

Contenu

[Introduction](#)

[Informations générales](#)

[Suivi d'arborescence LSP - Comment cela fonctionne](#)

[Suivi d'arborescence LSP - Exemple détaillé](#)

[Cisco relatif prennent en charge des discussions de la Communauté](#)

Introduction

Le ping MPLS LSP est un outil de base utilisé pour valider les santés du chemin commuté par étiquette (LSP) entre le d'entrée et le de sortie. Ce document vise à expliquer l'interaction des informations multivoies entre le demandeur et le responder dans le suivi d'arborescence LSP. Pour des options détaillées disponibles pour cet outil, il serait utile de se référer [ce document](#).

Informations générales

Cette implémentation de l'EM MPLS ? La caractéristique multivoie de suivi d'arborescence MPLS LSP est basée sur RFC 4379, *déteçant des pannes du plan de données (MPLS) commutées par étiquette multiprotocole*.

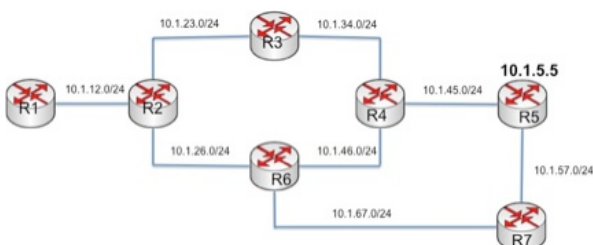
En plaçant l'adresse de destination IP du paquet de sonde comme adresse de bouclage (127.x.x.x), le suivi d'arborescence LSP peut être utilisé pour détecter la panne dans le LSP en évitant le paquet obtiennent l'IP conduit. Ainsi toutes les fois qu'il y a problème de connectivité de bout en bout, il est utile d'employer le ping LSP comme première étape pour éliminer n'importe quelle panne LSP.

En cas de scénarios multivoies, le ping LSP peut toujours ne pas aider à identifier toutes les pannes LSP. Car il pourrait noter, en étiquettent le routeur de commutateur (LSR) sur recevoir un paquet étiqueté qui peut être de plusieurs interfaces de sortie envoyées, emploient certaines clés du paquet et de l'entrée à l'algorithme de hachage pour décider l'interface de sortie. Selon le constructeur, le matériel, etc., des options ci-dessous l'un des peuvent être considérées pour le hachage :

1. Pile d'étiquette entrant seule.
2. Pile d'étiquette entrant et détails d'en-tête IP (si la charge utile est IP).
3. Pile d'étiquette entrant, en-tête IP, et détails d'en-tête de transport.

Normalement, les Routeurs de Cisco considèrent une combinaison de pile d'étiquette et d'en-tête IP si la pile est de taille inférieur ou égal à 3 (avec l'IP comme charge utile).

Assume après topologie.



R1-R7 sont des Routeurs. Dans la topologie ci-dessus, il y a 3 artères multi du chemin de coût égal (ECMP) de R1 à R5 en tant que ci-dessous,

PATH1 : R1-R2-R3-R4-R5

PATH2 : R1-R2-R6-R4-R5

PATH3 : R1-R2-R6-R7-R5

Supposez qu'il y a une question entre R6 et R7 (comme l'erreur de programmation cassée de protocole de distribution d'étiquette (LDP) ou d'étiquette, etc.) faisant chuter le trafic de R1 à R5 par l'intermédiaire de PATH3. Si le ping LSP de R1 prend PATH1 ou PATH2, vous pouvez finir par supposant que le chemin entre R1 et R5 est bien.

Le ping LSP permet placer l'adresse de destination IP en tant que n'importe quelle de la plage 127.0.0.0/8. Tandis qu'une option simple est d'essayer manuellement envoyer de plusieurs paquets de ping avec l'adresse de destination différente, il n'y a aucune garantie que tous les chemins possibles ECMP seront validés. Vous avez besoin d'une manière dont questionne et valide tous les chemins possibles entre la source et la destination. Le suivi multivoie d'arborescence LSP accroît ? Codage multivoie de l'information ? défini dans la section 3.3.1 de RFC4379 et vous aide à valider tous les chemins ECMP.

Suivi d'arborescence LSP - Comment cela fonctionne

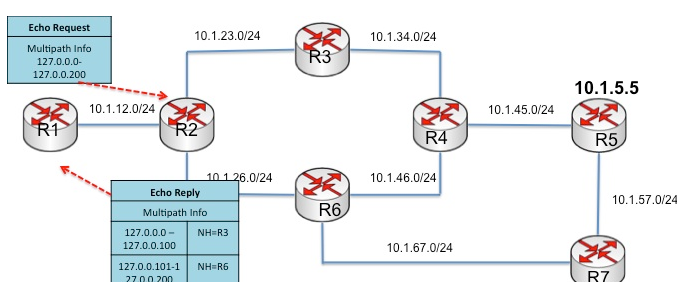
Un ping ou une traceroute du militaire de carrière MPLS peut indiquer qu'il n'y a aucune panne selon la façon dont le load-share de Routeurs de transit les paquets au-dessus d'ECMP, toutefois le suivi d'arborescence LSP fournit une meilleure méthode pour valider que tous les chemins fonctionnent réellement.

Dans le suivi d'arborescence LSP, le routeur de demandeur envoie la requête d'écho MPLS à chaque saut en plaçant le TTL dans l'étiquette supérieure d'une manière incrémentale (à partir de 1). La requête d'écho portera la TLV multivoie de l'information qui porte une plage d'adresse IP (à moins de marge 127.0.0.0/8) ou la plage d'étiquette d'entropie. Actuellement les périphériques de Cisco prennent en charge l'option de destination IP et ainsi notre exemple sera détaillé avec la plage d'adresses IP.

Chaque transit LSR sur recevoir le paquet de demandes répondra avec toutes les interfaces sortantes ECMP et associera une plage d'adresse IP (ou l'étiquette d'entropie) de la demande de chaque interface.

Suivi d'arborescence LSP - Exemple détaillé

Assume après topologie par exemple ci-dessous.



Pour la simplicité, cet exemple utilise la plage d'adresses de 127.0.0.0-127.0.0.200. Voici les

détails des étapes dans un suivi d'arborescence LSP.

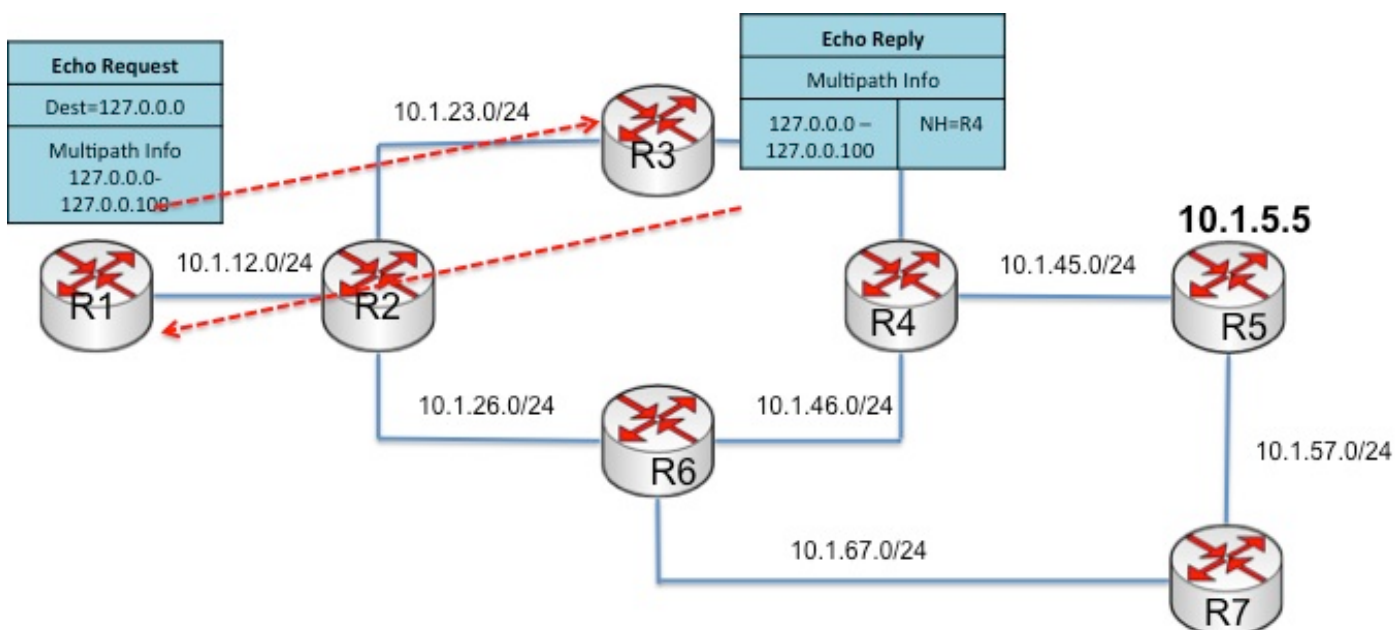
1) Le demandeur (R1) envoie la requête d'écho avec les détails ci-dessous :

- Destination IP en tant que 127.0.0.0
- TLV multivoie de l'information portant la plage d'adresses comme 127.0.0.0 à 127.0.0.200.
- Le TTL de l'étiquette supérieure sera placé à 1.

2) R2 sur recevoir la même chose répondra de retour avec les informations multivoies pour chaque interface de sortie. Dans cet exemple, il répondra en tant que ci-dessous :

- Si la destination IP est dans 127.0.0.0 à 127.0.0.100, le paquet sera envoyé à R3.
- Si la destination IP est dans 127.0.0.101 à 127.0.0.200, le paquet sera envoyé à R6.

3) R1 se rend compte qu'il y a 2 chemins possibles ECMP et ainsi il doit envoyer la requête d'écho 2 avec le TTL réglé à 2. De divers tests, on l'a observé que le demandeur finit toujours avec 1 chemin avant d'aller à ensuite. (Mais ceci pourrait être vrai pour une implémentation spécifique).

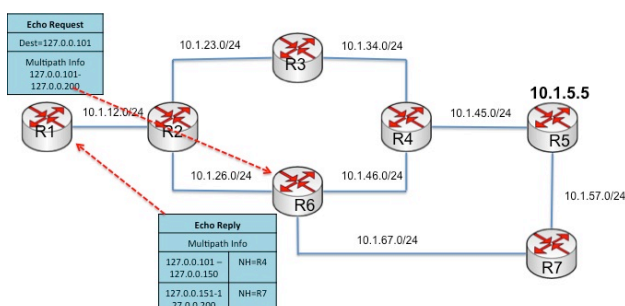


4) R1 envoie maintenant la requête d'écho avec les détails ci-dessous :

- Destination IP en tant que 127.0.0.0
- TLV multivoie de l'information portant la plage d'adresses comme 127.0.0.0 à 127.0.0.100.
- Le TTL de l'étiquette supérieure sera placé à 2.

5) R2 expédiera le paquet à R3 (car l'adresse de destination est 127.0.0.0). R3 sur recevoir la même chose répondra de retour avec les mêmes informations multivoies, car il y a seulement une interface de sortie.

Le même juge vrai jusqu'à ce qu'il atteigne R5.



6) Une fois que le suivi PATH1 est complet (après réception de la réponse du de sortie), le demandeur questionnera maintenant PATH2. Ceci est exécuté en envoyant la requête d'écho avec les détails ci-dessous :

- Destination IP en tant que 127.0.0.101
- TLV multivoie de l'information portant la plage d'adresses comme 127.0.0.101 à 127.0.0.200
- Le TTL du positionnement d'étiquette supérieur à 2.

7) R2 expédiera le paquet à R6 (car l'adresse de destination est 127.0.0.101). R6 sur recevoir la même chose répondra de retour avec les informations multivoies en tant que ci-dessous :

- Si la destination IP est dans 127.0.0.101 à 127.0.0.150, le paquet sera envoyé à R4.
- Si la destination IP est dans 127.0.0.151 à 127.0.0.200, le paquet sera envoyé à R7.

8) R1 se rend compte qu'il y a un plus de chemin ECMP faisant tous les chemins possibles car 3. R1 continué à questionner PATH2 en envoyant la prochaine requête d'écho avec les détails ci-dessous :

- Destination IP en tant que 127.0.0.101
- TLV multivoie de l'information portant la plage d'adresses comme 127.0.0.101 à 127.0.0.150
- Le TTL du positionnement d'étiquette supérieur à 3.

9) R2 expédiera le paquet à R6 (car la destination est 127.0.0.101) et R6 l'expédiera à R4 (car la destination est 127.0.0.101). Doesn R4 ? t ont n'importe quel chemin ECMP et ainsi répondront de retour avec les mêmes informations multivoies. Le paquet suivant atteindra le de sortie R5.

10) Puisque le suivi PATH2 est complet, R1 continuera la requête pour PATH3. Ceci est exécuté en envoyant la requête d'écho avec les détails ci-dessous :

- Destination IP en tant que 127.0.0.151
- TLV multivoie de l'information portant la plage d'adresses comme 127.0.0.151 à 127.0.0.200
- Le TTL du positionnement d'étiquette supérieur à 3.

11) R2 expédiera le paquet à R6, qui à leur tour l'expédiera à R7. R7 répondra de retour avec la même TLV multivoie de l'information. Le paquet suivant atteint le routeur R5 de sortie.

Après que ces étapes soient complètes, R1 aura au-dessous des détails :

| 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 |

À l'aide de l'adresse de destination dans 127.0.0.0 et 127.0.0.100, le transfert de paquet sera influencé au-dessus de PATH1 tandis que l'utilisation de l'adresse d'autres plages influencera expédier le paquet au-dessus des chemins respectifs.

12) Maintenant le demandeur enverra 3 paquets de demande d'écho avec le positionnement TTL

à 255 et sélectionnera l'adresse de chaque plage de sorte que tous les chemins soient de bout en bout validé.

La commande d'être utilisé pour le suivi ECMP est `<mask> de <prefix> d'ipv4 de traceroute mpls multipath`. Être suit un résultat témoin.

```
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
```

Notez qu'au-dessus de la sortie prouve qu'il y a 3 chemins et tous les chemins fonctionnent bien. Utilisant la molette bavarde dedans au-dessus de la commande répertoriera tous les sauts en tant que ci-dessous :

```
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
```