

# 了解PIM断言机制

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[什么是PIM断言机制？](#)

[场景1. LHR原理](#)

[摘自RFC 7761第4.2.2节。](#)

[场景2.断言路径选择](#)

[摘自RFC 7761第4.6.3节。](#)

[摘要](#)

## 简介

本文档介绍协议独立组播(PIM)断言机制，重点介绍PIM断言获胜者标准，并深入探讨某些角落情况。

## 先决条件

### 要求

思科建议您了解PIM断言机制。

### 使用的组件

本文档中的信息基于Cisco CSR1000V版本16.4.1

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

## 什么是PIM断言机制？

当共享网段上有多个启用PIM的路由器时，这些路由器可能会遇到重复的组播流量。这种情况可能是因为同一共享网段上的两台或多台路由器可能具有有效(S, G)或(\*,G)条目，该条目会填充指向同一源IP/目标组的共享网段的传出接口。

PIM断言机制用于检测和消除共享网段上组播流量的重复。请注意，此机制不会阻止重复发生，而是使用组播流量的重复作为触发器来激活此机制，该机制为此数据流选择单个转发器。

当共享网段上有重复的组播流量时，可以假设有多台路由器在共享网段上发送相同(S, G)或(\*,G)。如果您选择一台路由器来有效转发此流，则可消除重复。

PIM利用PIM断言消息，当您在传出接口列表(OIL)上收到组播数据包时，会触发这些消息。这些断

言消息包含度量，然后这些度量用于计算谁将成为断言赢家。LAN上的下游路由器也会收到PIM断言消息。然后，下游设备使用这些消息将适当的加入/修剪消息发送到赢得断言选择的的上游路由器。

## 场景1. LHR原理

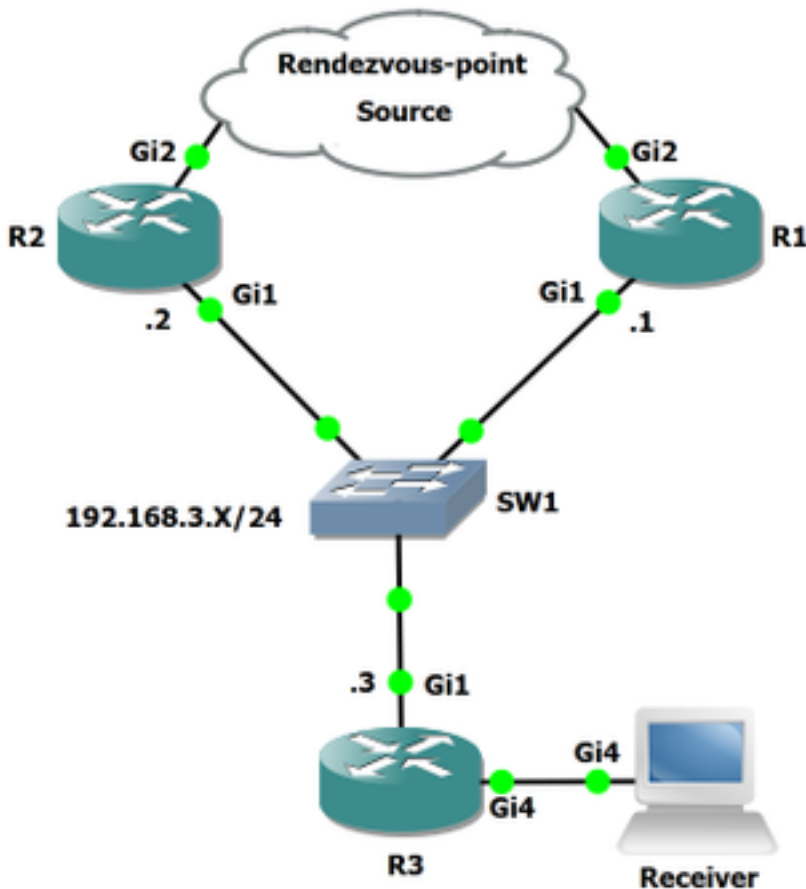


图 1.

在网络图中，R3是最后一跳路由器(LHR),R3通过共享网段连接到R2和R1。

当您从接收方收到互联网组管理协议(IGMP)报告时，R3会检查RPF邻居是谁。在拓扑中，R1是指向RP的RPF邻居，因此R3向R1发送(\*,G)连接。一旦R1下拉流（假设组处于活动状态），R3向源发送(S, G)连接并下拉源树。R2是指向源树的RPF邻居，这意味着R3将向R2发送(S, G)连接。R3的RPF接口与RP和源相同。在此，您可以看到组239.1.1.1的R3mroute表。

```
R3#show ip mroute
IP Multicast Routing Table
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 239.1.1.1), 00:00:55/stopped, RP 192.168.0.100, flags: SJC
  Incoming interface: GigabitEthernet1, RPF nbr 192.168.3.1
  Outgoing interface list:
    GigabitEthernet4, Forward/Sparse, 00:00:55/00:02:04

(10.0.0.2, 239.1.1.1), 00:00:52/00:02:07, flags: JT
  Incoming interface: GigabitEthernet1, RPF nbr 192.168.3.2, Mroute
  Outgoing interface list:
    GigabitEthernet4, Forward/Sparse, 00:00:52/00:02:07
```

```
(* , 224.0.1.40), 00:01:22/00:02:09, RP 192.168.0.100, flags: SJPCCL  
Incoming interface: GigabitEthernet1, RPF nbr 192.168.3.1
```

如您在R3上看到的，(\*,G)RPF邻居为192.168.3.1，而指向(S,G)的RPF邻居为192.168.3.2。现在，这应导致R1和R2都具有指向R3的有效OIL。让我们看一下这些条目：

R1#**show ip mroute**

```
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join  
Timers: Uptime/Expires  
Interface state: Interface, Next-Hop or VCD, State/Mode
```

```
(* , 239.1.1.1), 00:15:02/00:02:33, RP 192.168.0.100, flags: S  
Incoming interface: GigabitEthernet2, RPF nbr 192.168.5.2  
Outgoing interface list:  
GigabitEthernet1, Forward/Sparse, 00:15:02/00:02:33
```

```
(10.0.0.2, 239.1.1.1), 00:13:24/00:02:33, flags: PR  
Incoming interface: GigabitEthernet2, RPF nbr 192.168.5.2  
Outgoing interface list: Null
```

```
(* , 224.0.1.40), 00:29:17/00:02:51, RP 192.168.0.100, flags: SJCL  
Incoming interface: GigabitEthernet2, RPF nbr 192.168.5.2  
Outgoing interface list:  
GigabitEthernet1, Forward/Sparse, 00:16:06/00:02:51  
Outgoing interface list: Null
```

R2#**show ip mroute**

```
IP Multicast Routing Table  
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join  
Timers: Uptime/Expires  
Interface state: Interface, Next-Hop or VCD, State/Mode
```

```
(* , 239.1.1.1), 00:08:00/stopped, RP 192.168.0.100, flags: SP  
Incoming interface: GigabitEthernet2, RPF nbr 192.168.4.1  
Outgoing interface list: Null
```

```
(10.0.0.2, 239.1.1.1), 00:00:03/00:02:56, flags: T  
Incoming interface: GigabitEthernet2, RPF nbr 192.168.4.1  
Outgoing interface list:  
GigabitEthernet1, Forward/Sparse, 00:00:03/00:03:26
```

```
(* , 224.0.1.40), 01:37:30/00:02:22, RP 192.168.0.100, flags: SJPL  
Incoming interface: GigabitEthernet2, RPF nbr 192.168.4.1
```

如前所述，当共享网段上填充有有效OIL的两个上游路由器时，可触发断言。由于R1和R2都具有有效的OIL，请检查数据包捕获中是否存在断言机制。

此数据包捕获是在R3 Gi1接口上捕获到SW1的。

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.3.1	224.0.0.5	OSPF	98	Hello Packet
2	0.705389	192.168.3.2	224.0.0.5	OSPF	98	Hello Packet
3	3.124776	192.168.3.3	224.0.0.5	OSPF	98	Hello Packet
4	7.733948	192.168.3.2	224.0.0.13	PIMv2	72	Hello
5	9.480827	192.168.3.1	224.0.0.5	OSPF	98	Hello Packet
6	10.256987	192.168.3.2	224.0.0.5	OSPF	98	Hello Packet
7	11.954130	192.168.3.1	224.0.0.13	PIMv2	72	Hello
8	12.621371	192.168.3.3	224.0.0.13	PIMv2	72	Hello
9	13.015136	192.168.3.3	224.0.0.5	OSPF	98	Hello Packet
10	19.046520	192.168.3.1	224.0.0.5	OSPF	98	Hello Packet
11	19.670571	192.168.3.2	224.0.0.5	OSPF	98	Hello Packet
12	22.114741	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=0/0, ttl=253 (multicast)
13	22.137371	192.168.3.3	224.0.0.13	PIMv2	68	Join/Prune
14	22.137597	192.168.3.3	224.0.0.13	PIMv2	68	Join/Prune
15	22.972394	192.168.3.3	224.0.0.5	OSPF	98	Hello Packet
16	23.085520	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=1/256, ttl=253 (multicast)
17	24.087827	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=2/512, ttl=253 (multicast)
18	24.723777	192.168.3.3	224.0.0.13	PIMv2	96	Join/Prune
19	25.088340	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=3/768, ttl=253 (multicast)
20	26.091246	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=4/1024, ttl=253 (multicast)
21	27.091219	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=5/1280, ttl=253 (multicast)
22	28.109058	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=6/1536, ttl=253 (multicast)
23	29.000065	192.168.3.1	224.0.0.5	OSPF	98	Hello Packet
24	29.118436	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=7/1792, ttl=253 (multicast)
25	29.225379	192.168.3.2	224.0.0.5	OSPF	98	Hello Packet

在此数据包捕获中，即使在R1、R2和R3之间的共享网段上创建重复，您也看不到任何断言数据包。激活(S, G)流时，为什么不看到任何PIM断言数据包？

RFC 7761似乎可以回答这些问题。

## 摘自RFC 7761第4.2.2节。

### 4.2.2. Setting and Clearing the (S,G) SPTbit

Basically, Update\_SPTbit(S,G,iif) will set the SPTbit if we have the appropriate (S,G) join state, and if the packet arrived on the correct upstream interface for S, and if one or more of the following conditions apply:

1. The source is directly connected, in which case the switch to the SPT is a no-op.
2. The RPF interface to S is different from the RPF interface to the RP. The packet arrived on RPF\_interface(S), and so the SPT must have been completed.
3. No one wants the packet on the RP tree.
4.  $RPF'(S,G) == RPF'(*,G)$ . In this case, the router will never be able to tell if the SPT has been completed, so it should just switch immediately. The  $RPF'(S,G) != NULL$  check ensures that the SPTbit is set only if the RPF neighbor towards S is valid.

**In the case where the RPF interface is the same for the RP and for S, but  $RPF'(S,G)$  and  $RPF'(*,G)$  differ, we wait for an Assert(S,G), which indicates that the upstream router with (S,G) state believes the SPT has been completed.**

(S, G)SPT位用于区分是在(\*,G)状态还是在(S, G)状态转发。当从RP树切换到源树时，当建立上游(\*,G)状态时，数据会由于上游(\*,G)状态而到达，此时，路由器应继续仅在(\*,G)状态转发。这防止了在上游(S, G)状态完成之前发送修剪(S, G, rpt)而导致的临时黑洞。

虽然场景似乎可以与上述最后一点关联。如果RP和S的RPF接口相同，但RPF'(S, G)和RPF'(\*,G)不同，我们等待Assert(S, G)，这表示具有(S, G)状态的上游路由器认为SPT已完成。

要触发断言，路由器必须在其已填充的OIL上为网段上的同一源IP/目标组接收重复数据包。R3也是LHR，这意味着当从(\*,G)接收数据包时，它被指定从(\*,G)切换到SPT(S, G)。

在数据包捕获中，我们观察到未触发任何断言。虽然我们确实看到在收到第一个ICMP回应后立即发送修剪。

The screenshot shows a Wireshark capture of network traffic. The main table lists packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. Packet 13 is highlighted in blue. Below the table, the packet details pane is expanded to show the PIM Options section, where the SR flag is checked under 'IP address: 10.0.0.2/32 (SR)'. The status bar at the bottom indicates 'PIM Options (pim.option), 30 bytes'.

No.	Time	Source	Destination	Protocol	Length	Info
7	11.954130	192.168.3.1	224.0.0.13	PIMv2	72	Hello
8	12.621371	192.168.3.3	224.0.0.13	PIMv2	72	Hello
9	13.015136	192.168.3.3	224.0.0.5	OSPF	98	Hello Packet
10	19.046520	192.168.3.1	224.0.0.5	OSPF	98	Hello Packet
11	19.670571	192.168.3.2	224.0.0.5	OSPF	98	Hello Packet
12	22.114741	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=0/0, ttl=253 (multicast)
13	22.137371	192.168.3.3	224.0.0.13	PIMv2	68	Join/Prune
14	22.137597	192.168.3.3	224.0.0.13	PIMv2	68	Join/Prune
15	22.972394	192.168.3.3	224.0.0.5	OSPF	98	Hello Packet
16	23.085520	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=1/256, ttl=253 (multicast)
17	24.087827	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=2/512, ttl=253 (multicast)
18	24.723777	192.168.3.3	224.0.0.13	PIMv2	96	Join/Prune
19	25.088340	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=3/768, ttl=253 (multicast)
20	26.001246	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=4/1024, ttl=253 (multicast)

```

> Frame 13: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
  > Ethernet II, Src: Cheertek_e7:cc:00 (00:15:e5:e7:cc:00), Dst: IPv4mcast_0d (01:00:5e:00:00:0d)
    > Destination: IPv4mcast_0d (01:00:5e:00:00:0d)
    > Source: Cheertek_e7:cc:00 (00:15:e5:e7:cc:00)
    Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: 192.168.3.3, Dst: 224.0.0.13
  > Protocol Independent Multicast
    0010 .... = Version: 2
    ... 0011 = Type: Join/Prune (3)
    Reserved byte(s): 00
    Checksum: 0x163d [correct]
    [Checksum Status: Good]
  > PIM Options
    Upstream-neighbor: 192.168.3.1
    Reserved byte(s): 00
    Num Groups: 1
    Holdtime: 210
    > Group 0: 239.1.1.1/32
      Num Joins: 0
      > Num Prunes: 1
        IP address: 10.0.0.2/32 (SR)
  
```

如您所见，一旦R3接口G1收到第一个Internet控制消息协议(ICMP)请求数据包，就会向上游邻居192.168.3.1发送(\*,G)SR位修剪。此修剪(\*,G)用于定义的特定源。

您还可以看到以下标志设置：(SR):

The S flag: indicates that the multicast group is a sparse mode group.

The R flag: The R flag is the RP-bit flag and indicates that the information in the (S, G) entry is applicable to the shared tree.

在第二个PIM数据包第14号中，您可以看到R3尝试加入(S, G)树。

\*Standard input [SW1 Ethernet2 to R3 Gi1]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F> Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
7	11.954130	192.168.3.1	224.0.0.13	PIMv2	72	Hello
8	12.621371	192.168.3.3	224.0.0.13	PIMv2	72	Hello
9	13.015136	192.168.3.3	224.0.0.5	OSPF	98	Hello Packet
10	19.046520	192.168.3.1	224.0.0.5	OSPF	98	Hello Packet
11	19.670571	192.168.3.2	224.0.0.5	OSPF	98	Hello Packet
12	22.114741	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=0/0, ttl=253 (multicast)
13	22.137371	192.168.3.3	224.0.0.13	PIMv2	68	Join/Prune
14	22.137597	192.168.3.3	224.0.0.13	PIMv2	68	Join/Prune
15	22.972394	192.168.3.3	224.0.0.5	OSPF	98	Hello Packet
16	23.085520	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=1/256, ttl=253 (multicast)
17	24.087827	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=2/512, ttl=253 (multicast)
18	24.723777	192.168.3.3	224.0.0.13	PIMv2	96	Join/Prune
19	25.088340	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=3/768, ttl=253 (multicast)
20	26.091246	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000d, seq=4/1024, ttl=253 (multicast)

> Frame 14: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0

▼ Ethernet II, Src: Cheertek\_e7:cc:00 (00:15:e5:e7:cc:00), Dst: IPv4mcast\_0d (01:00:5e:00:00:0d)

- > Destination: IPv4mcast\_0d (01:00:5e:00:00:0d)
- > Source: Cheertek\_e7:cc:00 (00:15:e5:e7:cc:00)
- Type: IPv4 (0x0800)

> Internet Protocol Version 4, Src: 192.168.3.3, Dst: 224.0.0.13

▼ Protocol Independent Multicast

- 0010 .... = Version: 2
- .... 0011 = Type: Join/Prune (3)
- Reserved byte(s): 00
- Checksum: 0x173c [correct]
- [Checksum Status: Good]

▼ PIM Options

- Upstream-neighbor: 192.168.3.2
- Reserved byte(s): 00
- Num Groups: 1
- Holdtime: 210
- ▼ Group 0: 239.1.1.1/32
- ▼ Num Joins: 1
- IP address: 10.0.0.2/32 (S)
- Num Prunes: 0

wireshark\_-\_20171228095051\_a07600 | Packets: 25 · Displayed: 25 (100.0%) · Dropped: 0 (0.0%) | Profile: Default

观察到，一旦接收到第一数据平面，数据包R3会剪除(\*,G)并构建(S, G)。这就是您看不到PIM断言数据包的原因。当您的LHR具有(S, G)和(\*,G)的相同RPF接口时，此特定场景生效。虽然此行为可能与RFC 7761稍有不同，但不应导致任何问题。

现在，我们继续说一下场景2。此场景的图表可在此处查看：

## 场景2.断言路径选择

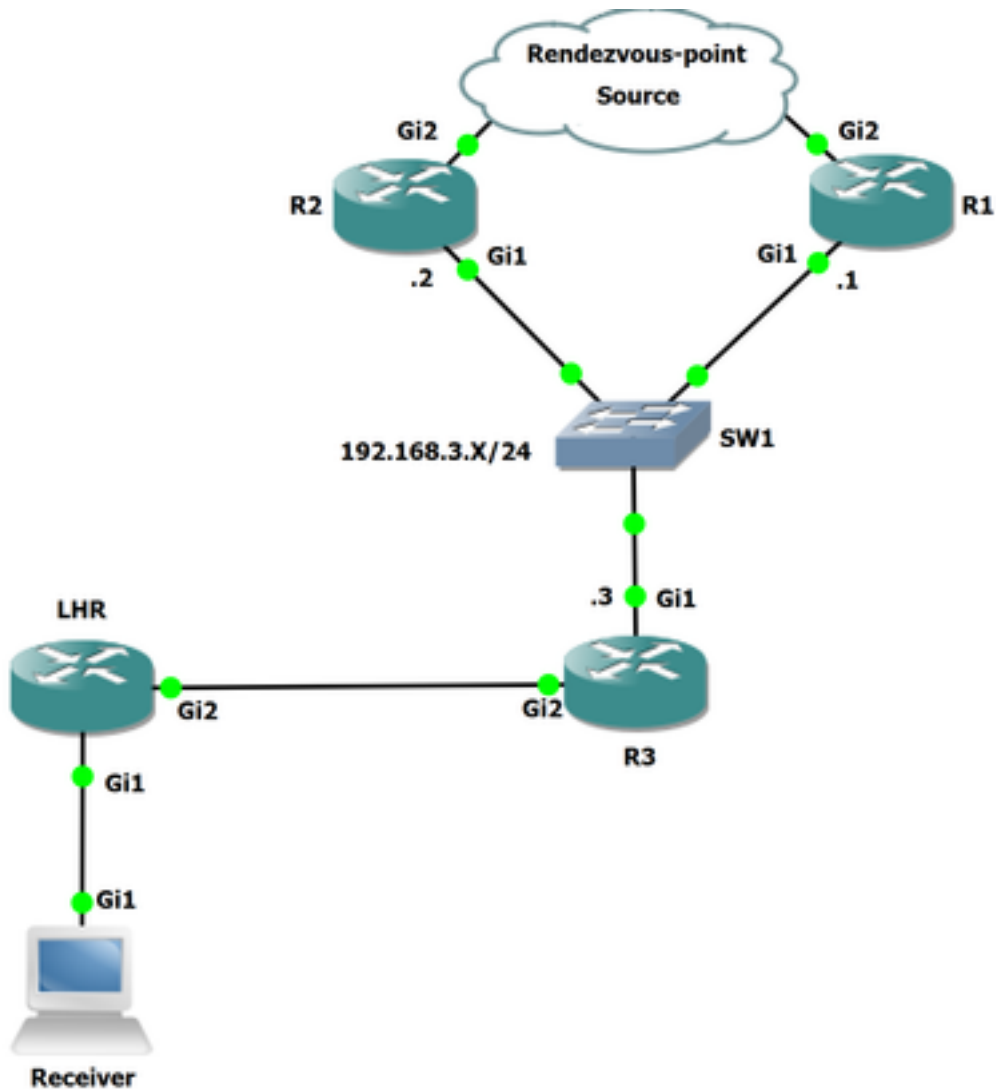


图 2.

在此拓扑中，R3上连接了另一台路由器，即LHR。LHR直接连接到接收器。源和RP都位于R2和R1上方。R3上指向RP的RPF邻居是R1，而指向源的RPF邻居是R2。

让我们检查源和RP的RPF邻居。

此处显示指向RP的RPF邻居：192.168.0.100是192.168.3.1。

```
R3#show ip rpf 192.168.0.100
RPF information for ? (192.168.0.100)
  RPF interface: GigabitEthernet1
  RPF neighbor: ? (192.168.3.1)
  RPF route/mask: 192.168.0.100/32
  RPF type: unicast (ospf 1)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base, originated from ipv4 unicast base
```

此处您看到指向源的RPF邻居：10.0.0.2是192.168.3.2。

```
R3#show ip rpf 10.0.0.2
RPF information for ? (10.0.0.2)
  RPF interface: GigabitEthernet1
  RPF neighbor: ? (192.168.3.2)
```

```
RPF route/mask: 10.0.0.0/24
RPF type: unicast (ospf 1)
Doing distance-preferred lookups across tables
RPF topology: ipv4 multicast base, originated from ipv4 unicast base
```

在激活源之前，让我们查看R3上的mroute表，因为您可以看到组239.1.1.1已有(\*,G)。这是因为连接到LHR的接收方已请求指定组。

```
R3#show ip mroute
IP Multicast Routing Table
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.1.1.1), 00:00:57/00:02:32, RP 192.168.0.100, flags: S
  Incoming interface: GigabitEthernet1, RPF nbr 192.168.3.1
  Outgoing interface list:
    GigabitEthernet2, Forward/Sparse, 00:00:57/00:02:32

(*, 224.0.1.40), 00:11:24/00:02:41, RP 192.168.0.100, flags: SJCL
  Incoming interface: GigabitEthernet1, RPF nbr 192.168.3.1
  Outgoing interface list:
    GigabitEthernet2, Forward/Sparse, 00:02:02/00:02:41
```

现在，激活R3接口Gi1上的源数据包并捕获数据包。

The screenshot shows a Wireshark capture on interface SW1 Ethernet2 to R3 Gi1. The packet list pane shows several OSPF Hello packets and PIMv2 Join/Prune/Assert packets. Packet 11 is selected, and its details pane is expanded to show the PIM Options section.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.3.1	224.0.0.5	OSPF	98	Hello Packet
2	3.164783	192.168.3.3	224.0.0.13	PIMv2	68	Join/Prune
3	5.264729	192.168.3.3	224.0.0.13	PIMv2	68	Join/Prune
4	7.447012	192.168.3.2	224.0.0.5	OSPF	98	Hello Packet
5	8.150289	192.168.3.3	224.0.0.5	OSPF	98	Hello Packet
6	9.674810	192.168.3.1	224.0.0.5	OSPF	98	Hello Packet
7	12.016714	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000f, seq=0/0, ttl=253 (multicast)
8	12.166782	192.168.3.3	224.0.0.13	PIMv2	68	Join/Prune
9	13.974441	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000f, seq=1/256, ttl=253 (multicast)
10	13.975383	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000f, seq=1/256, ttl=253 (multicast)
11	13.980084	192.168.3.1	224.0.0.13	PIMv2	62	Assert
12	13.980901	192.168.3.2	224.0.0.13	PIMv2	62	Assert
13	15.976508	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000f, seq=2/512, ttl=253 (multicast)
14	16.865001	192.168.3.3	224.0.0.13	PIMv2	68	Join/Prune
15	17.334577	192.168.3.2	224.0.0.5	OSPF	98	Hello Packet
16	17.987218	10.0.0.2	239.1.1.1	ICMP	114	Echo (ping) request id=0x000f, seq=3/768, ttl=253 (multicast)
17	18.032846	192.168.3.3	224.0.0.5	OSPF	98	Hello Packet

Frame 11: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0  
> Ethernet II, Src: Cheertek\_9c:3a:00 (00:15:e5:9c:3a:00), Dst: IPv4mcast\_0d (01:00:5e:00:00:0d)  
> Internet Protocol Version 4, Src: 192.168.3.1, Dst: 224.0.0.13  
▼ Protocol Independent Multicast  
 0010 .... = Version: 2  
 ... 0101 = Type: Assert (5)  
 Reserved byte(s): 00  
 Checksum: 0x5e6a [correct]  
 [Checksum Status: Good]  
 ▼ PIM Options  
 Group: 239.1.1.1/32  
 Source: 10.0.0.2  
 1... .... = RP Tree: True  
 .000 0000 0000 0000 0000 0110 1110 = Metric Preference: 110  
 Metric: 2

正如您在此数据包捕获中看到的，PIM断言数据包已存在。



## 帧 11:

```
> Frame 11: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0
> Ethernet II, Src: Cheertek_9c:3a:00 (00:15:e5:9c:3a:00), Dst: IPv4mcast_0d (01:00:5e:00:00:0d)
> Internet Protocol Version 4, Src: 192.168.3.1, Dst: 224.0.0.13
v Protocol Independent Multicast
  0010 .... = Version: 2
  .... 0101 = Type: Assert (5)
  Reserved byte(s): 00
  Checksum: 0x5e6a [correct]
  [Checksum Status: Good]
v PIM Options
  Group: 239.1.1.1/32
  Source: 10.0.0.2
  1... .... = RP Tree: True
  .000 0000 0000 0000 0000 0000 0110 1110 = Metric Preference: 110
  Metric: 2
```

## 帧 12:

```
> Frame 12: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0
> Ethernet II, Src: Cheertek_8b:3e:00 (00:15:e5:8b:3e:00), Dst: IPv4mcast_0d (01:00:5e:00:00:0d)
> Internet Protocol Version 4, Src: 192.168.3.2, Dst: 224.0.0.13
v Protocol Independent Multicast
  0010 .... = Version: 2
  .... 0101 = Type: Assert (5)
  Reserved byte(s): 00
  Checksum: 0xde6a [correct]
  [Checksum Status: Good]
v PIM Options
  Group: 239.1.1.1/32
  Source: 10.0.0.2
  0... .... = RP Tree: False
  .000 0000 0000 0000 0000 0000 0110 1110 = Metric Preference: 110
  Metric: 2
```

当您查看这些数据包时，您应该能够确定谁是断言的赢家。现在，我们来了解一下PIM断言转发器选择。

度量首选项是管理距离(AD)。 这是指在路由表中安装路由的路由协议的管理距离，用于查找源IP地址，度量是路由的开销。

还有其他属性用于确定谁是断言赢家。您可以在RFC 7761中看到这些详细信息。

## 摘自RFC 7761第4.6.3节。

### 4.6.3. Assert Metrics

Assert metrics are defined as:

```
struct assert_metric {
    rpt_bit_flag;
    metric_preference;
    route_metric;
    ip_address;
};
```

When comparing assert\_metrics, the rpt\_bit\_flag, metric\_preference, and route\_metric fields are compared in order, where the first lower

value wins. If all fields are equal, the primary IP address of the router that sourced the Assert message is used as a tie-breaker, with the highest IP address winning.

使用定义的这些字段和路径选择，您可以确定在此场景中断言获胜者是谁。如果再次查看断言数据包，您会看到度量首选项未进行比较，因为决策是根据第一个选择标准rpt\_bit\_flag做出的。

在此场景中，比较R1和R2。两台路由器都发送以前看到的断言消息，并且当两台设备看到对方的断言消息后，它们可以相互比较度量以确定谁是赢家。

由于R2发送了包含RP树的断言消息：False（值为0），它确实比R1随RP树发送的值低：值为1的True。RP树位设置为0或1。

RP树位设置为1表示您当前在共享树上；清除RPT位表示断言的发送方在接口上处于(S, G)转发状态。

当(S, G)断言优先于(\*,G)断言时，R2应成为断言赢家。请转换到“我是断言赢家”状态。如RFC 7761中前面的语句中所述，更优先选择较低的值。

让我们看看R1和R2，看看谁是断言赢家。

```
R2#show ip mroute
```

```
IP Multicast Routing Table
```

```
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
```

```
Timers: Uptime/Expires
```

```
Interface state: Interface, Next-Hop or VCD, State/Mode
```

```
(* , 239.1.1.1), 00:42:52/stopped, RP 192.168.0.100, flags: SP
```

```
Incoming interface: GigabitEthernet2, RPF nbr 192.168.4.1
```

```
Outgoing interface list: Null
```

```
(10.0.0.2, 239.1.1.1), 00:42:52/00:01:40, flags: T
```

```
Incoming interface: GigabitEthernet2, RPF nbr 192.168.4.1
```

```
Outgoing interface list:
```

```
GigabitEthernet1, Forward/Sparse, 00:42:52/00:03:07, A
```

```
(* , 224.0.1.40), 00:43:23/00:02:25, RP 192.168.0.100, flags: SJPL
```

```
Incoming interface: GigabitEthernet2, RPF nbr 192.168.4.1
```

```
Outgoing interface list: Null
```

在此输出中，您可以看到R2上的(S, G)在OIL上设置了A标志，表明它是断言赢家。在R1上，您在(S, G)上没有OIL，并且设置了P标志，这表示在本例中，特定(S, G)已被剪除：它不是断言的赢家。

**注意：**当在共享网段上存在assert时，下游邻居将Join(\*,G)和Join(S, G)定期消息发送到适当的RPF邻居，即RPF邻居，如assert进程所修改。它们并不总是按照MRIB的指示发送到RPF邻居。

```
R1#show ip mroute
```

```
IP Multicast Routing Table
```

```
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
```

```
Timers: Uptime/Expires
```

```
Interface state: Interface, Next-Hop or VCD, State/Mode
```

```
(*, 239.1.1.1), 00:44:32/00:03:09, RP 192.168.0.100, flags: S
  Incoming interface: GigabitEthernet2, RPF nbr 192.168.5.2
  Outgoing interface list:
    GigabitEthernet1, Forward/Sparse, 00:44:32/00:03:09, A

(10.0.0.2, 239.1.1.1), 00:44:19/00:03:09, flags: PR
  Incoming interface: GigabitEthernet2, RPF nbr 192.168.5.2
  Outgoing interface list: Null

(*, 224.0.1.40), 00:44:50/00:02:53, RP 192.168.0.100, flags: SJCL
  Incoming interface: GigabitEthernet2, RPF nbr 192.168.5.2
  Outgoing interface list:
    GigabitEthernet1, Forward/Sparse, 00:43:56/00:02:53
```

如果R1和R2的RP树位均设置为1，则可以考虑AD最低的路由器；如果等于，则查看度量。如果两台路由器上的RP树位都为true，则将度量与RP IP地址进行比较。如果RP树位为0，则将度量与组播流的源进行比较。

如果所有这些值都相同，则最高IP地址来源断言消息是获胜者。

## 摘要

在场景1中，您没有观察断言数据包，但是，根据RFC，它们应该已被触发。如前所述，这是因为R3在(S, G)的控制平面建立之前修剪(\*,G)。

在场景2中，您会看到断言数据包。在LHR上收到第一个数据包时，它会向R3发送(S, G)加入/修剪以拉取源/组。然后，R3将向R2发送同一源/组的加入/修剪数据包。这将导致R1和R2都填充有效OIL。现在，R3在R3(S, G)状态上填充T标志时，只会剪除(S, G)RP位集。要实现此目的，您需要从共享网段接收另一个数据平面数据包。由于已为(S, G)构建控制平面，因此会导致在共享网段上重复触发断言消息。