响应（请参阅 Cisco 路由器和 Linux 的输出）仍然失败的原因。Device 7C 不生成 ICMP 消息，并且每个设备的 **traceroute 命令输出都等待超过一秒钟。**使用 MS Windows **tracert 命令输出时，因为 Cisco 路由器中不存在 UDP 端口 137，所以生成 ICMP 消息。**
- 如果有 Echo，type=8，code=0，则 Echo 探报数据包由 MS Windows PC 发送。
- 如果有 Echo 回复，type=0，code=0，则在到达目的地时会发送对上一数据包的回复。这只适用于 MS Windows **tracert 命令。**

相关信息

- 技术支持和文档 - Cisco Systems