

# 目录

[简介](#)

[背景信息](#)

[LSP树trace -它如何工作](#)

[LSP树trace -详细示例](#)

[相关的思科支持社区讨论](#)

## 简介

MPLS LSP Ping是用于的一个基本工具验证标签交换路径(LSP)的健康入口和出口之间。 本文打算解释多重通道的信息的交互作用在发起者和响应方之间的在LSP树trace。对于详细的选项可用为此工具，参考[本文](#)是有用的。

## 背景信息

MPLS EM的此实施？MPLS LSP多重通道的树Trace功能根据RFC 4379，检测多协议标签交换的(MPLS)数据层面失败。

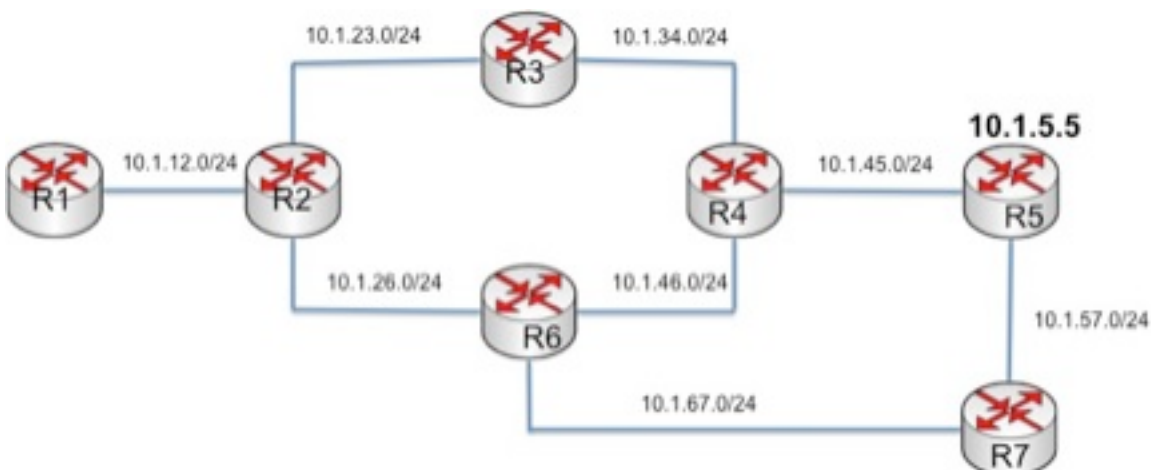
通过设置探针信息包的IP目的地地址作为环回地址(127.x.x.x)，LSP树trace可以用于通过避免数据包检测LSP的失败获得IP路由。因此，每当有端到端连接问题，使用LSP Ping作为第一步排除所有LSP故障是有用的。

在多重通道的方案的情况下，LSP ping可能不总是帮助识别所有LSP故障。因为它也许是要注意的，所有标签交换路由器(LSR)在接收可以是被派出的多出口接口的标记的信息包，使用从数据包和输入的某些密钥到哈希算法决定出口接口。根据供应商、硬件等等，其中任一个下面的选项可能为切细设想：

1. 单独流入的标签的堆叠。
2. 流入的标签堆叠和IP报头详细信息(如果有效负载是IP)。
3. 流入的标签堆叠、IP报头和传输报头详细信息。

通常，Cisco路由器考虑标签栈和IP报头的组合，如果堆叠是大小小于或等于3 (与IP作为有效负载)。

assume根据拓扑。



R1-R7是路由器。在上述拓扑，有3个等价多路径(ECMP)路由从R1到R5作为下面，

PATH1 : R1-R2-R3-R4-R5

PATH2 : R1-R2-R6-R4-R5

PATH3 : R1-R2-R6-R7-R5

假设有在R6和R7之间的问题(类似残破的标签转发协议(LDP)或标记错误程序设计等等)造成流量从R1到R5通过PATH3下降。如果从R1的LSP Ping采取PATH1或PATH2，您可以导致假设，R1和R5之间的路径优良是。

LSP Ping允许设置IP目的地地址作为任何一个从127.0.0.0/8范围。当一简单选项是手工尝试发送多个ping信息包和另外目的地址一起时，没有保证所有可能的ECMP路径将验证。您需要查询并且验证源和目的之间的所有可能的路径的方法。LSP多重通道的树trace有效利用？多重通道的信息编码？定义在RFC4379的部分3.3.1和帮助您验证所有ECMP路径。

## LSP树trace -它如何工作

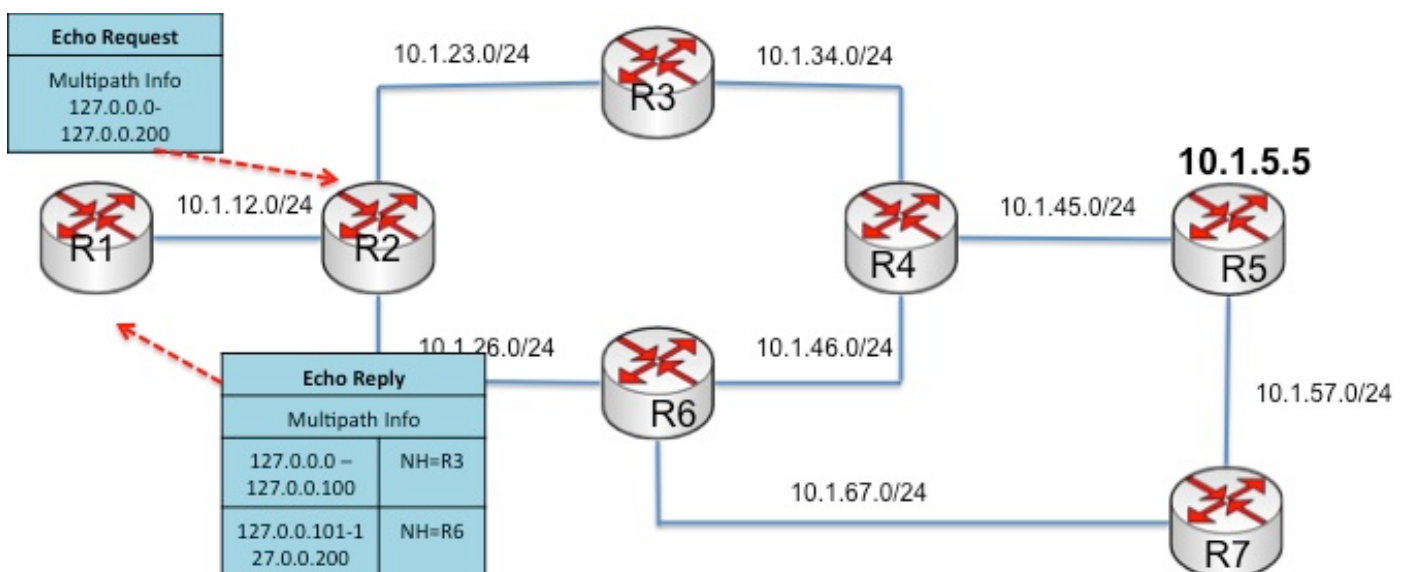
—正常MPLS ping或traceroute可能表明没有失败根据转接路由器如何负载共享在ECMP的数据包，然而LSP树trace提供一个更加好的方法验证所有路径实际上工作。

在LSP树trace，发起者路由器发送MPLS ECHO请求对每跳通过设置在顶部标签的TTL以递增方式(从1)开始。ECHO请求将传播运载范围IP地址的多重通道的信息TLV (在127.0.0.0/8范围之内)或熵标签范围。目前Cisco设备支持IP目的地选项和，因此我们的示例将选派与IP地址范围。

在接收请求包的每传输LSR将回复以所有ECMP流出接口并且关联范围IP地址(或熵标签)从要求每个接口。

## LSP树trace -详细示例

assume根据拓扑例如下面。



为了简化，此示例使用地址范围127.0.0.0-127.0.0.200。这是步骤详细信息在LSP树trace的。

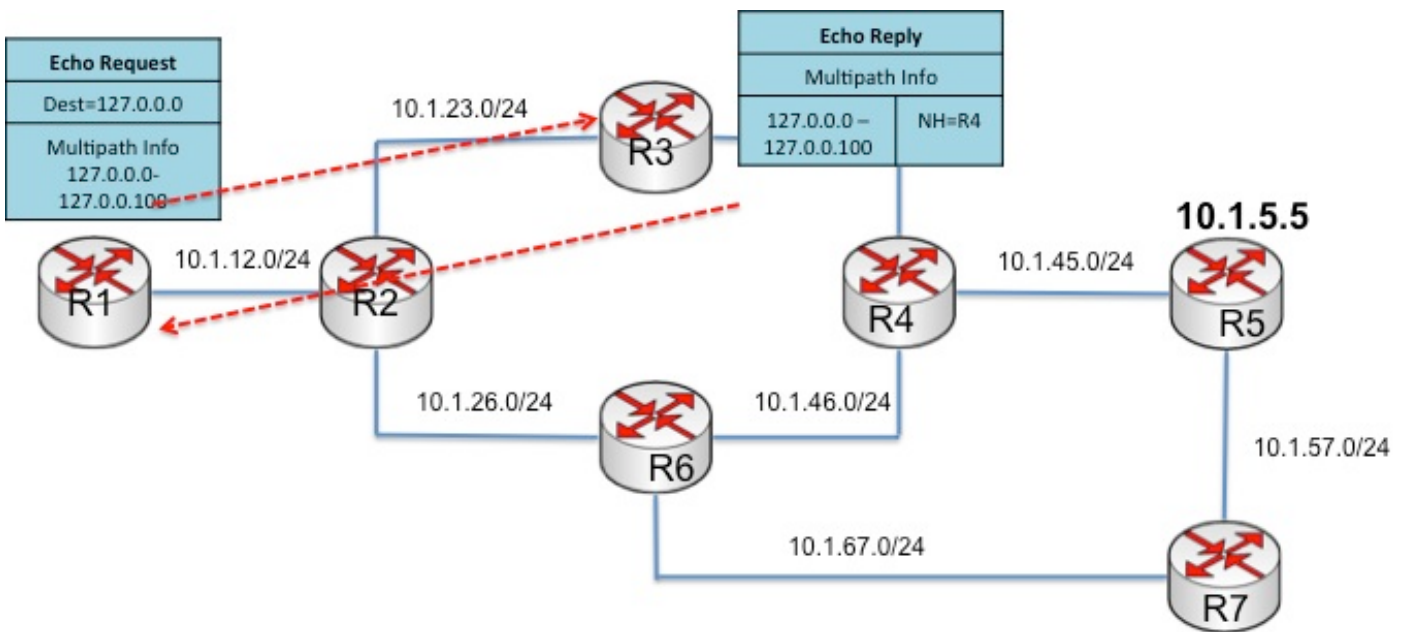
1) 发起者(R1)发送与下面的详细信息的ECHO请求：

- IP目的地作为127.0.0.0
- 运载地址范围的多重通道的信息TLV作为127.0.0.0对127.0.0.200。
- 顶部标签的TTL将设置到1。

2) 在接收同样的R2将应答有多重通道的信息的上一步每出口接口的。在本例中，它将应答作为如下：

- 如果IP目的地在127.0.0.0内对127.0.0.100，数据包将发送对R3。
- 如果IP目的地在127.0.0.101内对127.0.0.200，数据包将发送对R6。

3) R1意识到有2个可能的ECMP路径和，因此需要发送2 ECHO请求与设置的TTL到2。从多种测验，注意到发起者用1个路径总是完成在去其次前。(但是这也许真实对一个特定实施)。

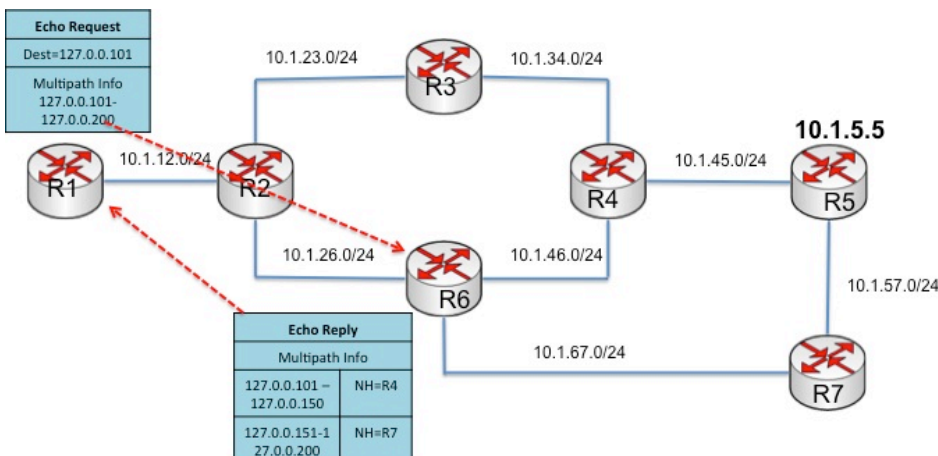


4) R1当前发送与下面的详细信息的ECHO请求：

- IP目的地作为127.0.0.0
- 运载地址范围的多重通道的信息TLV作为127.0.0.0对127.0.0.100。
- 顶部标签的TTL将设置到2。

5) (因为目的地址是127.0.0.0)， R2将转发数据包对R3。尽管只有一出口接口，在接收同样的R3将应答有同样多重通道的信息的上一步。

到达R5，同样适用。



6) 一旦PATH1 trace完成(在接收从出口口的回复以后), 发起者当前将查询PATH2。这通过发送ECHO请求执行与下面的详细信息:

- IP目的地作为127.0.0.101
- 运载地址范围的多重通道的信息TLV作为127.0.0.101到127.0.0.200
- 顶部标号组的TTL到2。

7) (因为目的地是127.0.0.101), R2将转发数据包对R6。在接收同样的R6将应答有多重通道的信息的上一步作为如下:

- 如果IP目的地在127.0.0.101内对127.0.0.150, 数据包将发送对R4。
- 如果IP目的地在127.0.0.151内对127.0.0.200, 数据包将发送对R7。

8) R1意识到有另外一个ECMP路径进行总可能的路径, 当3. R1继续通过发送与下面的详细信息的下个ECHO请求查询PATH2:

- IP目的地作为127.0.0.101
- 运载地址范围的多重通道的信息TLV作为127.0.0.101到127.0.0.150
- 顶部标号组TTL到3。

9) R2将转发数据包对R6 (因为目的地是127.0.0.101), 并且R6将转发它对R4 (因为目的地是127.0.0.101)。R4 doesn't有所有ECMP路径和, 因此应答有同样多重通道的信息的上一步。下一个信息包将到达出口R5。

10) 因为PATH2 trace完成, R1将继续PATH3的查询。这通过发送ECHO请求执行与下面的详细信息:

- IP目的地作为127.0.0.151
- 运载地址范围的多重通道的信息TLV作为127.0.0.151到127.0.0.200
- 顶部标号组TTL到3。

11) R2将转发数据包对R6, 反之将转发它对R7。R7将应答有同样多重通道的信息的TLV上一步。下一个信息包到达出口路由器R5。

在这些步骤完成后, R1在详细信息之下将有:

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

通过使用在127.0.0.0和127.0.0.100内的目的地地址, 当曾经从其他范围的地址将影响转发在各自路径时的数据包, 信息包转发在PATH1将影响。

12) 现在发起者将发送3 ECHO请求数据包与设置的TTL到255并且选择从每个范围的地址, 以便所有路径验证的端到端。

命令用于ECMP trace是 `traceroute mpls multipath ipv4 <prefix> <mask>`。以下输出示例:。

```
R1#traceroute 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
```

在输出上注意，显示有3个路径，并且所有路径优良工作。使用冗长的瘤上面命令将列出所有跳作为如下：

```
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
```