

Use the Traceroute Command on Operating Systems

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Conventions](#)

[Background Information](#)

[General Operation](#)

[Cisco IOS and Linux](#)

[Microsoft Windows](#)

[ICMP Unreachables Rate Limitation](#)

[Examples](#)

[Cisco Router with Cisco IOS Software](#)

[PC with Linux](#)

[PC Running MS Windows](#)

[Additional Notes](#)

[Summary](#)

[Related Information](#)

Introduction

This document describes how the `traceroute` command operates on different systems.

Prerequisites

Requirements

Readers of this document must have basic knowledge of one of these operating systems:

- Cisco IOS® Software
- Linux
- Microsoft Windows

Components Used

The information in this document applies to these software and hardware versions:

- Cisco Router that runs Cisco IOS Software Release 12

- PC that runs Red Hat Linux
- PC that runs MS Windows

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Conventions

For more information on document conventions, refer to the [Cisco Technical Tips Conventions](#).

Background Information

The `tracert` command allows you to determine the path a packet takes in order to get to a destination from a given source by returning the sequence of hops the packet has traversed. This utility comes with your host operating system (for example, Linux or Microsoft (MS) Windows), as well as with Cisco IOS Software.

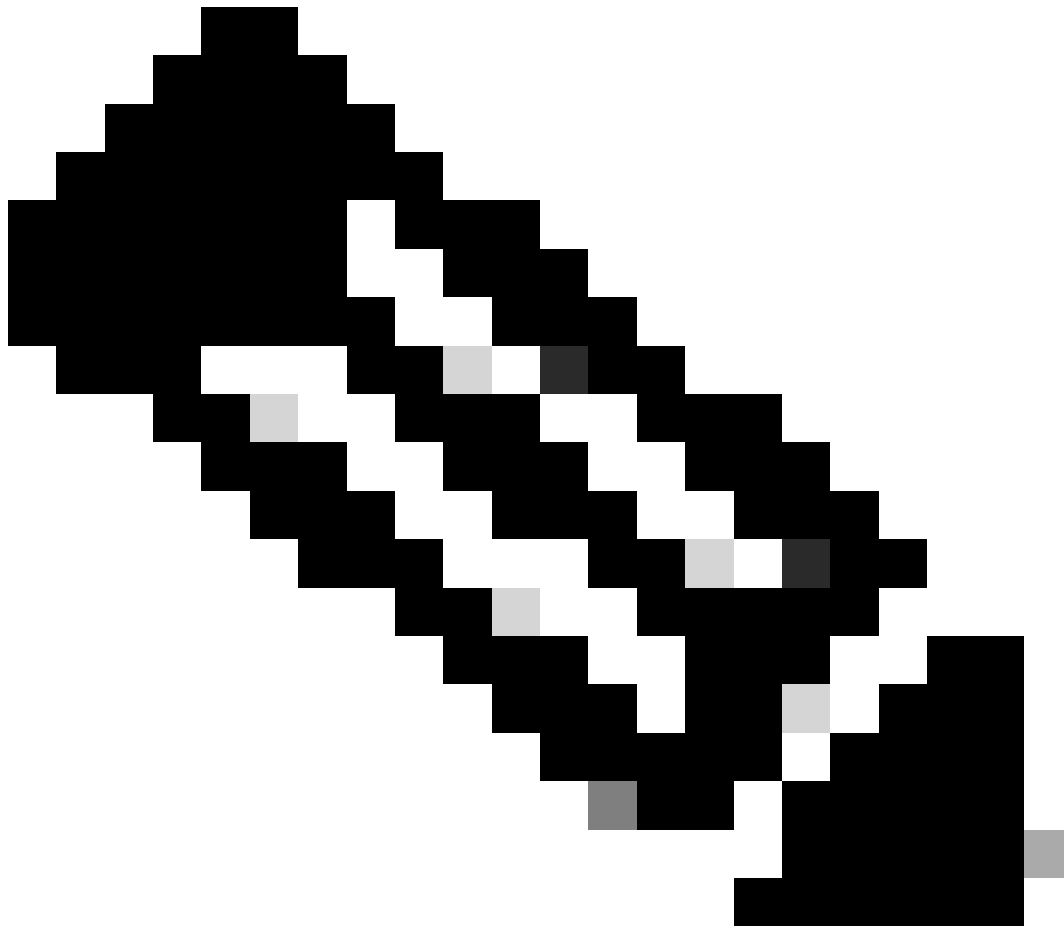
General Operation

If you execute the `tracert ip-address` command on a source device (such as a host, or a router acting as a host), it sends IP packets toward the destination with Time To Live (TTL) values that increment up to the maximum specified hop count. This is 30 by default. Typically, each router in the path towards the destination decrements the TTL field by one unit while it forwards these packets. When a router in the middle of the path finds a packet with TTL = 1, it responds with an Internet Control Message Protocol (ICMP) "time exceeded" message to the source. This message lets the source know that the packet traverses that particular router as a hop

There are some differences with the way the `tracert` command is implemented in the various operating systems this document discusses.

Cisco IOS and Linux

The TTL for the initial User Datagram Protocol (UDP) datagram probe is set to 1 (or the minimum TTL, as specified by user in the extended `tracert` command). The destination UDP port of the initial datagram probe is set to 33434 (or as specified in the extended `tracert` command output). The extended `tracert` command is a variation of the ordinary `tracert` command which allows the default values of the parameters used by the `tracert` operation such as TTL and destination port number to be modified. For more information on how to use the extended `tracert` command, refer to [Understand the Extended ping and Extended tracert Commands](#). The source UDP port of the initial datagram probe is randomized and has logical operator OR with 0x8000 (ensures a minimum source port of 0x8000). These steps illustrate what happens when the UDP datagram is launched:



Note: The parameters are configurable. This example starts with $n = 1$ and finishes with $n = 3$.

-
1. The UDP datagram is dispatched with $TTL = 1$, destination UDP port= 33434, and the source port randomized.
 2. The UDP destination port is incremented, the source UDP port is randomized, and the second datagram dispatched.
 3. Step 2 is repeated for up to three probes (or as many times as requested in an extended `traceroute` command output). For each of the probes sent, you receive a "TTL exceeded" message, which is used to build a step-by-step path to the destination host.
 4. TTL is incremented, and this cycle repeats with incremental destination port numbers, if the ICMP time exceeded message is received. You can also get one of these messages:
 - An ICMP type 3, code 3 (destination unreachable, port unreachable) message, which indicates that a host has been reached.
 - A host unreachable, net unreachable, maximum TTL exceeded, or a timeout type of message, which means that the probe is resent.

Cisco routers send UDP probe packets with a random source port and an incremental destination port (to distinguish the different probes). Cisco routers send the ICMP message time exceeded back to the source from where the UDP/ICMP packet was received.

The Linux `traceroute` command is similar to the Cisco router implementation. However, it uses a fixed source port. The `-n` option in the `traceroute` command is used to avoid a request to a name server.

Microsoft Windows

The MS Windows `tracert` command uses ICMP echo request datagrams instead of UDP datagrams as probes. ICMP echo requests are launched with incrementing TTL, and the same operation as described in [Cisco IOS and Linux](#) occurs. The significance of using ICMP echo request datagrams is that the final hop does not rely on the response of an ICMP unreachable message from the destination host. It relies instead on an ICMP echo reply message.

The command syntax is:

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

This table explains the command parameters:

Parameter	Description
-d	Specifies not to resolve addresses to computer names.
-h maximum_hops	Specifies the maximum number of hops to search for a target.
-j computer-list	Specifies a loose source route along computer-list.
-w timeout	Waits the number of milliseconds specified by the timeout for each reply.
target_name	Name of the target computer.

ICMP Unreachables Rate Limitation

ICMP unreachables are limited to one packet per 500 ms (as a protection for Denial of Service (DoS) attacks) in a Cisco Router. From Cisco IOS Software Release 12.1 and later, this rate value is configurable. The command introduced is:

```
<#root>
Router(config)#
ip icmp rate-limit unreachable ?
<1-4294967295> Once per milliseconds
DF code 4, fragmentation needed and DF set
```

This limitation is for the aggregate rate of all the ICMP unreachables, as this output shows. Refer to [RFC 792](#) for more information.

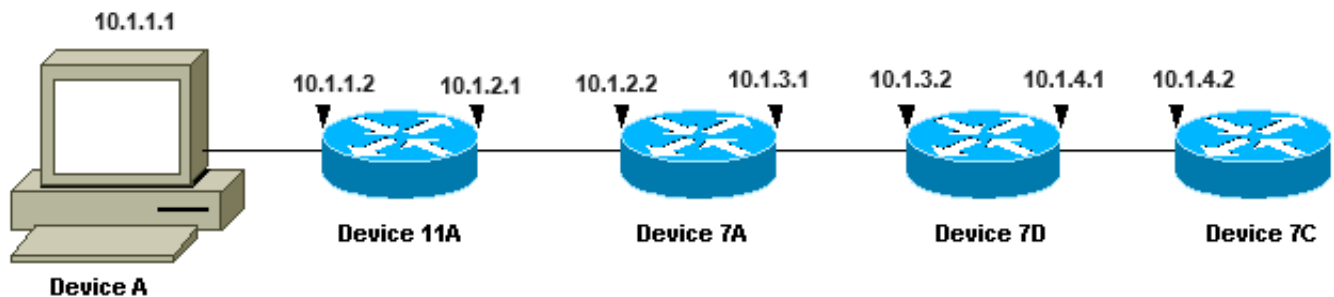
```
type = 3, code
```

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

This limitation does not affect other packets like ICMP echo requests or ICMP time exceeded messages.

Examples

This network topology is used for the examples:



Network Topology

In each of the three examples, a different Device A is used. From Device A, the `traceroute 10.1.4.2` command is executed to Device 7C.

In each of the examples, the `debug ip packet detail` command runs on Device 11A.

Cisco Router with Cisco IOS Software

This extended `traceroute` command example shows the options you can change when you execute a `traceroute` command from a Cisco router. In this example, everything is left default:

```
<#root>

rp-10c-2611#
traceroute

Protocol [ip]:
Target IP address: 10.1.4.2
Source address: 10.1.1.1
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 10.1.4.2

 1 10.1.1.2 4 msec 0 msec 4 msec
```

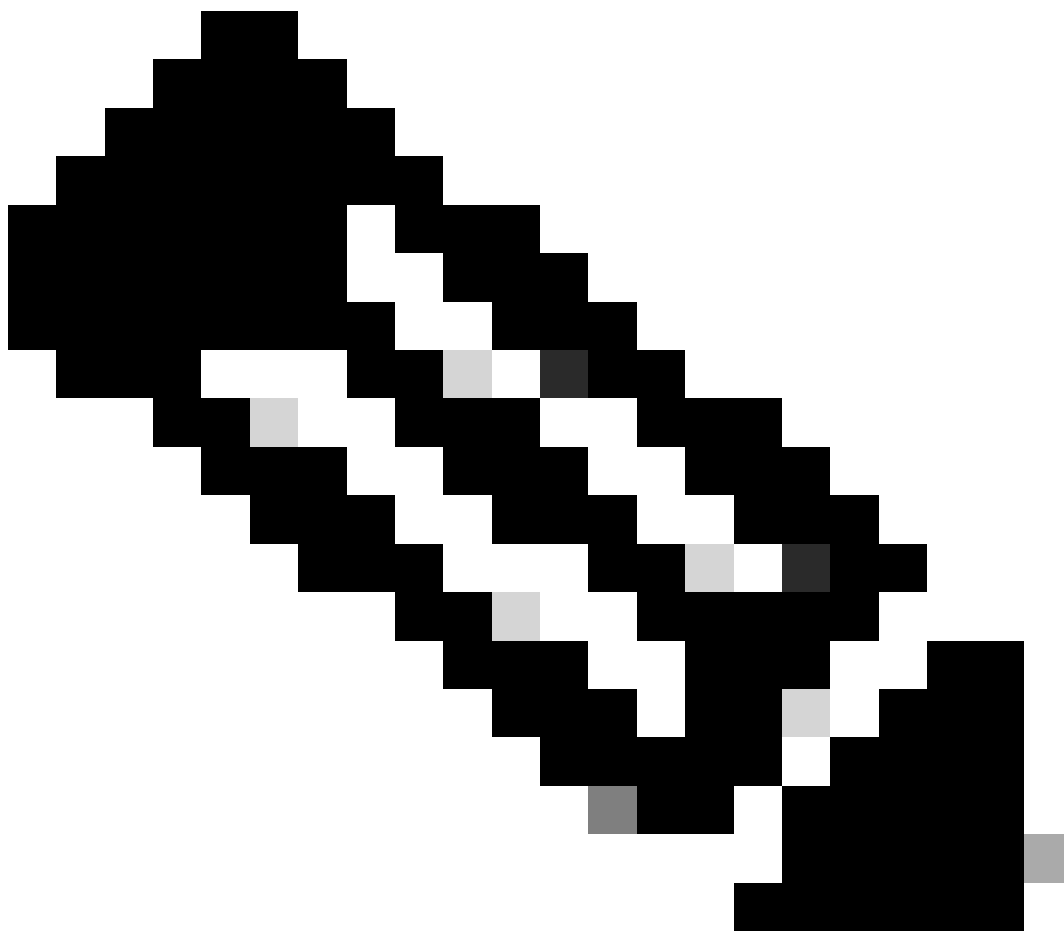
```
2 10.1.2.2 4 msec 4 msec 0 msec
3 10.1.3.2 0 msec 0 msec 4 msec
4 10.1.4.2 4 msec * 0 msec
```

rp-11a-7204#

```
*Dec 29 13:13:57.060: IP: s=10.1.1.2 (local), d=10.1.1.1 (Ethernet4/0),
  len 56, sending
*Dec 29 13:13:57.060: ICMP type=11, code=0

*Dec 29 13:13:57.064: IP: s=10.1.1.2 (local), d=10.1.1.1 (Ethernet4/0),
  len 56, sending
*Dec 29 13:13:57.064: ICMP type=11, code=0
*Dec 29 13:13:57.064: IP: s=10.1.1.2 (local), d=10.1.1.1 (Ethernet4/0),
  len 56, sending
*Dec 29 13:13:57.068: ICMP type=11, code=0
```

In this debug output, Device 11A sends ICMP time exceeded messages to the source of the probes (10.1.1.1). These ICMP messages are in response to the initial probes that had a TTL=1. Device 11A decrements the TTL to zero, and responds with the time exceeded messages.



Note: You do not see the UDP probes in this debug output for two reasons:

- Device 11A is not the destination of the UDP probes.
- The TTL is decremented to zero, and the packet is never routed. Therefore, the debug never recognizes the packet.

<#root>

```
*Dec 29 13:13:57.068: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 28, forward
*Dec 29 13:13:57.068: UDP src=40309, dst=33437
*Dec 29 13:13:57.068: IP: s=10.1.2.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Dec 29 13:13:57.068: ICMP type=11, code=0

*Dec 29 13:13:57.072: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 28, forward
*Dec 29 13:13:57.072: UDP src=37277, dst=33438
*Dec 29 13:13:57.072: IP: s=10.1.2.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Dec 29 13:13:57.072: ICMP type=11, code=0
*Dec 29 13:13:57.076: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 28, forward
*Dec 29 13:13:57.076: UDP src=36884, dst=33439
*Dec 29 13:13:57.076: IP: s=10.1.2.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Dec 29 13:13:57.076: ICMP type=11, code=0
```

This debug output shows the UDP probe from source 10.1.1.1 destined to 10.1.4.2.

Note: In these probes the TTL=2 (this cannot be seen with debug). Device 11A decrements the TTL to 1 and forwards the UDP packets onto Device 7A. Device 7A decrements the TTL to zero, and responds with ICMP time exceeded messages.

<#root>

```
*Dec 29 13:13:57.080: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 28, forward
*Dec 29 13:13:57.080: UDP src=37479, dst=33440
*Dec 29 13:13:57.080: IP: s=10.1.3.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Dec 29 13:13:57.080: ICMP type=11, code=0

*Dec 29 13:13:57.084: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 28, forward
*Dec 29 13:13:57.084: UDP src=40631, dst=33441
*Dec 29 13:13:57.084: IP: s=10.1.3.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Dec 29 13:13:57.084: ICMP type=11, code=0
*Dec 29 13:13:57.084: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 28, forward
```



```
*Dec 29 13:13:57.088: UDP src=39881, dst=33442
*Dec 29 13:13:57.088: IP: s=10.1.3.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Dec 29 13:13:57.088: ICMP type=11, code=0
```

You see the next three UDP probes in this debug output. The TTL for these probes is 3. Device 11A decrements the TTL to 2 and forwards them on to Device 7A. Device 7A decrements the TTL to 1 and forwards the packets on to Device 7B, which decrements the TTL to zero and responds with ICMP time exceeded messages.

```
<#root>
```

```
*Dec 29 13:13:57.088: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 28, forward
*Dec 29 13:13:57.088: UDP src=39217, dst=33443
*Dec 29 13:13:57.092: IP: s=10.1.4.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Dec 29 13:13:57.092: ICMP type=3, code=3

*Dec 29 13:13:57.092: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 28, forward
*Dec 29 13:13:57.096: UDP src=34357, dst=33444
*Dec 29 13:14:00.092: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 28, forward
*Dec 29 13:14:00.092: UDP src=39587, dst=33445
*Dec 29 13:14:00.092: IP: s=10.1.4.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Dec 29 13:14:00.092: ICMP type=3, code=3
```

You can see the last three UDP probes in this debug output. The original TTL of these probes was 4. The TTL was decremented to 3 by Device 11A, then decremented to 2 by Device 7A, then decremented to 1 by Device 7B. Device 7C responds with ICMP port unreachable messages, since it was the destination of the probes.

Note: Device 7C only sends two ICMP port unreachable messages because of the rate limitation.

PC with Linux

```
<#root>
```

```
[root#linux-pc]#
```

```
traceroute -n 10.1.4.2
```

```
traceroute to 10.1.4.2 (10.1.4.2), 30 hops max, 40 byte packets
```

```
1. 10.1.1.2 1.140 ms 0.793 ms 0.778 ms
2. 10.1.2.2 2.213 ms 2.105 ms 3.491 ms
1. 10.1.3.2 3.146 ms 2.314 ms 2.347 ms
1. 10.1.4.2 3.579 ms * 2.954 ms
```

```
rp-11a-7204#
```

```
*Jan 2 07:17:27.894: IP: s=10.1.1.2 (local), d=10.1.1.1 (Ethernet4/0),
len 56, sending
```

```
*Jan 2 07:17:27.894: ICMP type=11, code=0

*Jan 2 07:17:27.894: IP: s=10.1.1.2 (local), d=10.1.1.1 (Ethernet4/0),
len 56, sending
*Jan 2 07:17:27.894: ICMP type=11, code=0
*Jan 2 07:17:27.894: IP: s=10.1.1.2 (local), d=10.1.1.1 (Ethernet4/0),
len 56, sending
*Jan 2 07:17:27.894: ICMP type=11, code=0
```

In this debug output, Device 11A sends ICMP time exceeded messages to the source of the probes (10.1.1.1). These ICMP messages are in response to the initial probes that had a TTL=1. Device 11A decrements the TTL to zero, and responds with the time exceeded messages.

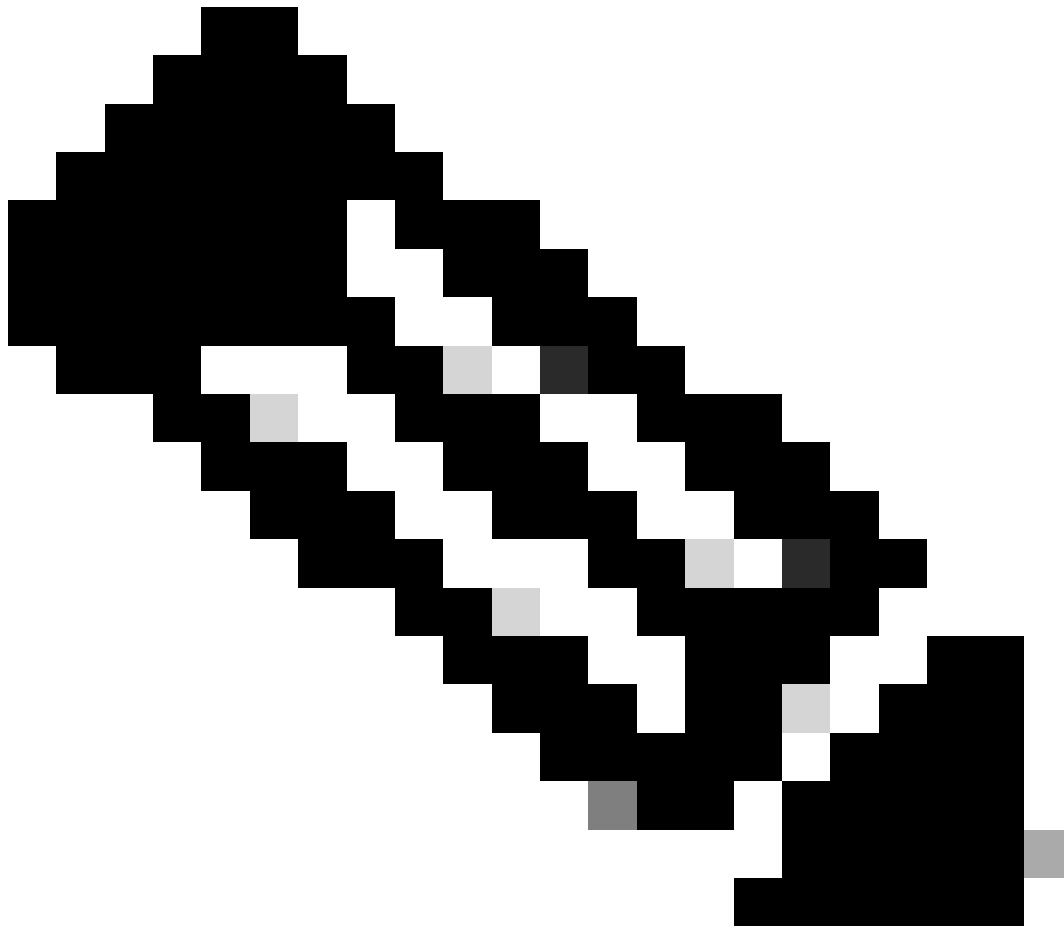
You do not see the UDP probes in this debug output for two reasons:

1. Device 11A is not the destination of the UDP probes.
2. The TTL is decremented to zero, and the packet is never routed. Therefore, the debug never recognizes the packet.

<#root>

```
*Jan 2 07:17:27.894: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2(FastEthernet0/0),
g=10.1.2.2, len 40, forward
*Jan 2 07:17:27.894: UDP src=33302, dst=33438
*Jan 2 07:17:27.898: IP: s=10.1.2.2 (FastEthernet0/0), d=10.1.1.1(Ethernet4/0),
g=10.1.1.1, len 56, forward
*Jan 2 07:17:27.898: ICMP type=11, code=0

*Jan 2 07:17:27.898: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2(FastEthernet0/0),
g=10.1.2.2, len 40, forward
*Jan 2 07:17:27.898: UDP src=33302, dst=33439
*Jan 2 07:17:27.898: IP: s=10.1.2.2 (FastEthernet0/0), d=10.1.1.1(Ethernet4/0),
g=10.1.1.1, len 56, forward
*Jan 2 07:17:27.898: ICMP type=11, code=0
*Jan 2 07:17:27.898: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2(FastEthernet0/0),
g=10.1.2.2, len 40, forward
*Jan 2 07:17:27.898: UDP src=33302, dst=33440
*Jan 2 07:17:27.902: IP: s=10.1.2.2 (FastEthernet0/0), d=10.1.1.1(Ethernet4/0),
g=10.1.1.1, len 56, forward
*Jan 2 07:17:27.902: ICMP type=11, code=0
```



Note: In this debug output, you now see the UDP probe from source 10.1.1.1 destined to 10.1.4.2.

Note: In these probes the TTL=2 (this cannot be seen with debug). Device 11A decrements the TTL to 1 and forwards the UDP packets onto Device 7A. Device 7A decrements the TTL to zero, and responds with ICMP time exceeded messages.

<#root>

```
*Jan 2 07:17:27.902: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2(FastEthernet0/0),
g=10.1.2.2, len 40, forward
*Jan 2 07:17:27.902: UDP src=33302, dst=33441
*Jan 2 07:17:27.906: IP: s=10.1.3.2 (FastEthernet0/0), d=10.1.1.1(Ethernet4/0),
g=10.1.1.1, len 56, forward
*Jan 2 07:17:27.906: ICMP type=11, code=0

*Jan 2 07:17:27.906: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2(FastEthernet0/0),
g=10.1.2.2, len 40, forward
*Jan 2 07:17:27.906: UDP src=33302, dst=33442
*Jan 2 07:17:27.910: IP: s=10.1.3.2 (FastEthernet0/0), d=10.1.1.1(Ethernet4/0),
g=10.1.1.1, len 56, forward
*Jan 2 07:17:27.910: ICMP type=11, code=0
*Jan 2 07:17:27.910: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2(FastEthernet0/0),
g=10.1.2.2, len 40, forward
```

```
*Jan 2 07:17:27.910: UDP src=33302, dst=33443
*Jan 2 07:17:27.910: IP: s=10.1.3.2 (FastEthernet0/0), d=10.1.1.1(Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Jan 2 07:17:27.910: ICMP type=11, code=0
```

The next three UDP probes are now seen in this debug output. The TTL for these probes is 3. Device 11A decrements the TTL to 2 and forwards them on to Device 7A. Device 7A decrements the TTL to 1 and forwards the packets on to Device 7B, which decrements the TTL to zero and responds with ICMP time exceeded messages.

```
<#root>
```

```
*Jan 2 07:17:27.910: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2(FastEthernet0/0),
  g=10.1.2.2, len 40, forward
*Jan 2 07:17:27.910: UDP src=33302, dst=33444
*Jan 2 07:17:27.914: IP: s=10.1.4.2 (FastEthernet0/0), d=10.1.1.1(Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Jan 2 07:17:27.914: ICMP type=3, code=3

*Jan 2 07:17:27.914: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2(FastEthernet0/0),
  g=10.1.2.2, len 40, forward
*Jan 2 07:17:27.914: UDP src=33302, dst=33445
*Jan 2 07:17:32.910: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2(FastEthernet0/0),
  g=10.1.2.2, len 40, forward
*Jan 2 07:17:32.910: UDP src=33302, dst=33446
*Jan 2 07:17:32.914: IP: s=10.1.4.2 (FastEthernet0/0), d=10.1.1.1(Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Jan 2 07:17:32.914: ICMP type=3, code=3
```

This debug output shows the last three UDP probes. The original TTL of these probes was 4. The TTL was decremented to 3 by Device 11A, then decremented to 2 by Device 7A, then decremented to 1 by Device 7B.

Device 7C then responds with ICMP port unreachable messages, since it was the destination of the probes.

Note: Device 7C only sends two ICMP port unreachable messages because of the rate limiting.

PC Running MS Windows

```
<#root>
```

```
C:\>
```

```
tracert 10.1.4.2
```

```
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=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
```

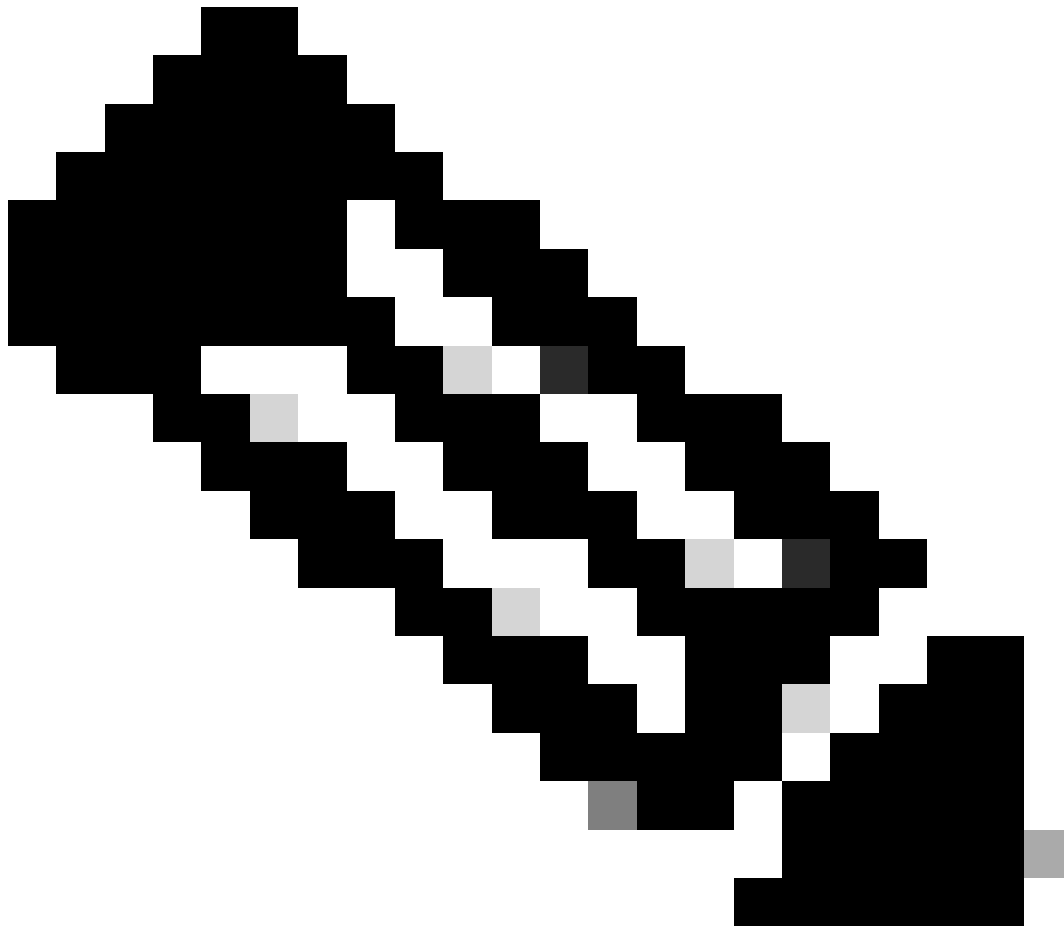
```
g=10.1.2.2, len 78, forward
*Dec 29 14:02:22.236: UDP src=137, dst=137
*Dec 29 14:02:22.240: IP: s=10.1.4.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
g=10.1.1.1, len 56, forward
*Dec 29 14:02:22.240: ICMP type=3, code=3

*Dec 29 14:02:23.732: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
g=10.1.2.2, len 78, forward
*Dec 29 14:02:23.732: UDP src=137, dst=137
*Dec 29 14:02:23.736: IP: s=10.1.4.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
g=10.1.1.1, len 56, forward
*Dec 29 14:02:23.736: ICMP type=3, code=3
*Dec 29 14:02:25.236: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
g=10.1.2.2, len 78, forward
*Dec 29 14:02:25.236: UDP src=137, dst=137
*Dec 29 14:02:25.236: IP: s=10.1.4.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
g=10.1.1.1, len 56, forward
*Dec 29 14:02:25.240: ICMP type=3, code=3

*Dec 29 14:02:26.748: IP: s=10.1.1.2 (local), d=10.1.1.1 (Ethernet4/0),
len 56, sending
*Dec 29 14:02:26.748: ICMP type=11, code=0

*Dec 29 14:02:26.752: IP: s=10.1.1.2 (local), d=10.1.1.1 (Ethernet4/0),
len 56, sending
*Dec 29 14:02:26.752: ICMP type=11, code=0
*Dec 29 14:02:26.752: IP: s=10.1.1.2 (local), d=10.1.1.1 (Ethernet4/0),
len 56, sending
*Dec 29 14:02:26.752: ICMP type=11, code=0
```

In this debug output, Device 11A sends ICMP time exceeded messages to the source of the probes (10.1.1.1). These ICMP messages are in response to the initial probes, which are ICMP echo request packets with a TTL=1. Device 11A decrements the TTL to zero and responds with the ICMP messages.



Note: At the top you see the NETBIOS name requests. These requests are seen as UDP packets with source and destination ports of 137. For clarity reasons, the NETBIOS packets are removed from the rest of the debug output. You can use the `-d` option in the `tracert` command to disable the NETBIOS behavior.

You do not see the ICMP probes in this debug output for two reasons:

1. Device 11A is not the destination of the ICMP probes.
2. The TTL is decremented to zero, and the packet is never routed. Therefore, the debug never recognizes the packet.

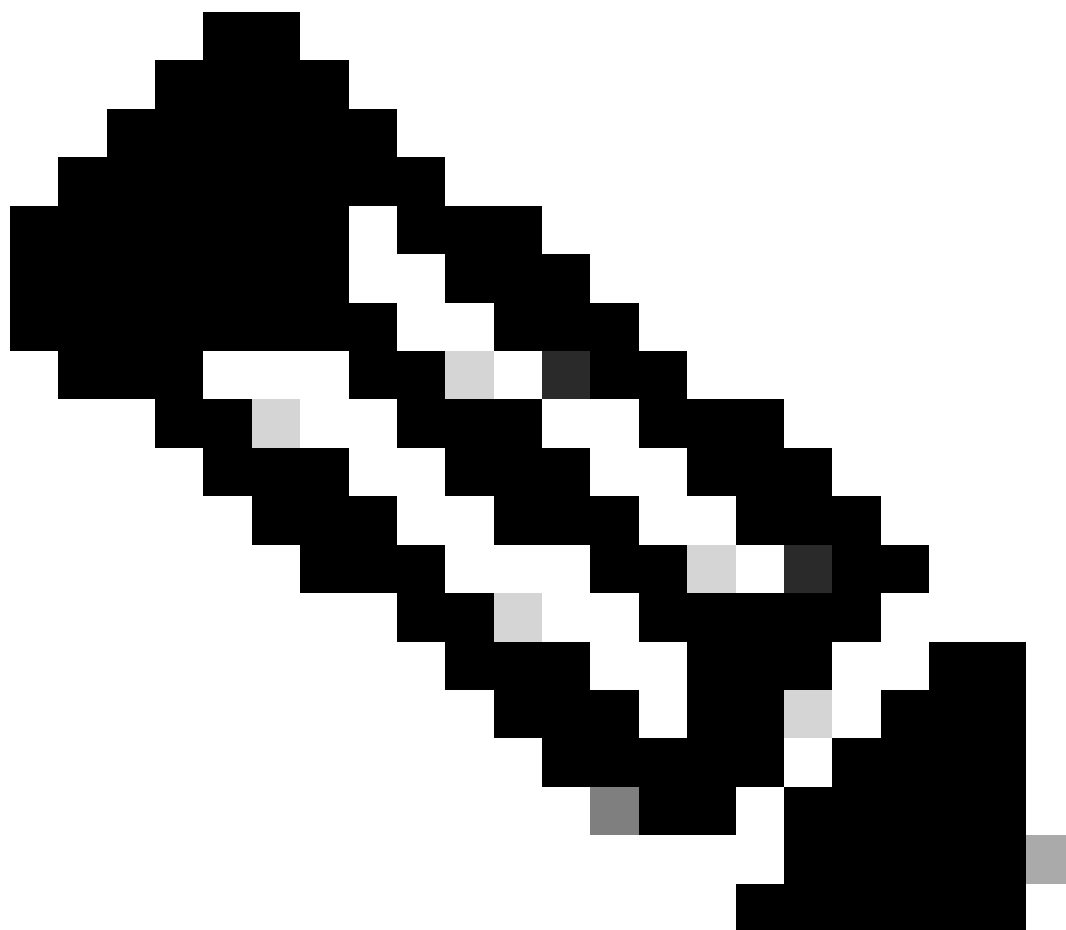
<#root>

```
*Dec 29 14:02:32.256: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 92, forward
*Dec 29 14:02:32.256: ICMP type=8, code=0
*Dec 29 14:02:32.260: IP: s=10.1.2.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 56, forward
```

```
*Dec 29 14:02:32.260: ICMP type=11, code=0

*Dec 29 14:02:32.260: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 92, forward
*Dec 29 14:02:32.260: ICMP type=8, code=0
*Dec 29 14:02:32.260: IP: s=10.1.2.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Dec 29 14:02:32.260: ICMP type=11, code=0
*Dec 29 14:02:32.264: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 92, forward
*Dec 29 14:02:32.264: ICMP type=8, code=0
*Dec 29 14:02:32.264: IP: s=10.1.2.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Dec 29 14:02:32.264: ICMP type=11, code=0
```

In this debug output, you now see the ICMP probe from source 10.1.1.1 destined to 10.1.4.2.



Note: In these probes, the TTL=2 (this cannot be seen with debug). Device 11A decrements the TTL to 1 and forwards the UDP packets on to Device 7A. Device 7A decrements the TTL to zero, and responds with ICMP time exceeded messages.

<#root>

```
*Dec 29 14:02:37.776: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 92, forward
*Dec 29 14:02:37.776: ICMP type=8, code=0
*Dec 29 14:02:37.776: IP: s=10.1.3.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Dec 29 14:02:37.776: ICMP type=11, code=0

*Dec 29 14:02:37.780: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 92, forward
*Dec 29 14:02:37.780: ICMP type=8, code=0
*Dec 29 14:02:37.780: IP: s=10.1.3.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Dec 29 14:02:37.780: ICMP type=11, code=0
*Dec 29 14:02:37.780: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 92, forward
*Dec 29 14:02:37.780: ICMP type=8, code=0
*Dec 29 14:02:37.784: IP: s=10.1.3.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 56, forward
*Dec 29 14:02:37.784: ICMP type=11, code=0
```

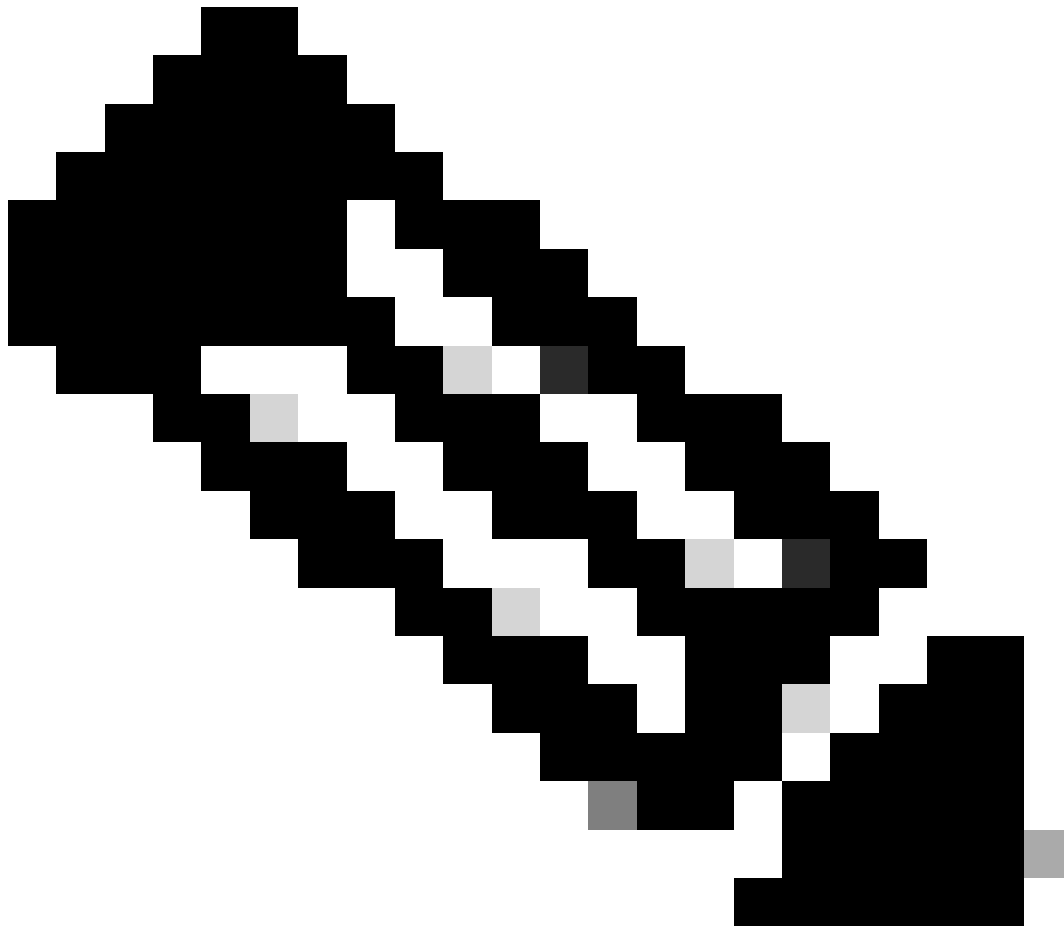
You see the next three ICMP probes in this debug output. The TTL for these probes is 3. Device 11A decrements the TTL to 2 and forwards them on to Device 7A. Device 7A decrements the TTL to 1 and forwards the packets on to Device 7B, which decrements the TTL to zero and responds with ICMP time exceeded messages.

<#root>

```
*Dec 29 14:02:43.292: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 92, forward
*Dec 29 14:02:43.292: ICMP type=8, code=0
*Dec 29 14:02:43.296: IP: s=10.1.4.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 92, forward
*Dec 29 14:02:43.296: ICMP type=0, code=0

*Dec 29 14:02:43.296: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 92, forward
*Dec 29 14:02:43.296: ICMP type=8, code=0
*Dec 29 14:02:43.300: IP: s=10.1.4.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 92, forward
*Dec 29 14:02:43.300: ICMP type=0, code=0
*Dec 29 14:02:43.300: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
  g=10.1.2.2, len 92, forward
*Dec 29 14:02:43.300: ICMP type=8, code=0
*Dec 29 14:02:43.304: IP: s=10.1.4.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
  g=10.1.1.1, len 92, forward
*Dec 29 14:02:43.304: ICMP type=0, code=0
```

This debug output shows the last three ICMP probes. The original TTL of these probes was 4. The TTL was decremented to 3 by Device 11A, then decremented to 2 by Device 7A, then decremented to 1 by Device 7B. Device 7C then responds with ICMP echo reply messages (type=0, code=0), since it was the destination of the probes.



Note: The ICMP echo reply messages are not rate limited as the ICMP port unreachable messages were. In this case, you see all three ICMP echo reply messages sent.

Additional Notes

In Cisco routers, the codes for a `traceroute` command reply are:

```
! -- 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)
```

If you run the `traceroute` command from UNIX, note these items:

- You can receive `traceroute: icmp socket: Permission denied` messages.
- The `traceroute` program relies on the Network Interface Tap (NIT) to snoop in the network. This device is only accessible by root. You must either run the program as root, or set the user ID for root.

Summary

This document has demonstrated how the `traceroute` command determines the path a packet takes from a given source to a given destination with the use of UDP and ICMP packets. The possible types of ICMP messages in the outputs are:

- If the TTL is exceeded in transit, `type=11, code=0`, then the packet is sent back by the transit router in all the cases where the TTL of the probe packets expires before the packets reach the destination.
- If the port is unreachable, `type=3, code=3`, then the packet is sent back in response to the UDP probe packets when they reach the destination (the UDP application is not defined). These packets are limited to one packet per 500 ms. This explains why the response from the destination (see the outputs for the [Cisco router](#) and [Linux](#)) failed in the even responses. Device 7C does not generate the ICMP message, and the `traceroute` command output in each device waits for more than one second. In the case of the MS Windows `tracert` command output, the ICMP message is generated because the UDP port 137 does not exist in a Cisco router.
- If there is an echo, `type=8, code=0`, then the echo probe packet is sent by the MS Windows PC.
- If there is an echo reply, `type=0, code=0`, then a reply to the previous packet is sent when the destination is reached. This only applies to the MS Windows `tracert` command.

Related Information

- [Cisco Technical Support & Downloads](#)