

Contenido

[Introducción](#)

[Antecedentes](#)

[Traza del árbol LSP - Cómo trabaja](#)

[Traza del árbol LSP - Ejemplo detallado](#)

[Discusiones relacionadas de la comunidad del soporte de Cisco](#)

Introducción

El ping MPLS LSP es una herramienta básica usada para validar la salud de la trayectoria conmutada de etiquetas (LSP) entre el ingreso y la salida. Este documento apunta explicar la interacción de la información de trayectoria múltiple entre el iniciador y el respondedor en la traza del árbol LSP. Para las opciones detalladas disponibles para esta herramienta, sería útil referir [este documento](#).

Antecedentes

¿Esta implementación del MPLS EM? La característica de trayectoria múltiple de la traza del árbol MPLS LSP se basa en el RFC 4379, *detectando los errores conmutados escritura de la etiqueta del avión de los datos del (MPLS) del multi-protocol*.

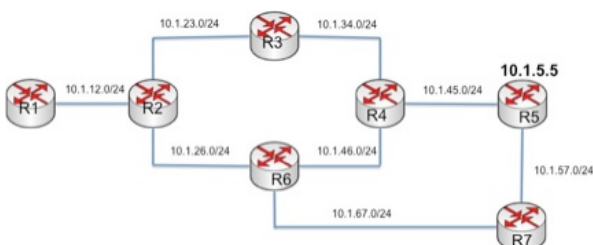
Fijando el IP Destination Address del paquete de sondeo como Loopback Address (127.x.x.x), la traza del árbol LSP se puede utilizar para detectar el error en el LSP evitando el paquete para conseguir el IP ruteado. Tan siempre que haya problema de conectividad de extremo a extremo, es útil utilizar el ping LSP como el primer paso para eliminar cualquier Falla del LSP.

En caso de los escenarios de trayecto múltiple, el ping LSP puede no ayudar siempre a identificar todas las Fallas del LSP. Pues puede ser que sea observado, cualquier Label Switch Router (LSR) en la recepción de un paquete etiquetado que pueda ser interfaces de egreso múltiples enviadas, utiliza ciertas claves del paquete y de la entrada al algoritmo de troceo para decidir la interfaz de egreso. Dependiendo del vendedor, del hardware, del etc, las opciones abajo unas de los se pueden considerar para desmenuzar:

1. Stack de la etiqueta entrante solo.
2. Stack de la etiqueta entrante y detalles del encabezado IP (si el payload es IP).
3. Stack de la etiqueta entrante, encabezado IP, y detalles de la encabezado del transporte.

Normalmente, los routers Cisco consideran una combinación de pila de etiquetas y de encabezado IP si el stack está de tamaño inferior o igual 3 (con el IP como el payload).

Assume después de la topología.



R1-R7 son Routers. En la topología antedicha, hay 3 rutas multi de la trayectoria del igual costo (ECMP) del r1 al R5 como abajo,

PATH1: R1-R2-R3-R4-R5

PATH2: R1-R2-R6-R4-R5

PATH3: R1-R2-R6-R7-R5

Asume que hay un problema entre el R6 y el R7 (como el Protocolo de distribución de etiquetas (LDP) quebrado o etiqueta el error de programación, el etc.) haciendo el tráfico del r1 al R5 vía PATH3 caer. Si el ping LSP del r1 toma PATH1 o PATH2, usted puede terminar para arriba si se asume que la trayectoria entre el r1 y el R5 está muy bien.

El ping LSP permite el fijar del IP Destination Address como a partir del rango el 127.0.0.0/8. Mientras que una opción simple es intentar manualmente enviar los paquetes ping múltiples con diversa dirección destino, no hay garantía que todas las trayectorias posibles ECMP serán validadas. Usted necesita una manera que pregunte y valide todos los trayectos posibles entre la fuente y el destino. ¿La traza de trayectoria múltiple del árbol LSP leverages? ¿Codificación de trayectoria múltiple de la información? definido en la sección 3.3.1 del RFC4379 y le ayuda a validar todas las trayectorias ECMP.

Traza del árbol LSP - Cómo trabaja

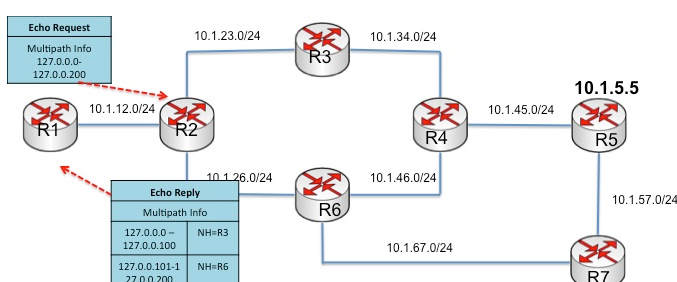
Un ping regular o el traceroute MPLS puede indicar que no hay incidente dependiendo de cómo la carga-parte de los routers de tránsito los paquetes sobre el ECMP, no obstante la traza del árbol LSP proporciona un mejor método para validar que todas las trayectorias están funcionando realmente.

En la traza del árbol LSP, el router del iniciador envía el pedido de eco MPLS a cada salto fijando TTL en la escritura de la etiqueta superior de una manera ampliada (a partir de 1). El pedido de eco llevará la información de trayectoria múltiple TLV que lleva un rango de la dirección IP (dentro del rango 127.0.0.0/8) o del rango de la escritura de la etiqueta de la entropía. Los dispositivos de Cisco soportan actualmente la opción del destino IP y así que nuestro ejemplo será detallado con el alcance del IP Address.

Cada uno transita el LSR en la recepción del paquete de pedidos contestará con todas las interfaces salientes ECMP y asociará un rango del IP Address (o de la escritura de la etiqueta de la entropía) del pedido cada interfaz.

Traza del árbol LSP - Ejemplo detallado

Assume después de la topología por ejemplo abajo.



Para la simplicidad, este ejemplo utiliza el intervalo de direcciones de 127.0.0.0-127.0.0.200. Aquí

están los detalles de los pasos en una traza del árbol LSP.

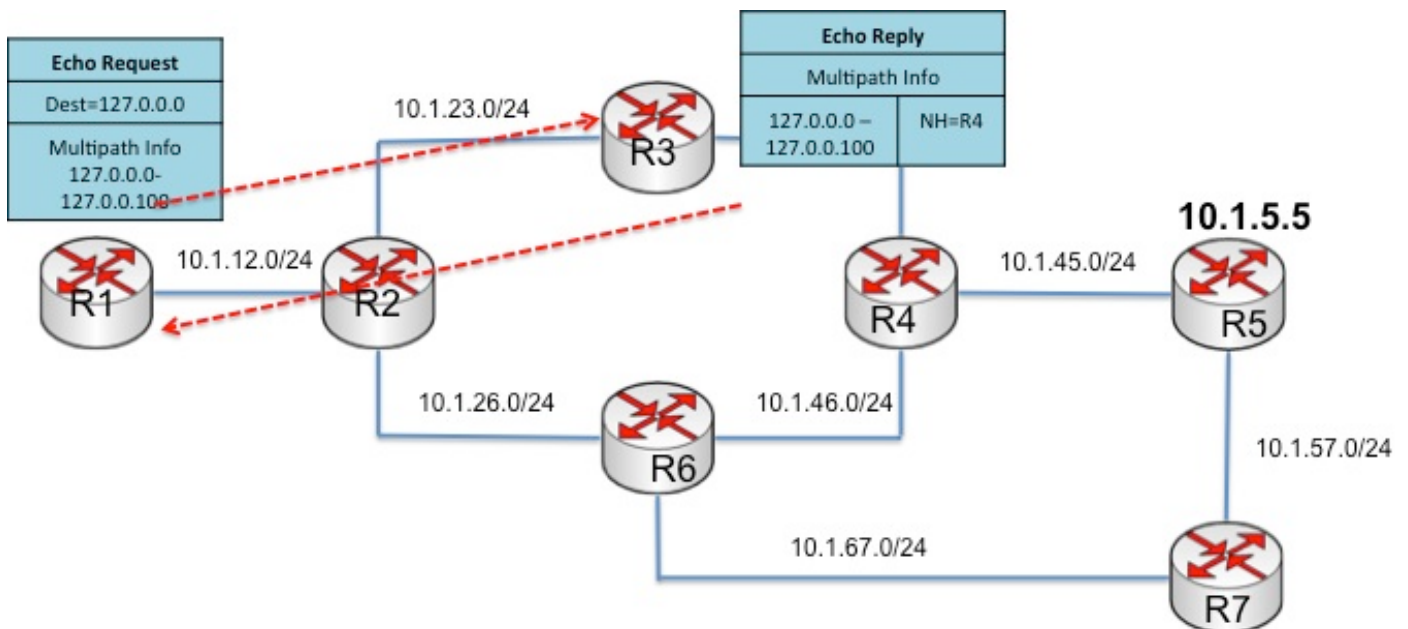
1) El iniciador (r1) envía el pedido de eco con los detalles abajo:

- Destino IP como 127.0.0.0
- Información de trayectoria múltiple TLV que lleva el intervalo de direcciones como 127.0.0.0 a 127.0.0.200.
- TTL de la escritura de la etiqueta superior será fijado a 1.

2) El r2 en la recepción lo mismo contestará detrás con la información de trayectoria múltiple para cada interfaz de egreso. En este ejemplo, contestará como abajo:

- Si el destino IP está dentro de 127.0.0.0 a 127.0.0.100, el paquete será enviado al R3.
- Si el destino IP está dentro de 127.0.0.101 a 127.0.0.200, el paquete será enviado al R6.

3) El r1 realiza que hay 2 trayectorias posibles ECMP y así que necesita enviar 2 que el pedido de eco con TTL fijó a 2. De las diversas pruebas, fue observado que el iniciador acaba siempre apagado con 1 trayectoria antes de ir a después. (Solamente esto pudo ser verdad para una implementación específica).

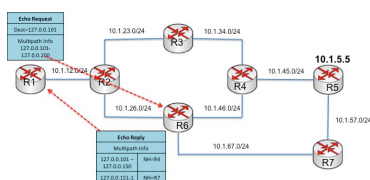


4) El r1 ahora envía el pedido de eco con los detalles abajo:

- Destino IP como 127.0.0.0
- Información de trayectoria múltiple TLV que lleva el intervalo de direcciones como 127.0.0.0 a 127.0.0.100.
- TTL de la escritura de la etiqueta superior será fijado a 2.

5) El r2 remitirá el paquete al R3 (pues la dirección destino es 127.0.0.0). El R3 en la recepción lo mismo contestará detrás con la misma información de trayectoria múltiple, pues hay solamente una interfaz de egreso.

Lo mismo es verdad hasta que alcanza el R5.



6) Una vez que la traza PATH1 es completa (después de recibir la contestación de la salida), el iniciador ahora preguntará PATH2. Esto es realizada enviando el pedido de eco con los detalles abajo:

- Destino IP como 127.0.0.101
- Información de trayectoria múltiple TLV que lleva el intervalo de direcciones como 127.0.0.101 a 127.0.0.200
- TTL del conjunto de escritura de la etiqueta superior a 2.

7) El r2 remitirá el paquete al R6 (pues la dirección destino es 127.0.0.101). El R6 en la recepción lo mismo contestará detrás con la información de trayectoria múltiple como abajo:

- Si el destino IP está dentro de 127.0.0.101 a 127.0.0.150, el paquete será enviado al R4.
- Si el destino IP está dentro de 127.0.0.151 a 127.0.0.200, el paquete será enviado al R7.

8) El r1 realiza que hay una más trayectoria ECMP que hace los trayectos posibles totales a medida que 3. r1 continúan preguntando PATH2 enviando el pedido de eco siguiente con los detalles abajo:

- Destino IP como 127.0.0.101
- Información de trayectoria múltiple TLV que lleva el intervalo de direcciones como 127.0.0.101 a 127.0.0.150
- TTL del conjunto de escritura de la etiqueta superior a 3.

9) El r2 remitirá el paquete al R6 (pues el destino es 127.0.0.101) y el R6 lo remitirá al R4 (pues el destino es 127.0.0.101). ¿Doesn R4? t tiene cualquier trayectoria ECMP y así que contestará detrás con la misma información de trayectoria múltiple. El próximo paquete alcanzará la salida R5.

10) Puesto que la traza PATH2 es completa, el r1 continuará la interrogación para PATH3. Esto es realizada enviando el pedido de eco con los detalles abajo:

- Destino IP como 127.0.0.151
- Información de trayectoria múltiple TLV que lleva el intervalo de direcciones como 127.0.0.151 a 127.0.0.200
- TTL del conjunto de escritura de la etiqueta superior a 3.

11) El r2 remitirá el paquete al R6, que a su vez lo remitirá al R7. El R7 contestará detrás con la misma información de trayectoria múltiple TLV. El próximo paquete alcanza al router Egress R5.

Después de que estos pasos sean completos, el r1 tendrá debajo de los detalles:

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

Usando la dirección destino dentro de 127.0.0.0 y de 127.0.0.100, el reenvío de paquete será

influenciado sobre PATH1 mientras que usar el direccionamiento de otros rangos influenciará el envío del paquete sobre las trayectorias respectivas.

12) Ahora el iniciador enviará 3 paquetes de pedido de eco con TTL fijó a 255 y selecciona el direccionamiento de cada rango de modo que todas las trayectorias sean de punta a punta validado.

El comando de ser utilizado para la traza ECMP es el *<mask> de trayectoria múltiple ipv4 de los mpls del traceroute <prefix>*. Lo que sigue es una salida de muestra.

```
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
```

Observe que sobre la salida muestra que hay 3 trayectorias y todas las trayectorias están funcionando muy bien. Usando el botón prolijo adentro sobre el comando enumerará todos los saltos como abajo:

```
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
```