

# 在作業系統上使用 Traceroute 命令

## 目錄

---

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[慣例](#)

[背景資訊](#)

[一般操作](#)

[Cisco IOS和Linux](#)

[Microsoft Windows](#)

[ICMP不可達速率限制](#)

[範例](#)

[使用Cisco IOS軟體的Cisco路由器](#)

[使用Linux的PC](#)

[運行MS Windows的PC](#)

[其他說明](#)

[摘要](#)

[相關資訊](#)

---

## 簡介

本文檔介紹 traceroute 命令如何在不同系統上運行。

必要條件

需求

本文檔讀者必須具備以下作業系統之一的基本知識：

- Cisco IOS®軟體

- Linux

-

## 採用元件

本檔案中的資訊適用於以下軟體和硬體版本：

- 運行Cisco IOS軟體版本12的Cisco路由器
- 運行Red Hat Linux的PC
- 運行MS Windows的PC

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

## 慣例

如需文件慣例的詳細資訊，請參閱思科技術提示慣例。

## 背景資訊

traceroute 命令允許您透過返回資料包所經過的躍點序列，確定資料包從給定源到達目的地所透過的路徑。此公用程式隨附主機作業系統(例如，Linux或Microsoft (MS) Windows)以及Cisco IOS軟體。

## 一般操作

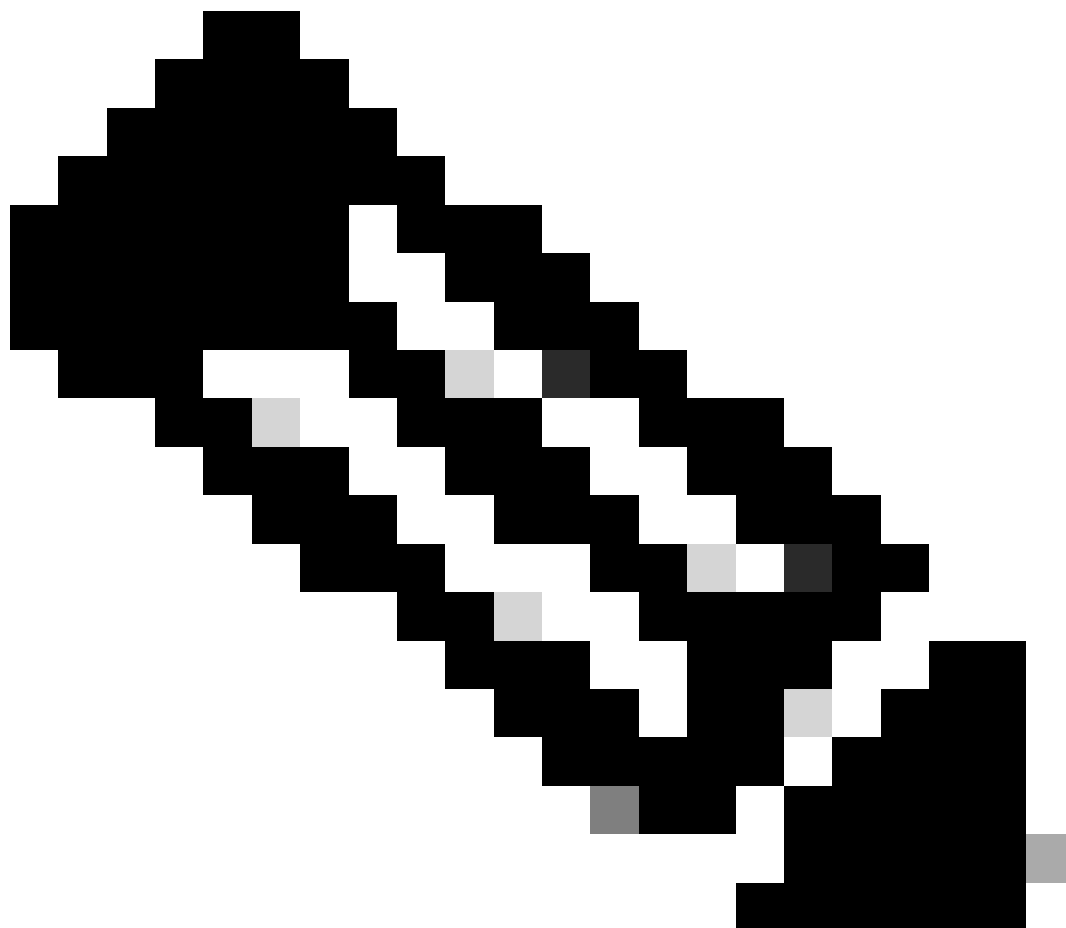
如果您執行 traceroute ip-address 命令在源裝置（例如主機或充當主機的路由器）上傳送IP資料包，這些資料包的存活時間(TTL)值會遞增到指定的最大跳數。預設值為30。通常，在通往目的地的路徑中，每台路由器在轉發這些資料包時，都會將TTL欄位遞減一個單位。當路徑中間的路由器發現TTL = 1的封包時，便會以網際網路控制訊息通訊協定(ICMP)「超過時間」訊息回應來源。此訊息會讓來源知道封包會以躍點形式透過該特定路由器

在本文檔討論的各種作業系統中，traceroute 命令的實現方式有所不同。

## Cisco IOS和Linux

初始使用者資料包協定(UDP)資料包探報的TTL設定為1(或為使用者在 traceroute 擴展命令中指定的最小TTL)。初始資料包探報的目標UDP埠設定為33434(或者按照擴展 traceroute 命令輸出中的指定設定)。extended traceroute 命令是普通 traceroute 命令的變體，它允許對 traceroute 操作使用的引數預設值 ( 如TTL和目標埠號 ) 進行修改。有關如何使用擴展 traceroute 命令的詳細資訊，請參閱[瞭解擴展 ping和擴展traceroute命令](#)。初始資料包探報的源UDP埠是隨機的，並且具有0x8000的邏輯運算子OR ( 確保最小源埠為0x8000 )。以下步驟說明啟動UDP資料包時發生的情況：

---



**注意：**引數是可配置的。此示例以n = 1開頭，以n = 3結尾。

---

•  
UDP資料包的傳送方式為TTL = 1，目標UDP埠= 33434，並且源埠是隨機的。

- UDP目標埠遞增，源UDP埠隨機化，然後傳送第二個資料包。

- 最多對三個探測重複第2步(或在擴展 traceroute 命令輸出中請求的次數)。對於傳送的每個探測，您都會收到「超出TTL」消息，該消息用於構建指向目標主機的分步路徑。

- TTL將遞增，如果收到ICMP超時消息，此循環將以遞增目標埠號重複。您也可以取得以下訊息之一：

- ICMP型別3，代碼3（目的地無法連線，連線埠無法連線）訊息，表示已到達主機。

- 主機無法連線、網路無法連線、超過最大TTL，或訊息的逾時型別，表示重新傳送探測。

Cisco路由器使用隨機源埠和增量目標埠傳送UDP探測資料包（以區分不同的探測）。Cisco路由器將ICMP消息超時傳送回接收UDP/ICMP資料包的源。

Linux traceroute 命令類似於Cisco路由器的實施。但是，它使用固定的源埠。其 -n traceroute 命令中的選項用於避免對名稱伺服器的請求。

Microsoft Windows

MS Windowstracert 命令使用ICMP Echo請求資料包而不是作為探報的UDP資料包。ICMP Echo請求使用遞增TTL啟動，[Cisco IOS](#)和[Linux](#) 中也會發生同樣的操作。使用ICMP回應請求資料包的意義在於，最終跳並不依賴於來自目標主機的ICMP不可達消息的響應。它依賴於ICMP回應應答消息。

指令語法為：

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

下表對命令引數進行了說明：

參數	說明
-d	指定不將地址解析為電腦名稱。
-h maximum_hops	指定搜尋目標的最大躍點數。
-j computer-list	指定沿電腦清單的鬆散源路由。
-w 超時	等待每個回覆的超時指定的毫秒數。
目標名稱	目標電腦的名稱。

#### ICMP不可達速率限制

在Cisco路由器中，ICMP不可達資訊被限制為每500毫秒一個資料包(作為拒絕服務(DoS)攻擊的保護)。從Cisco IOS軟體版本12.1及更高版本開始，此速率值是可配置的。引入的命令為：

```
<#root>
```

```
Router(config)#
```

```
ip icmp rate-limit unreachable ?
```

```
<1-4294967295> Once per milliseconds  
DF code 4, fragmentation needed and DF set
```

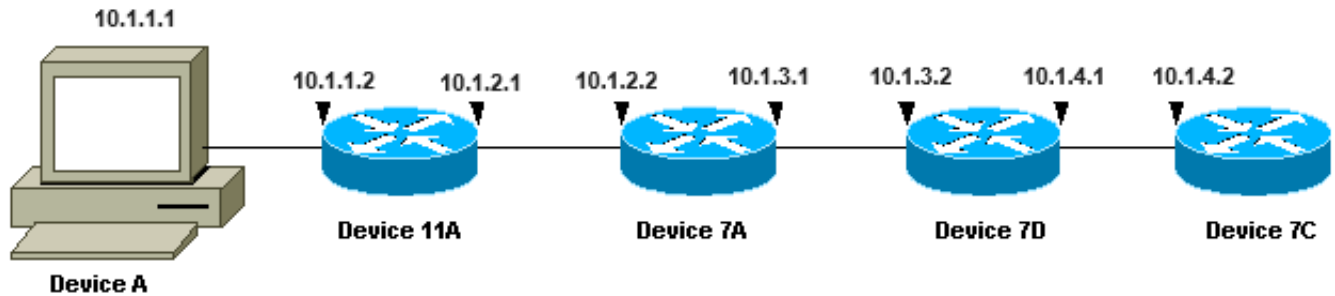
此限制適用於所有ICMP不可達裝置的聚合速率，如此輸出所示。有關詳細資訊，請參閱[RFC 792](#)。

```
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.
```

此限制不影響其他資料包，如ICMP回應請求或ICMP超時消息。

## 範例

以下為網路拓撲的範例：



## 網路拓撲

三個示例中的每一個都使用不同的裝置A。從Device A開始， traceroute 10.1.4.2 命令執行到Device 7C。

在每個示例中， debug ip packet detail 命令在裝置11A上運行。

使用Cisco IOS軟體的Cisco路由器

traceroute 此擴展命令示例顯示當從Cisco路由器執行 traceroute 命令時可以更改的選項。在此範例中，所有專案均保留預設值：

```
<#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
```

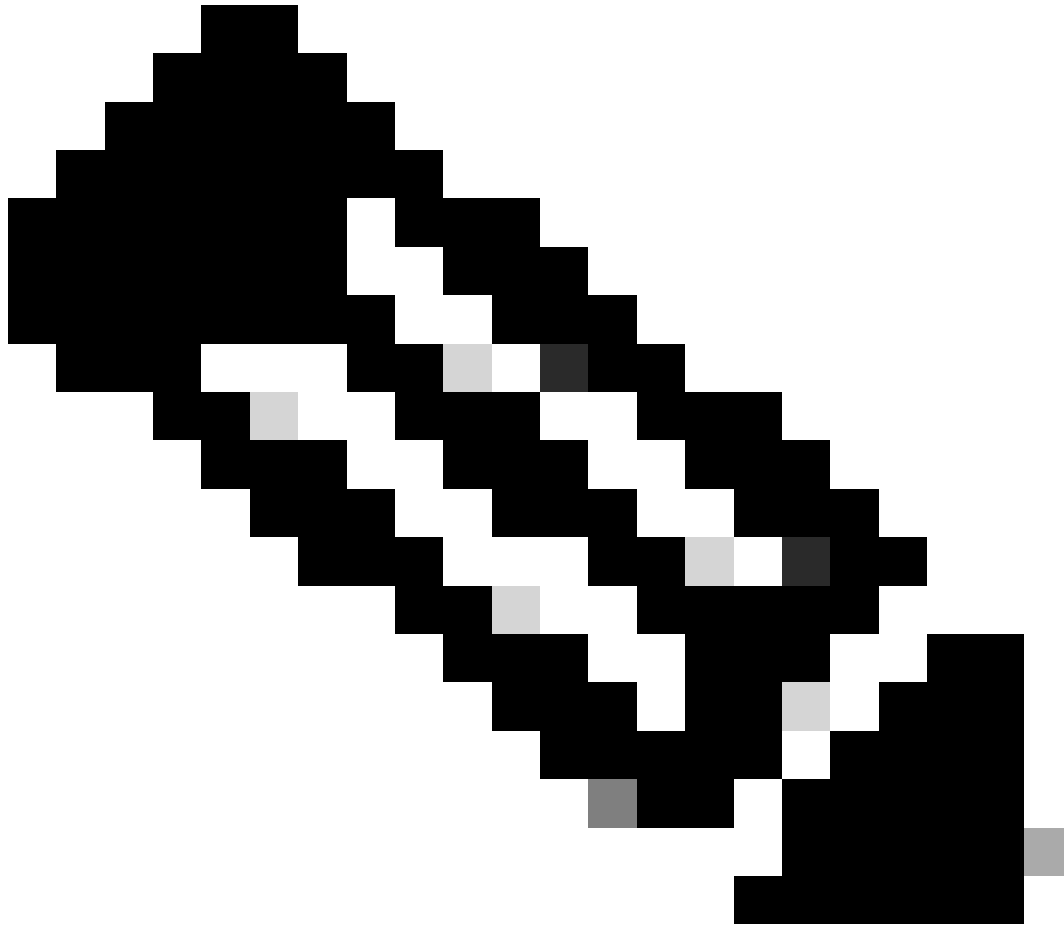
```
1 10.1.1.2 4 msec 0 msec 4 msec
2 10.1.2.2 4 msec 4 msec 0 msec
3 10.1.3.2 0 msec 0 msec 4 msec
4 10.1.4.2 4 msec * 0 msec
```

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
```

在此調試輸出中，裝置11A向探測的源(10.1.1.1)傳送ICMP超時消息。這些ICMP消息用於響應TTL = 1的初始探測。裝置11A將TTL遞減至零，並使用超時消息進行響應。



**注意：**出於兩個原因，您在調試輸出中看不到UDP探報：

---

- 

裝置11A不是UDP探測的目的地。

- 

TTL會遞減至零，且永遠不會路由封包。因此，調試程式永遠不會辨識資料包。

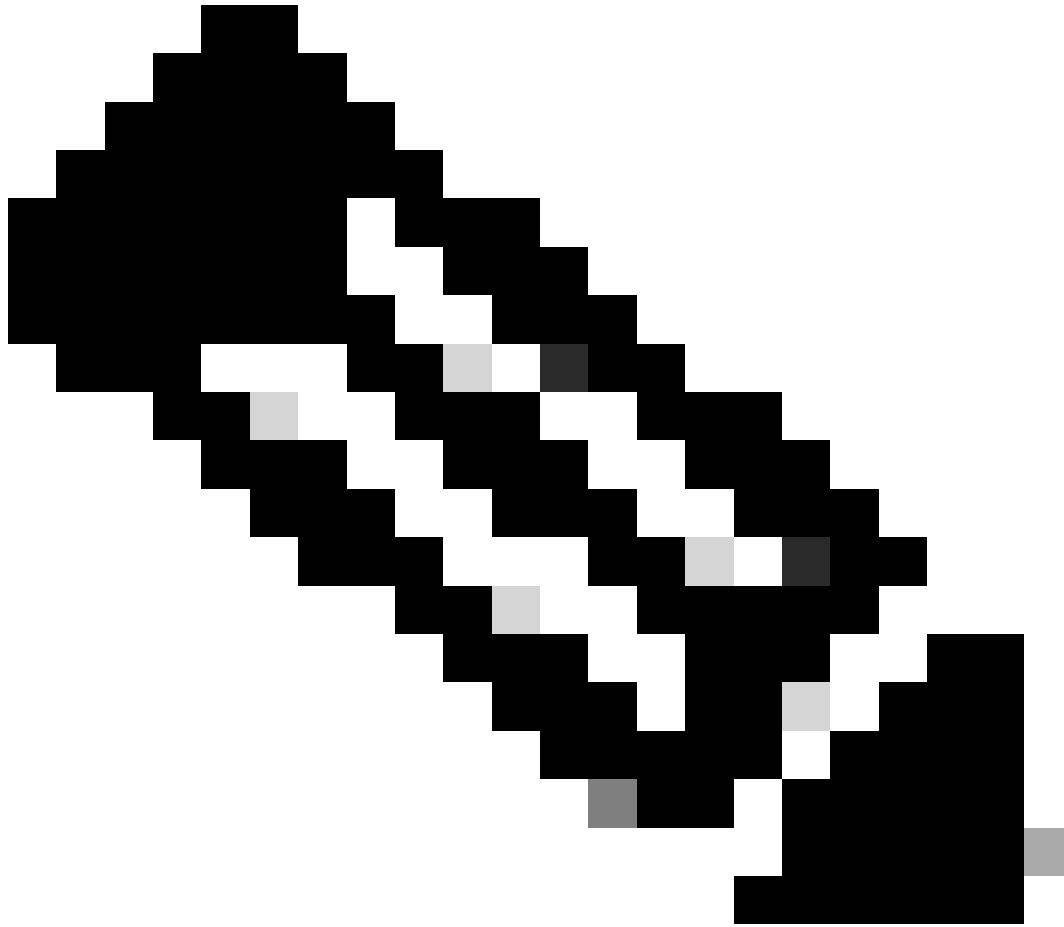


<#root>

```
*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
```

此調試輸出顯示從源10.1.1.1到10.1.4.2的UDP探頭。



**注意：**在這些探測中，TTL=2 ( 使用調試時看不到這種情況 )。裝置11A將TTL遞減為1並將UDP資料包轉發到裝置7A。Device 7A將TTL遞減至零，並以ICMP超出時間消息作出響應。

---

<#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
```

在此調試輸出中，您將看到以下三個UDP探報。這些探查的TTL為3。裝置11A將TTL遞減為2，並將其轉發到裝置7A。Device 7A將TTL遞減為1，並將資料包轉發到Device 7B，後者將TTL遞減為0並以ICMP超時消息進行響應。

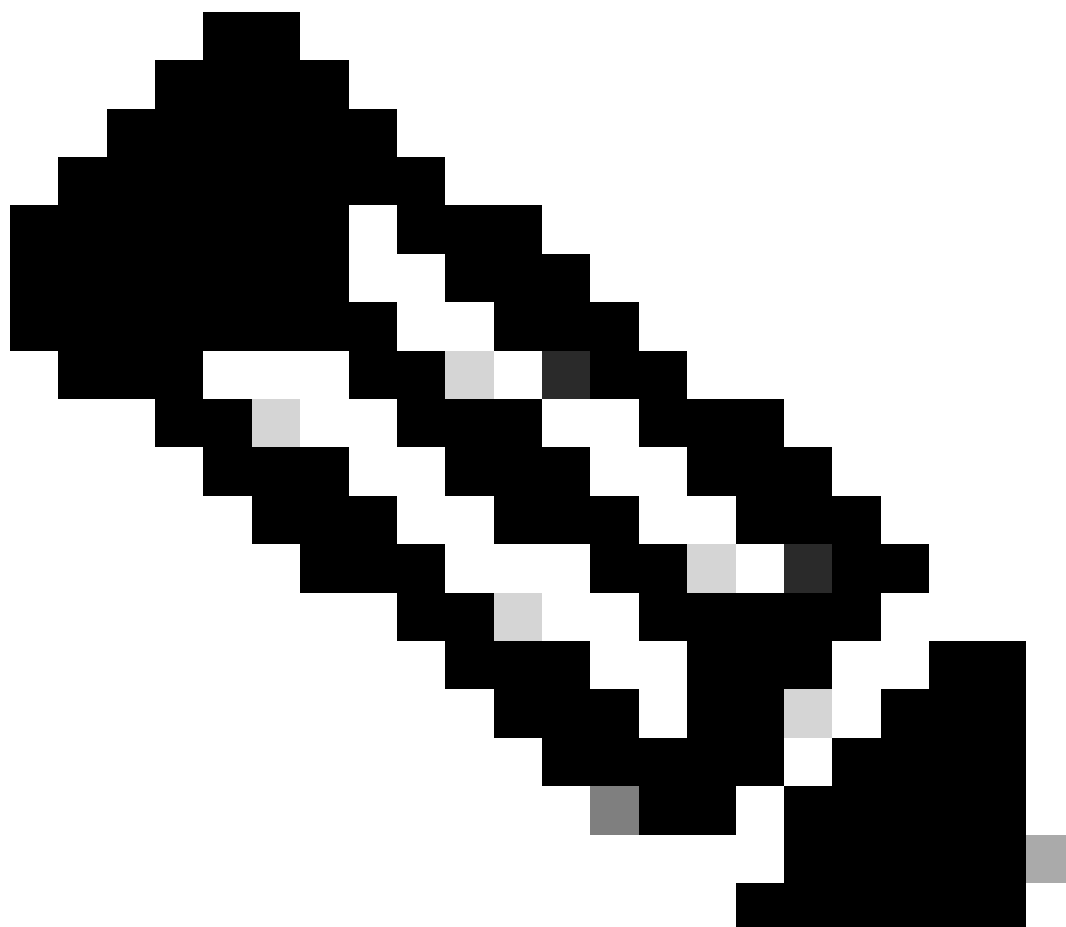
<#root>

```
*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
```

您可以在此調試輸出中看到最後三個UDP探報。這些探針的原始TTL為4。裝置11A的TTL遞減為3，然後裝置7A遞減為2，然後裝置7B遞減為1。裝置7C以ICMP埠不可達消息作出響應，因為它是探測的目標。

---



**注意：**由於速率限制，Device 7C僅傳送兩條ICMP埠不可達消息。

---

使用Linux的PC

<#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
```

在此調試輸出中，裝置11A向探測的源(10.1.1.1)傳送ICMP超時消息。這些ICMP消息用於響應TTL = 1的初始探測。裝置11A將TTL遞減至零，並使用超時消息進行響應。

出於兩個原因，您在調試輸出中看不到UDP探報：

- 

裝置11A不是UDP探測的目的地。

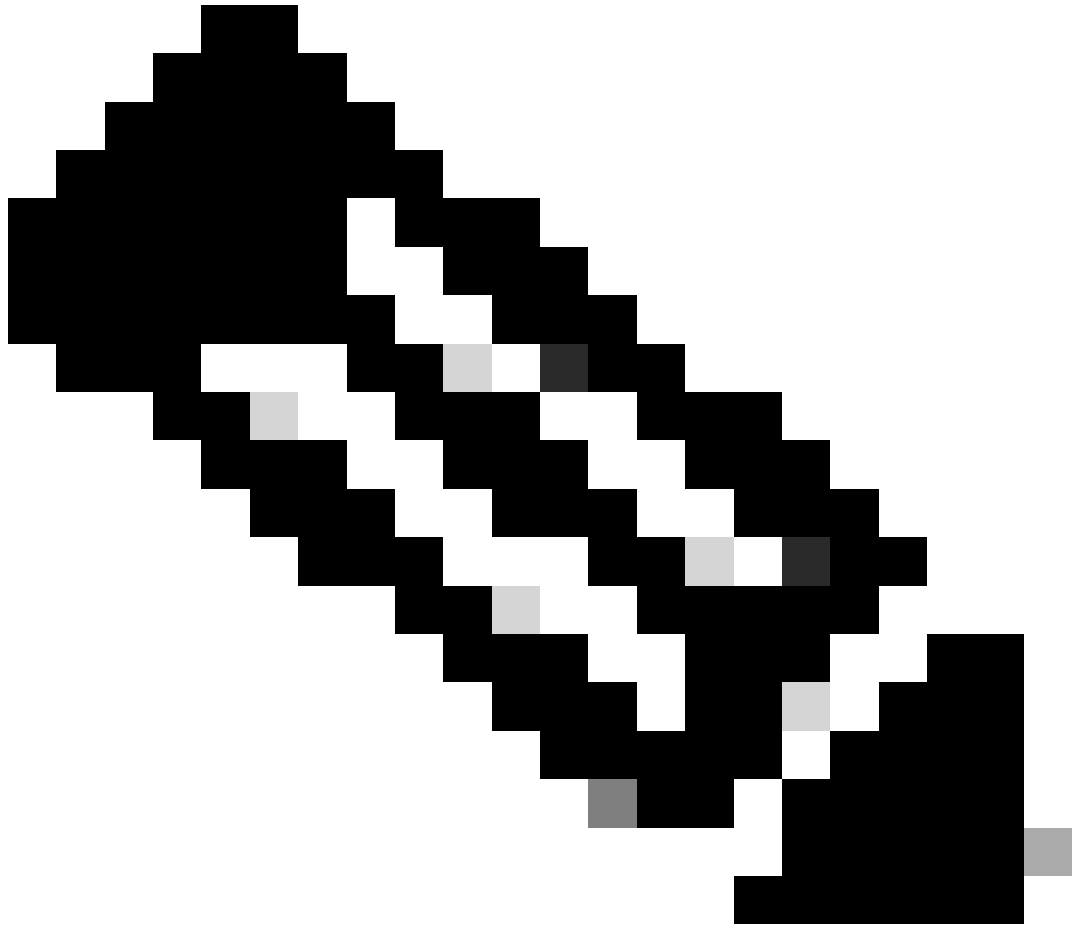
- 

TTL會遞減至零，且永遠不會路由封包。因此，調試程式永遠不會辨識資料包。

<#root>

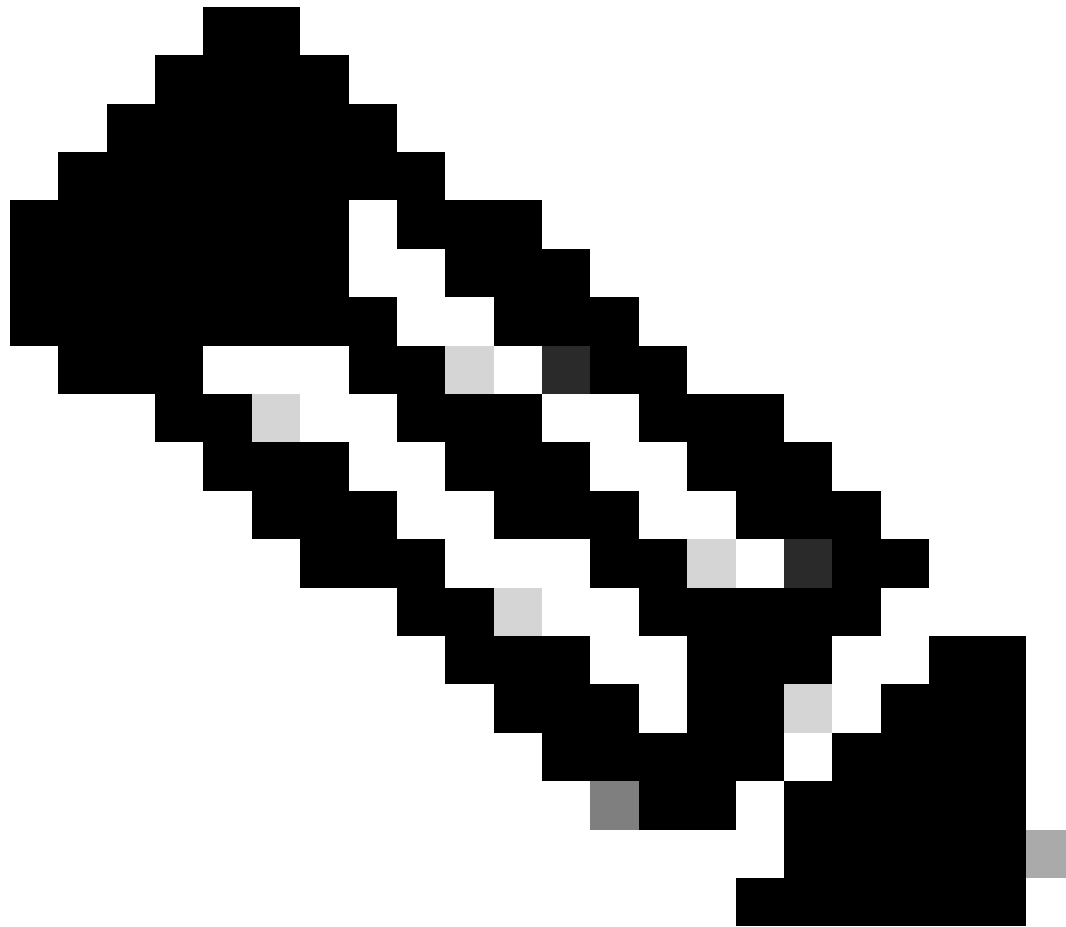
\*Jan 2 07:17:27.894: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2(FastEthernet0/0),  
g=10.1.2.2, len 40, forward  
\*Jan 2 07:17:27.894: UDP src=33302, dst=33438  
\*Jan 2 07:17:27.898: IP: s=10.1.2.2 (FastEthernet0/0), d=10.1.1.1(Ethernet4/0),  
g=10.1.1.1, len 56, forward  
\*Jan 2 07:17:27.898: ICMP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



**注意：**在此調試輸出中，您將看到從源10.1.1.1到10.1.4.2的UDP探頭。

---



**注意：**在這些探測中，TTL=2 ( 使用調試時看不到這種情況 )。裝置11A將TTL遞減為1並將UDP資料包轉發到裝置7A。Device 7A將TTL遞減至零，並以ICMP超出時間消息作出響應。

---

<#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
```

以下三個UDP探測現在可以在此調試輸出中看到。這些探查的TTL為3。裝置11A將TTL遞減為2，並將其轉發到裝置7A。Device 7A將TTL遞減為1，並將資料包轉發到Device 7B，後者將TTL遞減為0並以ICMP超時消息進行響應。

<#root>

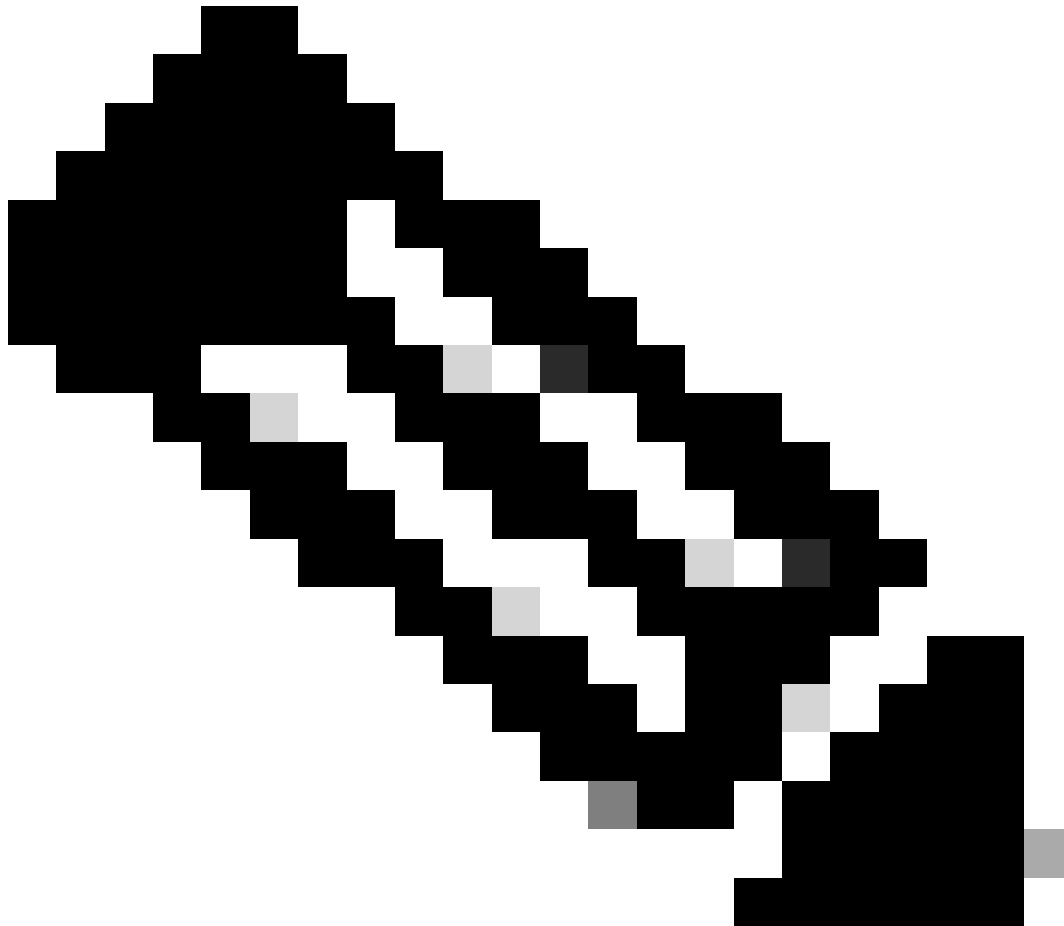
```
*Jan 2 07:17:27.910: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2(FastEthernet0/0),
g=10.1.2.2, len 40, forward
*Jan 2 07:17:27.910: UDP src=33302, dst=33444
*Jan 2 07:17:27.914: IP: s=10.1.4.2 (FastEthernet0/0), d=10.1.1.1(Ethernet4/0),
g=10.1.1.1, len 56, forward
*Jan 2 07:17:27.914: ICMP type=3, code=3

*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
```

此調試輸出顯示了最後三個UDP探報。這些探針的原始TTL為4。裝置11A的TTL遞減為3，然後裝置7A遞減為2，然後裝置7B遞減為1。

裝置7C隨後以ICMP埠不可達消息作出響應，因為它是探測的目標。

---



**注意：**由於速率限制，Device 7C僅傳送兩條ICMP埠不可達消息。

---

運行MS Windows的PC

<#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
```

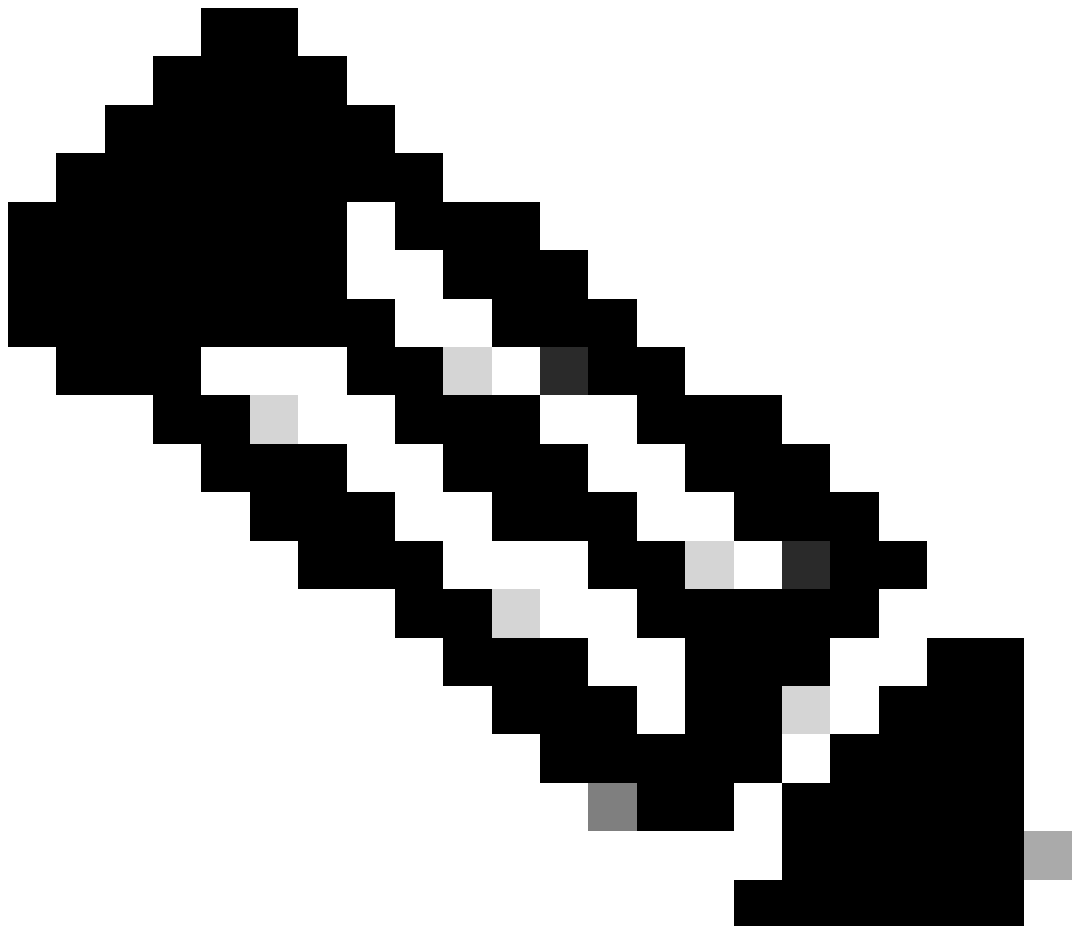
```
*Dec 29 14:02:23.732: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),
g=10.1.2.2, len 78, forward
*Dec 29 14:02:23.732: UDP src=137, dst=137
*Dec 29 14:02:23.736: IP: s=10.1.4.2 (FastEthernet0/0), d=10.1.1.1 (Ethernet4/0),
g=10.1.1.1, len 56, forward
*Dec 29 14:02:23.736: ICMP type=3,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

在此調試輸出中，裝置11A向探測的源(10.1.1.1)傳送ICMP超時消息。這些ICMP消息用於響應初始探測，這些探測是TTL=1的ICMP響應請求資料包。裝置11A將TTL遞減至零，並以ICMP消息做出響應。



**注意：**在頂部您會看到NETBIOS名稱請求。這些請求被視為源埠和目標埠為137的UDP資料包。為清楚起見，NETBIOS資料包將從調試輸出的其餘部分中刪除。可以使用 `tracert` 命令中的 `-d` 選項停用NETBIOS行為。

出於兩個原因，您在此調試輸出中看不到ICMP探報：

- 

裝置11A不是ICMP探測的目的地。

- 

TTL會遞減至零，且永遠不會路由封包。因此，調試程式永遠不會辨識資料包。

<#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
```

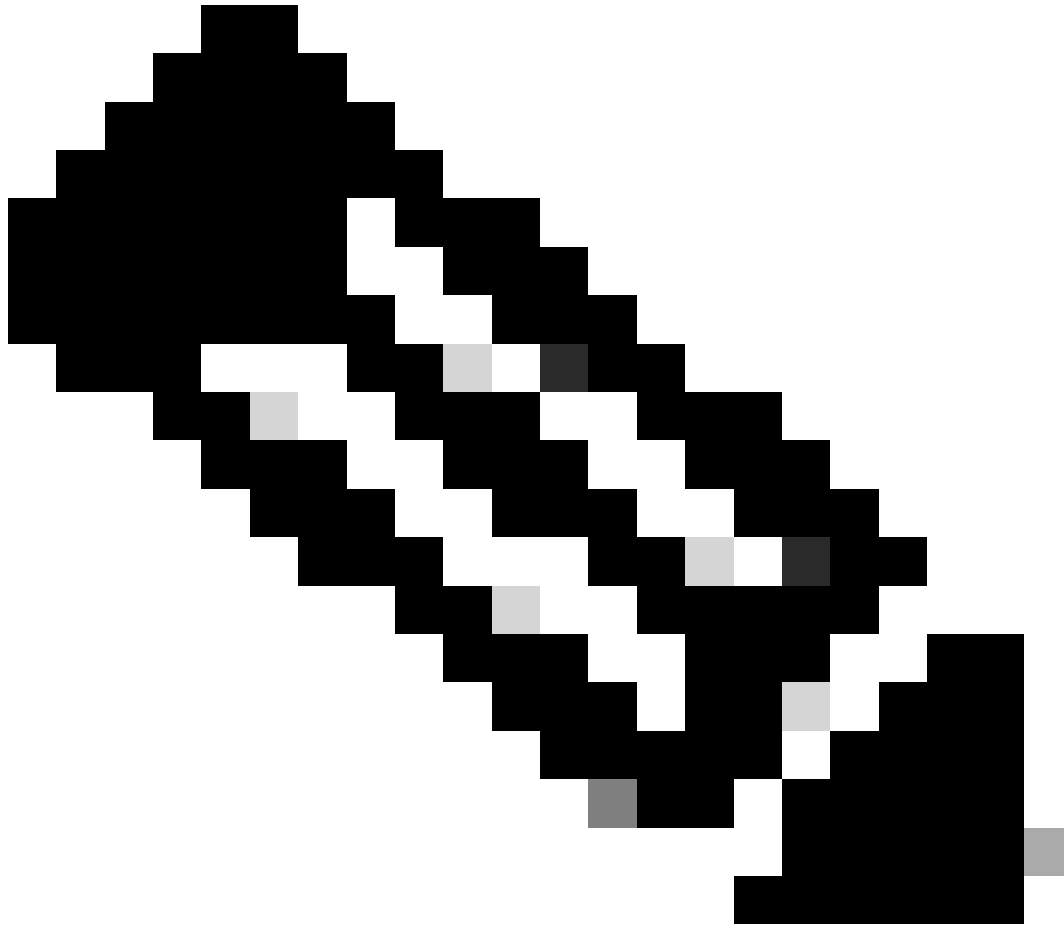
```
*Dec 29 14:02:32.264: IP: s=10.1.1.1 (Ethernet4/0), d=10.1.4.2 (FastEthernet0/0),  
g=10.1.2.2, len 92, forward
```

```
*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
```

在此調試輸出中，現在您將看到從源10.1.1.1到10.1.4.2的ICMP探報。



**注意：**在這些探測中，TTL = 2 ( 透過debug看不到這種情況 )。裝置11A將TTL遞減為1並將UDP資料包轉發到裝置7A。Device 7A將TTL遞減至零，並以ICMP超出時間消息作出響應。

---

<#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
```

您可以在此調試輸出中看到以下三個ICMP探報。這些探查的TTL為3。裝置11A將TTL遞減為2，並將其轉發到裝置7A。Device 7A將TTL遞減為1，並將資料包轉發到Device 7B，後者將TTL遞減為0並以ICMP超時消息進行響應。

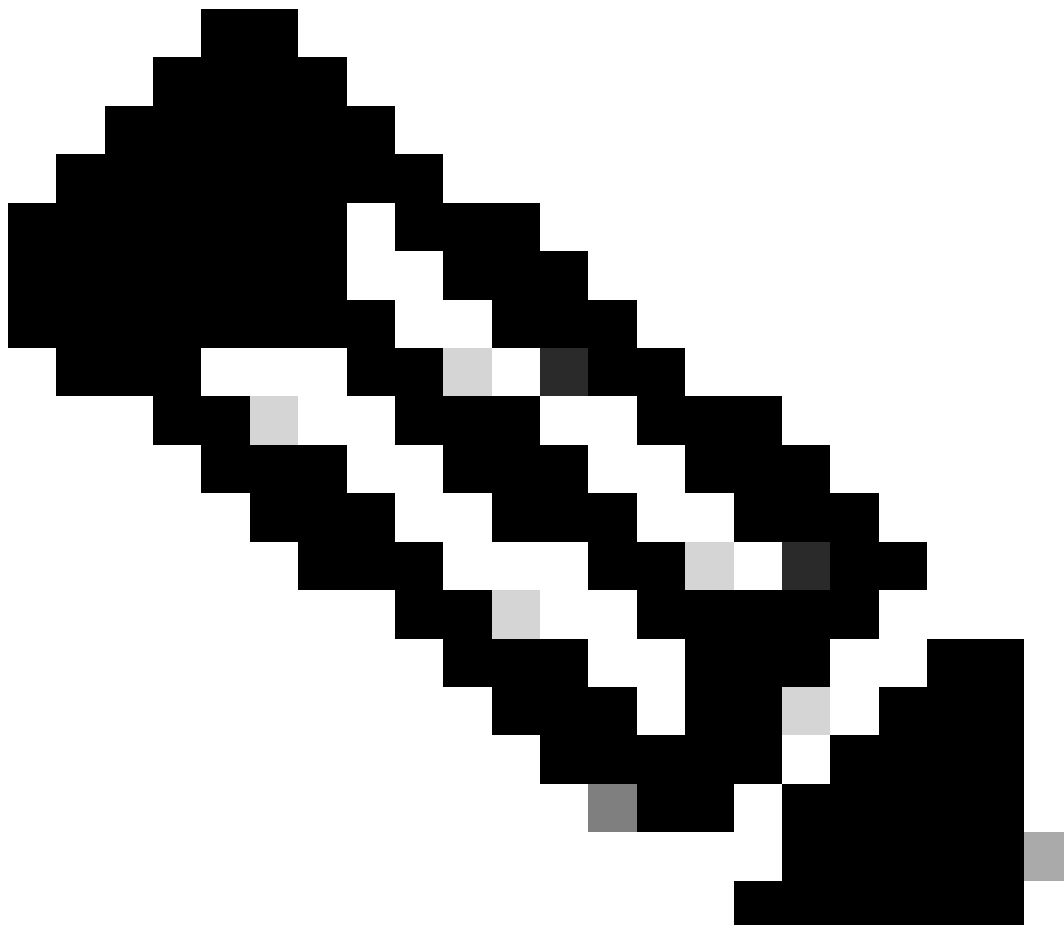
<#root>

```
*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
```

此調試輸出顯示最後三個ICMP探查。這些探針的原始TTL為4。裝置11A的TTL遞減為3，然後裝置7A遞減為2，然後裝置7B遞減為1。裝置7C隨後以ICMP回應應答消息（型別=0，代碼=0）進行響應，因為它是探測的目標。

---



**注意：**ICMP回應應答消息不受速率限制，而ICMP埠不可達消息不受速率限制。在這種情況下，您將看到所有三條ICMP應答消息都已傳送。

---

#### 其他說明

在Cisco路由器中，traceroute 命令回覆的代碼如下：



! -- 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)

如果從UNIX運行 traceroute 命令，請注意以下幾點：

•

您可以收到 traceroute: icmp socket: Permission denied 訊息。

traceroute

•

程式依靠網路介面分流器(NIT)來窺視網路。此裝置只能由root使用者訪問。您必須以root身份執行程式，或設定root的使用者ID。

## 摘要

本文檔演示了如何使用UDP和ICMP資料包，透過 traceroute 命令確定資料包從給定源到給定目的地的路徑。輸出中可能的ICMP消息型別包括：

•

如果在傳輸中超過TTL，type=11，code=0，則在所有探測資料包的TTL在資料包到達目的地之前過期的情況下，傳輸路由器都會將資料包傳送回。

•

如果連線埠無法連線(type=3，code=3)，則當封包到達目的地時（未定義UDP應用程式），系統會傳回封包以回應UDP探測封包。這些資料包限制為每500毫秒一個資料包。這解釋了為什麼來自目標的響應(請參閱[Cisco路由器](#)和[Linux](#)的輸出)在Even響應中失敗。Device 7C不生成ICMP消息，並且 traceroute 每個裝置中的命令輸出等待超過一秒。在MS Windows中tracert 命令輸出中，由於Cisco路由器中不存在UDP埠137，因此生成ICMP消息。

•

如果存在回聲，型別=8，代碼=0，則迴聲探測資料包由MS Windows PC傳送。

- 

如果有回應答，type=0，code=0，則到達目的地時會傳送對上一個資料包的應答。這只適用於MS Windows tracert 指令。

#### 相關資訊

- [思科技術支援與下載](#)

## 關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。