

# MPLS LSP多重路徑追蹤

## 目錄

---

[簡介](#)

[背景資訊](#)

[LSP樹跟蹤 — 工作原理](#)

[LSP樹跟蹤 — 詳細示例](#)

---

## 簡介

MPLS LSP Ping是一個基本工具，用於驗證入口和出口之間的標籤交換路徑(LSP)的運行狀況。本文檔旨在解釋LSP樹跟蹤中啟動器與響應方之間的多路徑資訊互動。如需此工具可用的詳細選項，請參閱本文[件很有用](#)。

## 背景資訊

MPLS EM - MPLS LSP多重路徑樹跟蹤功能的實現基於RFC 4379, Detecting Multi-Protocol Label Switched(MPLS)Data Plane Failures。

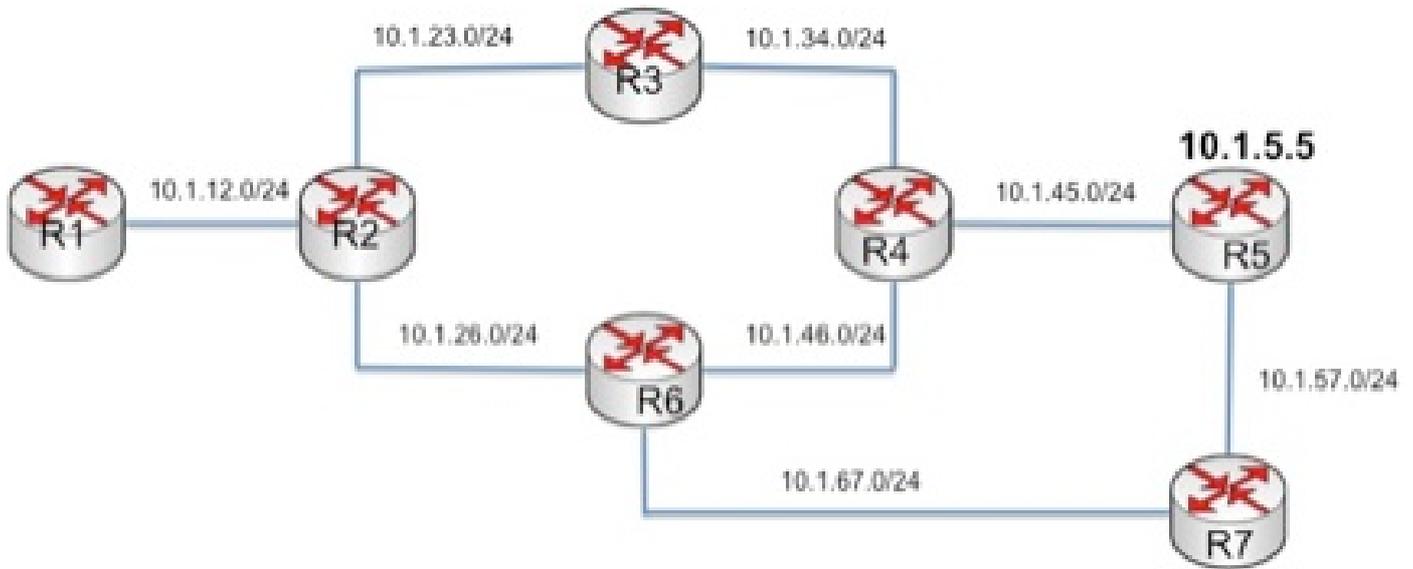
通過將探測資料包的IP目標地址設定為環回地址(127.x.x.x),LSP樹跟蹤可以通過避免資料包獲得IP路由來檢測LSP中的故障。因此，無論何時出現端到端連線問題，使用LSP Ping作為消除任何LSP故障的第一步都非常有用。

在多路徑場景中，LSP ping可能並不總是有助於識別所有LSP故障。如前所述，任何標籤交換路由器(LSR)在接收標籤分組時都使用分組中的特定鍵，並輸入到雜湊演算法以決定輸出介面，該標籤分組可被從多個輸出介面發出。根據供應商、硬體等的不同，雜湊時可以考慮以下任何選項：

1. 僅傳入標籤堆疊。
2. 傳入標籤堆疊和IP報頭詳細資訊（如果負載為IP）。
3. 傳入標籤堆疊、IP報頭和傳輸報頭詳細資訊。

通常，如果堆疊的大小小於或等於3（並將IP作為負載），思科路由器會考慮標籤堆疊和IP報頭的組合。

採用以下拓撲。



R1-R7是路由器。在上面的拓撲中，有3條從R1到R5的等價多路徑(ECMP)路由，如下所示，

路徑1:R1-R2-R3-R4-R5

路徑2:R1-R2-R6-R4-R5

路徑3:R1-R2-R6-R7-R5

假設R6和R7之間存在問題(如損壞的標籤分發協定(LDP)或標籤誤程式設計等)，導致通過PATH3從R1到R5的流量丟棄。如果從R1對LSP執行Ping操作時採用PATH1或PATH2，則最終可能會假設R1和R5之間的路徑是正常的。

LSP Ping允許將IP目標地址設定為127.0.0.0/8範圍內的任意地址。雖然一個簡單的選項是手動嘗試傳送具有不同目的地址的多個ping資料包，但不能保證所有可能的ECMP路徑都將被驗證。您需要一種查詢和驗證源與目標之間所有可能路徑的方法。LSP多路徑樹跟蹤利用RFC4379第3.3.1節中定義的「多路徑資訊編碼」，幫助您驗證所有ECMP路徑。

## LSP樹跟蹤 — 工作原理

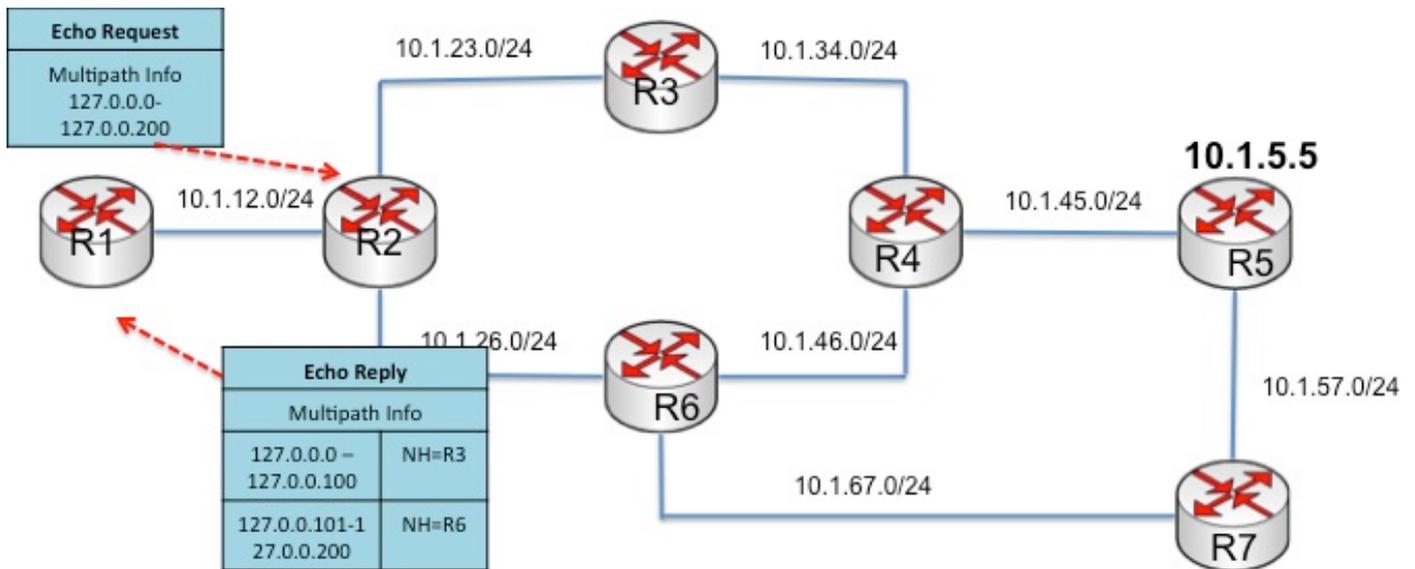
常規MPLS ping或traceroute可能表示沒有故障，具體取決於傳輸路由器如何通過ECMP載入共用資料包，但是LSP樹跟蹤提供了一種更佳的方法來驗證所有路徑是否真正工作。

在LSP樹跟蹤中，發起方路由器通過以增量方式在頂部標籤中設定TTL（從1開始），向每一跳傳送MPLS回應請求。回應要求將攜帶多路徑資訊TLV，它攜帶一個IP地址範圍(在127.0.0.0/8範圍內)或熵標籤範圍。目前Cisco裝置支援IP destination選項，因此我們的示例將詳細介紹IP地址範圍。

接收請求資料包時的每個傳輸LSR將回覆所有ECMP傳出介面，並將來自每個介面的請求的IP地址範圍（或熵標籤）相關聯。

## LSP樹跟蹤 — 詳細示例

假設如下所示的拓撲。



為簡單起見，此示例使用地址範圍127.0.0.0-127.0.0.200。以下是LSP樹跟蹤中步驟的詳細資訊。

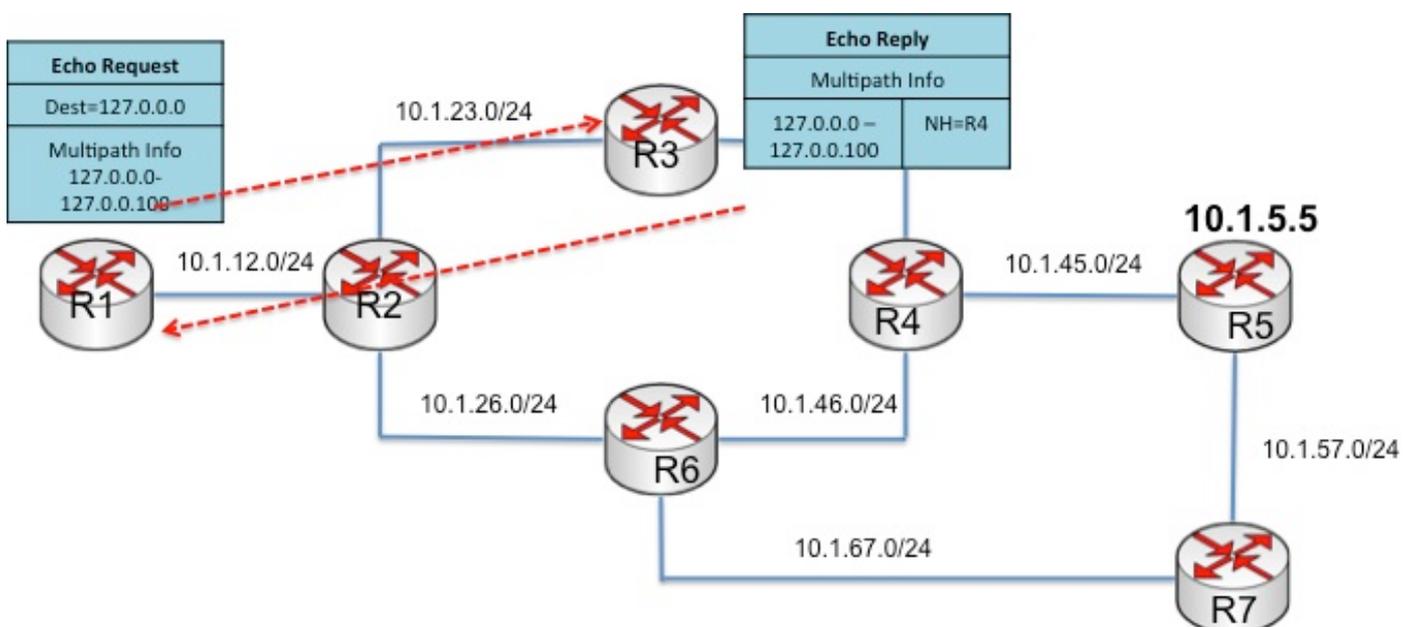
1)發起方(R1)傳送回應要求詳細資訊如下：

- IP目的地127.0.0.0
- 地址範圍從127.0.0.0到127.0.0.200的多路徑資訊TLV。
- 頂部標籤的TTL將設定為1。

2)R2收到相同的消息後，會用每個出口介面的多路徑資訊作出回覆。在本例中，它將按如下方式回覆：

- 如果IP目的地址在127.0.0.0到127.0.0.100之間，則資料包將傳送到R3。
- 如果IP目的地址在127.0.0.101到127.0.0.200之間，則資料包將傳送到R6。

3)R1意識到有2個可能的ECMP路徑，因此它需要傳送2個Echo Request (TTL設定為2)。從各種測試中，發現啟動器始終先以1個路徑結束再轉到下一個路徑。(但對特定實施來說可能確實如此)。

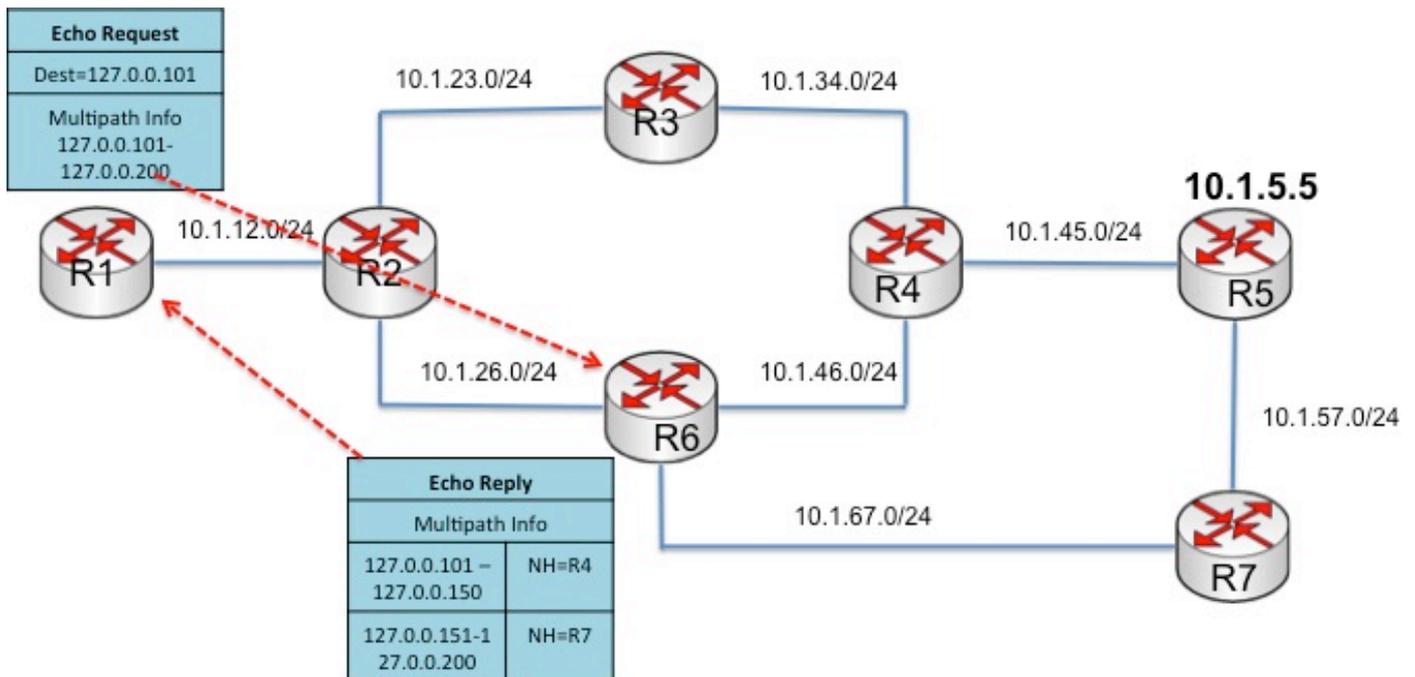


4)R1現在傳送回應請求，詳細資訊如下：

- IP目的地127.0.0.0
- 地址範圍從127.0.0.0到127.0.0.100的多路徑資訊TLV。
- 頂部標籤的TTL將設定為2。

5)R2會將資料包轉發到R3 ( 因為目的地址是127.0.0.0 )。R3收到相同消息時會使用相同的多路徑資訊回覆，因為只有一個輸出介面。

這同樣適用，直到到達R5。



6)完成PATH1跟蹤後 ( 收到出口回覆後 )，發起方現在將查詢PATH2。這是通過傳送回應要求執行的，詳細資訊如下：

- IP目的地as 127.0.0.101
- 地址範圍從127.0.0.101到127.0.0.200的多路徑資訊TLV
- 頂部標籤的TTL設定為2。

7)R2會將資料包轉發到R6 ( 因為目的地址是127.0.0.101 )。R6收到相同資訊後將回覆多路徑資訊，如下所示：

- 如果IP目的地址在127.0.0.101到127.0.0.150之間，則資料包將傳送到R4。
- 如果IP目的地址在127.0.0.151到127.0.0.200之間，則資料包將傳送到R7。

8)R1意識到有一條多的ECMP路徑，使得所有可能的路徑總數為3。R1繼續通過傳送包含下列詳細資訊的下一個回應請求來查詢PATH2:

- IP目的地as 127.0.0.101
- 地址範圍從127.0.0.101到127.0.0.150的多路徑資訊TLV
- 頂部標籤的TTL設定為3。

9)R2將資料包轉發到R6 ( 因為目的地是127.0.0.101 )，R6將資料包轉發到R4 ( 因為目的地是

127.0.0.101 )。R4沒有任何ECMP路徑，因此將使用相同的多路徑資訊回覆。下一個資料包將到達出口R5。

10)由於PATH2跟蹤已完成，R1將繼續查詢PATH3。這是通過傳送回應要求執行的，詳細資訊如下：

- IP目的地as 127.0.0.151
- 地址範圍從127.0.0.151到127.0.0.200的多路徑資訊TLV
- 頂部標籤的TTL設定為3。

11)R2將資料包轉發到R6，然後由R6轉發到R7。R7將用相同的多路徑資訊TLV回覆。下一個資料包到達出口路由器R5。

完成這些步驟後，R1將具有以下詳細資訊：

| Multipath Information |                            |                |
|-----------------------|----------------------------|----------------|
|                       | Address Range              | Path           |
| PATH1                 | 127.0.0.0 to 127.0.0.100   | R1-R2-R3-R4-R5 |
| PATH2                 | 127.0.0.101 to 127.0.0.150 | R1-R2-R6-R4-R5 |
| PATH3                 | 127.0.0.151 to 127.0.0.200 | R1-R2-R6-R7-R8 |

通過在127.0.0.0和127.0.0.100內使用目的地地址，PATH1會影響資料包轉發，而使用其他範圍的地址會影響通過相應路徑轉發資料包。

12)現在，Initiator將傳送3個TTL設定為255的回應請求資料包，並從每個範圍中選擇地址，以便端到端驗證所有路徑。

用於ECMP跟蹤的命令是tracert mpls multipath ipv4 <prefix> <mask>。以下是輸出示例。

<#root>

```
R1#tracert mpls multipath ipv4 10.1.5.5 255.255.255.255
```

```
Starting LSP Multipath Traceroute for 10.1.5.5/32
```

```
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,  
'L' - labeled output interface, 'B' - unlabeled output interface,  
'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,  
'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,  
'P' - no rx intf label prot, 'p' - premature termination of LSP,  
'R' - transit router, 'I' - unknown upstream index,  
'l' - Label switched with FEC change, 'd' - see DDMAP for return code,  
'X' - unknown return code, 'x' - return code 0
```

```
Type escape sequence to abort.
```

```
LLL!
```

```
Path 0 found,  
output interface Et0/0.12 nexthop 10.1.12.2  
source 10.1.12.1 destination 127.0.0.4
```

```
LL!
```

```
Path 1 found,
output interface Et0/0.12 nexthop 10.1.12.2
source 10.1.12.1 destination 127.0.0.2
L!
Path 2 found,
output interface Et0/0.12 nexthop 10.1.12.2
source 10.1.12.1 destination 127.0.0.0

Paths (found/broken/unexplored) (3/0/0)
```

```
Echo Request (sent/fail) (9/0)
Echo Reply (received/timeout) (9/0)
Total Time Elapsed 27 ms
```

請注意，上面的輸出顯示有3個路徑，並且所有路徑都工作正常。在上述命令中使用詳細命令將列出所有躍點，如下所示：

```
<#root>
```

```
R1#traceroute mpls multipath ipv4 10.1.5.5 255.255.255.255 verbose
```

```
Starting LSP Multipath Traceroute for 10.1.5.5/32
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'L' - labeled output interface, 'B' - unlabeled output interface,
'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
'P' - no rx intf label prot, 'p' - premature termination of LSP,
'R' - transit router, 'I' - unknown upstream index,
'l' - Label switched with FEC change, 'd' - see DDMAP for return code,
'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
LLL!
Path 0 found,
output interface Et0/0.12 nexthop 10.1.12.2
source 10.1.12.1 destination 127.0.0.4
0 10.1.12.1 10.1.12.2 MRU 1500 [Labels: 22 Exp: 0] multipaths 0
L 1 10.1.12.2 10.1.23.3 MRU 1500 [Labels: 23 Exp: 0] ret code 8 multipaths 2
L 2 10.1.23.3 10.1.34.4 MRU 1500 [Labels: 22 Exp: 0] ret code 8 multipaths 1
L 3 10.1.34.4 10.1.45.5 MRU 1500 [Labels: implicit-null Exp: 0] ret code 8 multipaths 1
! 4 10.1.45.5, ret code 3 multipaths 0
LL!
Path 1 found,
output interface Et0/0.12 nexthop 10.1.12.2
source 10.1.12.1 destination 127.0.0.2
0 10.1.12.1 10.1.12.2 MRU 1500 [Labels: 22 Exp: 0] multipaths 0
L 1 10.1.12.2 10.1.26.6 MRU 1500 [Labels: 16 Exp: 0] ret code 8 multipaths 2
L 2 10.1.26.6 10.1.46.4 MRU 1500 [Labels: 22 Exp: 0] ret code 8 multipaths 2
L 3 10.1.46.4 10.1.45.5 MRU 1500 [Labels: implicit-null Exp: 0] ret code 8 multipaths 1
! 4 10.1.45.5, ret code 3 multipaths 0
L!
Path 2 found,
output interface Et0/0.12 nexthop 10.1.12.2
source 10.1.12.1 destination 127.0.0.0
0 10.1.12.1 10.1.12.2 MRU 1500 [Labels: 22 Exp: 0] multipaths 0
L 1 10.1.12.2 10.1.26.6 MRU 1500 [Labels: 16 Exp: 0] ret code 8 multipaths 2
L 2 10.1.26.6 10.1.67.7 MRU 1500 [Labels: 17 Exp: 0] ret code 8 multipaths 2
L 3 10.1.67.7 10.1.57.5 MRU 1500 [Labels: implicit-null Exp: 0] ret code 8 multipaths 1
```

! 4 10.1.57.5, ret code 3 multipaths 0  
Paths (found/broken/unexplored) (3/0/0)  
Echo Request (sent/fail) (9/0)  
Echo Reply (received/timeout) (9/0)  
Total Time Elapsed 29 ms

## 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）。