

mLRE Feature on the IOS-XE Router

Contents

[Introduction](#)

[Problem](#)

[Solution](#)

Introduction

This document describes the Multicast Leaf Recycle Elimination (mLRE) feature on the IOS-XE router.

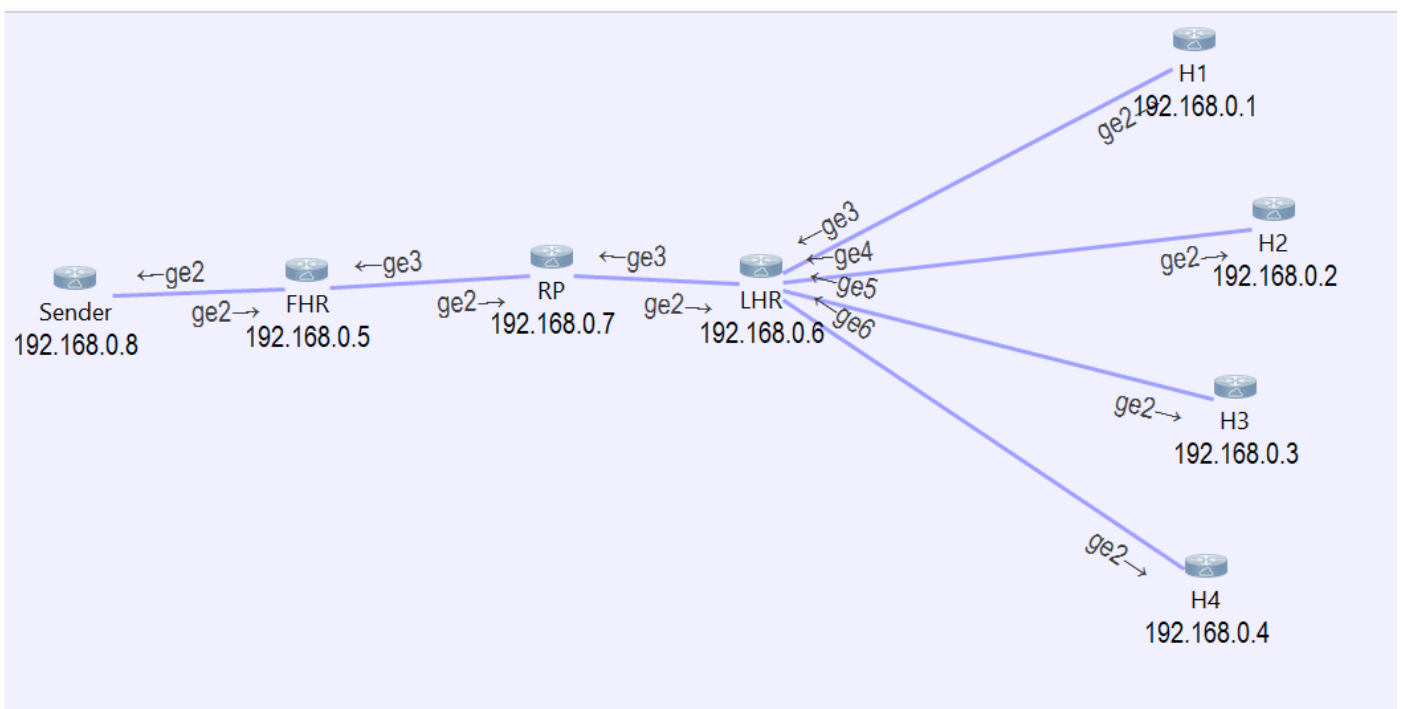
Problem

If hosts are connected on multiple interfaces and request multicast traffic on a router. The router has to make a copy of multicast traffic and has to send it on all the interfaces which request the multicast traffic of that particular multicast group. If the packets are processed serially, which means one packet after another, then it helps the router to improve performance. However, it creates an unfair delay in different nodes due to its nature. This serial processing of multicast traffic is called as LRE in the multicast routers and is by default enable on routers that depend upon their IOS version and model.

Even though processing packets serially causes a difference of 4-12 microseconds between to adjacently processed packets.

It could create a significant delay in time-critical environments like Trading company if there are a large number of leaf nodes that request multicast traffic.

This image shows the topology to understand this better.



As you can see that we have 4 hosts connected to the LHR and they are requesting traffic for multicast group 239.1.1.1.

If packet tracer runs on LHR then it is seen that the packet received from the source is silently consumed by LHR and then it created 4 similar packets and forwarded them to each interface connected to the host.

```
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					

If the details of captured packets are opened, you can see the start time and end time for each packet.

```
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)
```

If the start and end time of previously mentioned outputs are compared, it is understood that the packet processing is going on serially.

```
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)
```

If the stop time of packet 1 (02:43:56.203804) and packet 4 (02:43:56.203821) are compared, you can see that there is a time difference of 17 micro-seconds.

In some time-critical organizations this delay may not be acceptable and hence needs to be reduced.

Solution

In order to avoid this time delay, disable the LRE feature on the router.

If the LRE feature is disabled, the packet processing for the different interface for the replication of multicast traffic is independent of each other and are processed parallelly.

In order to disable the mLRE feature, use this command: **outer(config)# platform multicast lre off**