

Fonction mLRE sur le routeur IOS-XE

Contenu

[Introduction](#)

[Problème](#)

[Solution](#)

Introduction

Ce document décrit la fonction mLRE (Multicast Leaf Recycle Elimination) sur le routeur IOS-XE.

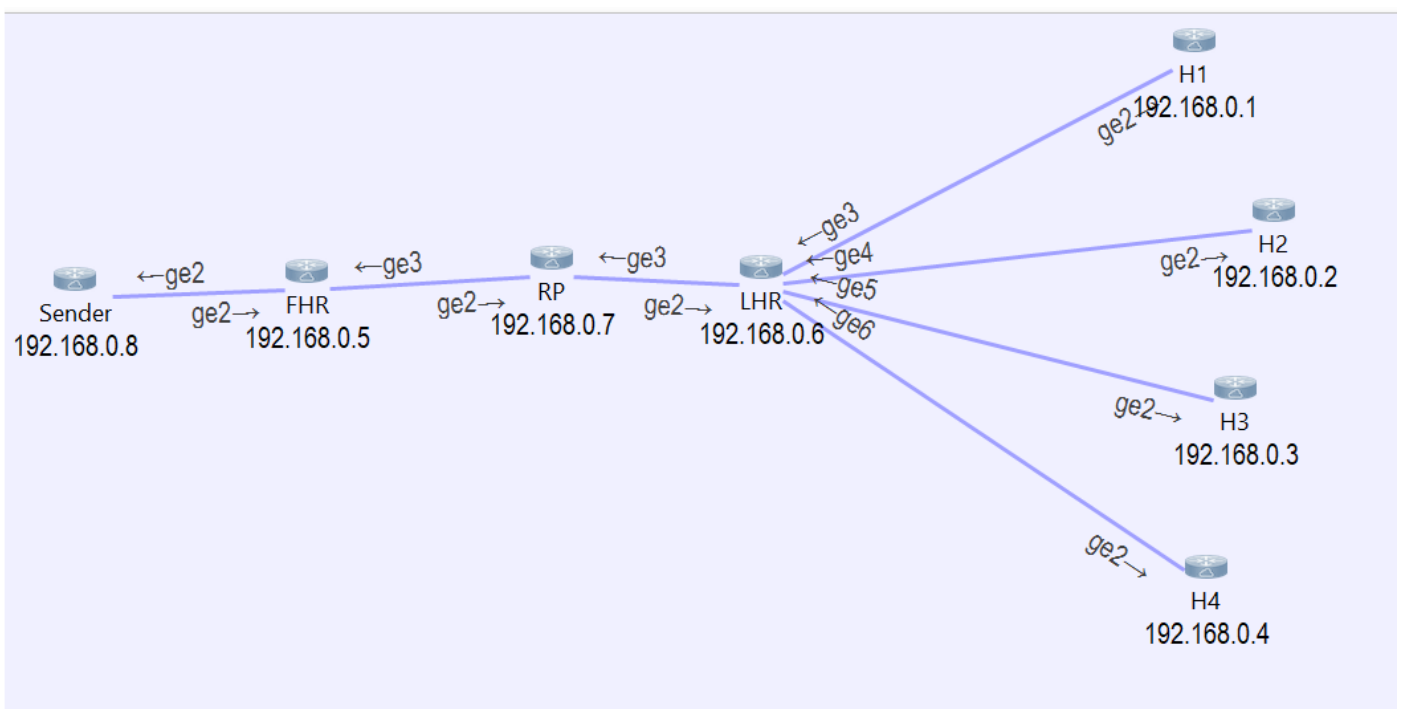
Problème

Si les hôtes sont connectés sur plusieurs interfaces et demandent le trafic de multidiffusion sur un routeur. Le routeur doit faire une copie du trafic de multidiffusion et doit l'envoyer sur toutes les interfaces qui demandent le trafic de multidiffusion de ce groupe de multidiffusion particulier. Si les paquets sont traités en série, c'est-à-dire un paquet après l'autre, cela aide le routeur à améliorer les performances. Cependant, elle crée un retard injuste dans différents noeuds en raison de sa nature. Ce traitement sériel du trafic de multidiffusion est appelé LRE dans les routeurs de multidiffusion et est par défaut activé sur les routeurs qui dépendent de leur version et de leur modèle IOS.

Même si le traitement des paquets provoque en série une différence de 4 à 12 microsecondes entre les paquets traités en contiguïté.

Il peut provoquer un retard important dans les environnements critiques, comme Trading Company, s'il existe un grand nombre de noeuds Leaf qui demandent le trafic de multidiffusion.

Cette image montre la topologie pour mieux comprendre cela.



Comme vous pouvez le voir, nous avons 4 hôtes connectés au LHR et ils demandent du trafic pour le groupe de multidiffusion 239.1.1.1.

Si Packet Tracer s'exécute sur LHR, il s'avère que le paquet reçu de la source est consommé en silence par LHR, puis il a créé 4 paquets similaires et les a transférés à chaque interface connectée à l'hôte.

```
LHR#show platform packet-trace summary
```

Pkt	Input	Output	State	Reason	
0	Gi2	<none>	CONS	Packet Consumed Silently	<<< recieved
packet from sender					
1	Gi2	Gi6	FWD		<<< first
replicated packet sent to int gig6					
2	Gi2	Gi5	FWD		<<< first
replicated packet sent to int gig5					
3	Gi2	Gi4	FWD		<<< first
replicated packet sent to int gig4					
4	Gi2	Gi3	FWD		<<< first
replicated packet sent to int gig3					

Si les détails des paquets capturés sont ouverts, vous pouvez voir l'heure de début et l'heure de fin de chaque paquet.

```
LHR#show platform packet-trace packet 0
```

```
Packet: 0          CBUG ID: 85
Summary
  Input       : GigabitEthernet2
  Output      : <none>
  State       : CONS Packet Consumed Silently
  Timestamp
    Start     : 37067929596524 ns (05/27/2020 02:43:56.203649 UTC)
    Stop      : 37067929669545 ns (05/27/2020 02:43:56.203722 UTC)
```

```
LHR#show platform packet-trace packet 1
```

```
Packet: 1          CBUG ID: 85
Summary
  Input       : GigabitEthernet2
  Output      : GigabitEthernet6
  State       : FWD
  Timestamp
    Start     : 37067929722925 ns (05/27/2020 02:43:56.203776 UTC)
    Stop      : 37067929750941 ns (05/27/2020 02:43:56.203804 UTC)
```

```
LHR#show platform packet-trace packet 2
```

```
Packet: 2          CBUG ID: 85
Summary
  Input       : GigabitEthernet2
  Output      : GigabitEthernet5
  State       : FWD
  Timestamp
    Start     : 37067929752437 ns (05/27/2020 02:43:56.203805 UTC)
    Stop      : 37067929759667 ns (05/27/2020 02:43:56.203812 UTC)
```

```
LHR#show platform packet-trace packet 3
```

```
Packet: 3          CBUG ID: 85
Summary
  Input       : GigabitEthernet2
  Output      : GigabitEthernet4
  State       : FWD
  Timestamp
```

```
Start   : 37067929760929 ns (05/27/2020 02:43:56.203814 UTC)
Stop    : 37067929766997 ns (05/27/2020 02:43:56.203820 UTC)
```

```
LHR#show platform packet-trace packet 4
```

```
Packet: 4          CBUG ID: 85
```

```
Summary
```

```
Input       : GigabitEthernet2
```

```
Output      : GigabitEthernet3
```

```
State       : FWD
```

```
Timestamp
```

```
Start       : 37067929768236 ns (05/27/2020 02:43:56.203821 UTC)
```

```
Stop        : 37067929774283 ns (05/27/2020 02:43:56.203827 UTC)
```

Si les heures de début et de fin des sorties précédemment mentionnées sont comparées, il est entendu que le traitement des paquets est en cours de traitement en série.

```
Start       : 37067929722925 ns (05/27/2020 02:43:56.203776 UTC) << packet1
Stop        : 37067929750941 ns (05/27/2020 02:43:56.203804 UTC)
Start       : 37067929752437 ns (05/27/2020 02:43:56.203805 UTC) << packet 2
Stop        : 37067929759667 ns (05/27/2020 02:43:56.203812 UTC)
Start       : 37067929760929 ns (05/27/2020 02:43:56.203814 UTC) << packet 3
Stop        : 37067929766997 ns (05/27/2020 02:43:56.203820 UTC)
Start       : 37067929768236 ns (05/27/2020 02:43:56.203821 UTC) << packet 4
Stop        : 37067929774283 ns (05/27/2020 02:43:56.203827 UTC)
```

Si le temps d'arrêt du paquet 1 (02:43:56.203804) et du paquet 4 (02:43:56.203821) est comparé, vous pouvez voir qu'il y a une différence de temps de 17 microsecondes.

Dans certaines organisations critiques, ce délai n'est peut-être pas acceptable et doit donc être réduit.

Solution

Afin d'éviter ce délai, désactivez la fonction LRE sur le routeur.

Si la fonctionnalité LRE est désactivée, le traitement des paquets pour les différentes interfaces pour la réplication du trafic de multidiffusion est indépendant l'un de l'autre et est traité de manière parallèle.

Afin de désactiver la fonctionnalité mLRE, utilisez cette commande : **outer(config)# platform multicast lre off**