

mLRE-Funktion auf dem IOS-XE-Router

Inhalt

[Einführung](#)

[Problem](#)

[Lösung](#)

Einführung

In diesem Dokument wird die mLRE-Funktion (Multicast Leaf Recycle Elimination) des IOS-XE-Routers beschrieben.

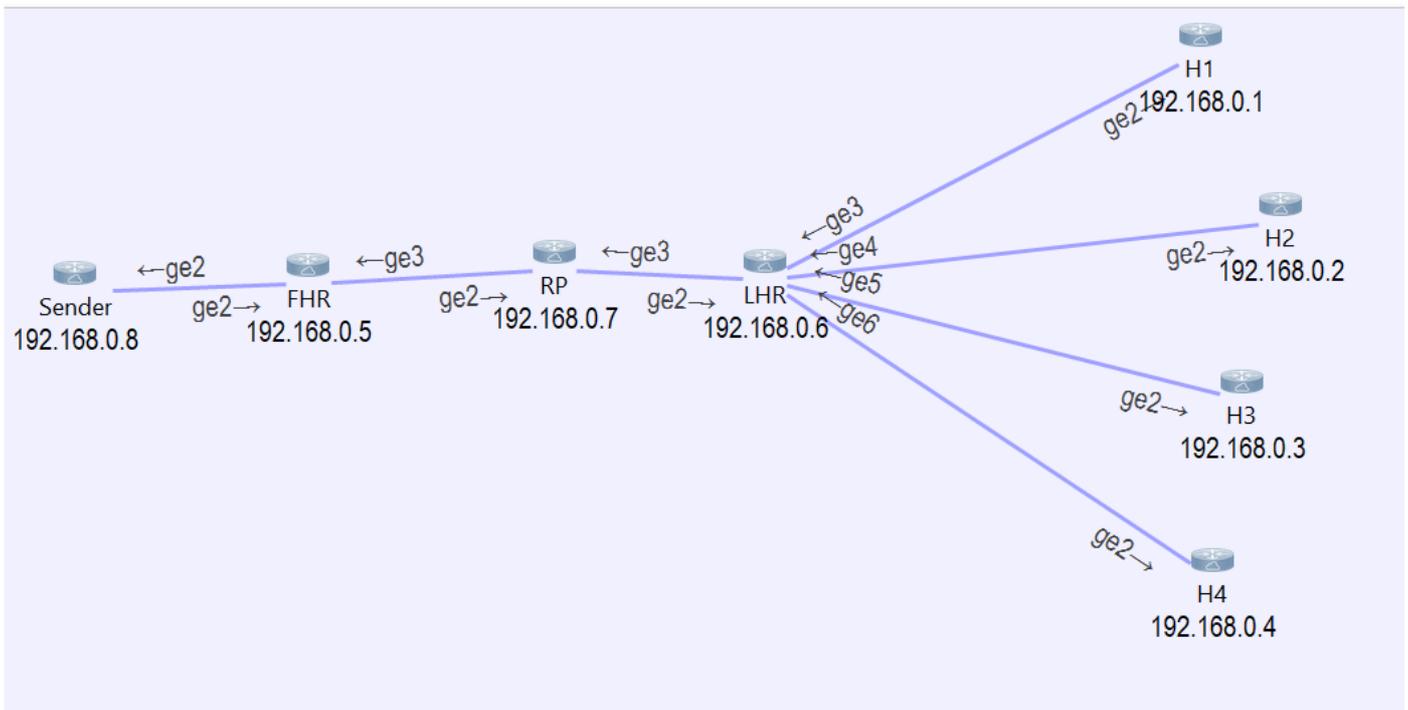
Problem

Wenn Hosts an mehreren Schnittstellen verbunden sind und Multicast-Datenverkehr auf einem Router anfordern. Der Router muss eine Kopie des Multicast-Datenverkehrs erstellen und diese an alle Schnittstellen senden, die den Multicast-Datenverkehr dieser bestimmten Multicast-Gruppe anfordern. Wenn die Pakete seriell verarbeitet werden, d. h. ein Paket nach dem anderen, hilft es dem Router, die Leistung zu verbessern. Sie führt jedoch aufgrund ihrer Natur zu einer unfairen Verzögerung in verschiedenen Knoten. Diese serielle Verarbeitung des Multicast-Datenverkehrs wird in den Multicast-Routern als LRE bezeichnet und ist standardmäßig auf Routern aktiviert, die von ihrer IOS-Version und -Modell abhängig sind.

Obwohl die Verarbeitung von Paketen seriell einen Unterschied von 4-12 Mikrosekunden zwischen den einander ergänzend verarbeiteten Paketen verursacht.

Wenn eine große Anzahl von Leaf-Knoten Multicast-Datenverkehr anfordern, kann dies zu erheblichen Verzögerungen in zeitkritischen Umgebungen wie dem Trading-Unternehmen führen.

Dieses Bild zeigt die Topologie, um dies besser zu verstehen.



Wie Sie sehen können, sind vier Hosts mit dem LHR verbunden, die Datenverkehr für die Multicast-Gruppe 239.1.1.1 anfordern.

Wenn der Paket-Tracker auf LHR ausgeführt wird, wird deutlich, dass das von der Quelle empfangene Paket von LHR unbemerkt verbraucht wird. Anschließend werden vier ähnliche Pakete erstellt und an jede Schnittstelle weitergeleitet, die mit dem Host verbunden ist.

```
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					

Wenn die Details der erfassten Pakete geöffnet werden, können Sie die Start- und Endzeit für jedes Paket sehen.

```
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)
```

Wenn Start- und Endzeit der zuvor erwähnten Ausgaben verglichen werden, wird davon ausgegangen, dass die Paketverarbeitung seriell läuft.

```
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)
```

Wenn die Stoppzeit von Paket 1 (02:43:56.203804) und Paket 4 (02:43:56.203821) verglichen wird, sehen Sie, dass es einen Zeitunterschied von 17 Mikrosekunden gibt.

In zeitkritischen Unternehmen ist diese Verzögerung möglicherweise nicht akzeptabel und muss daher reduziert werden.

Lösung

Um diese Zeitverzögerung zu vermeiden, deaktivieren Sie die LRE-Funktion auf dem Router.

Wenn die LRE-Funktion deaktiviert ist, ist die Paketverarbeitung für die unterschiedliche Schnittstelle zur Replikation von Multicast-Datenverkehr voneinander unabhängig und wird parallel

verarbeitet.

Um die mLRE-Funktion zu deaktivieren, verwenden Sie den folgenden Befehl: **outer(config)#
plattformmulticast wird später deaktiviert**