

PIM Assert-Mechanismus verstehen

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Was ist der PIM Assert Mechanism?](#)

[Szenario 1. LHR-Begründung](#)

[Abstract von RFC 7761, Abschnitt 4.2.2.](#)

[Szenario 2. Pfadauswahl übernehmen](#)

[Abstract von RFC 7761, Abschnitt 4.6.3.](#)

[Zusammenfassung](#)

Einleitung

Dieses Dokument beschreibt den Mechanismus zur Bestätigung von Protocol Independent Multicast (PIM), konzentriert sich auf die Kriterien für die Gewinner des PIM-Tests und geht näher auf bestimmte Eckpunkte ein.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über Kenntnisse des PIM-Assertionsmechanismus zu verfügen.

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf Cisco CSR1000V Version 16.4.1.

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Was ist der PIM Assert Mechanism?

Wenn mehrere PIM-fähige Router auf einem gemeinsam genutzten Segment vorhanden sind, können diese Router auf duplizierten Multicast-Datenverkehr stoßen. Dies kann der Fall sein, da zwei oder mehr Router im gleichen freigegebenen Segment möglicherweise einen gültigen (S,G) oder (*,G)-Eintrag haben, der die ausgehende Schnittstelle für das gemeinsam genutzte Segment für dieselbe Quell-IP/Zielgruppe ausfüllt.

Der PIM-Assertionsmechanismus dient zum Erkennen und Eliminieren von Duplikaten von Multicast-Datenverkehr in einem gemeinsam genutzten Segment. Dieser Mechanismus verhindert

nicht die doppelte Ausführung von Datenverkehr, sondern nutzt die doppelte Ausführung von Multicast-Datenverkehr als Auslöser, um diesen Mechanismus zu aktivieren, der einen einzelnen Forwarder für diesen Stream auswählt.

Wenn in einem gemeinsam genutzten Segment mehrere Multicast-Datenverkehr dupliziert werden, können Sie davon ausgehen, dass es mehrere Router gibt, die dasselbe (S,G) oder (*,G) in einem gemeinsam genutzten Segment senden. Wenn Sie einen Router auswählen, um diesen Stream effektiv weiterzuleiten, werden Doppelarbeit vermieden.

PIM nutzt PIM-Assert-Nachrichten, die ausgelöst werden, wenn Sie in der OIL (Outgoing Interface List) ein Multicast-Paket empfangen. Diese Assert-Meldungen enthalten Kennzahlen, anhand derer berechnet wird, wer Gewinner des Tests wird. Downstream-Router im LAN empfangen ebenfalls PIM-Assert-Nachrichten. Diese Nachrichten werden dann von Downstream-Geräten verwendet, um entsprechende Join/Prune-Nachrichten an den Upstream-Router zu senden, der die Assert-Auswahl gewonnen hat.

Szenario 1. LHR-Begründung

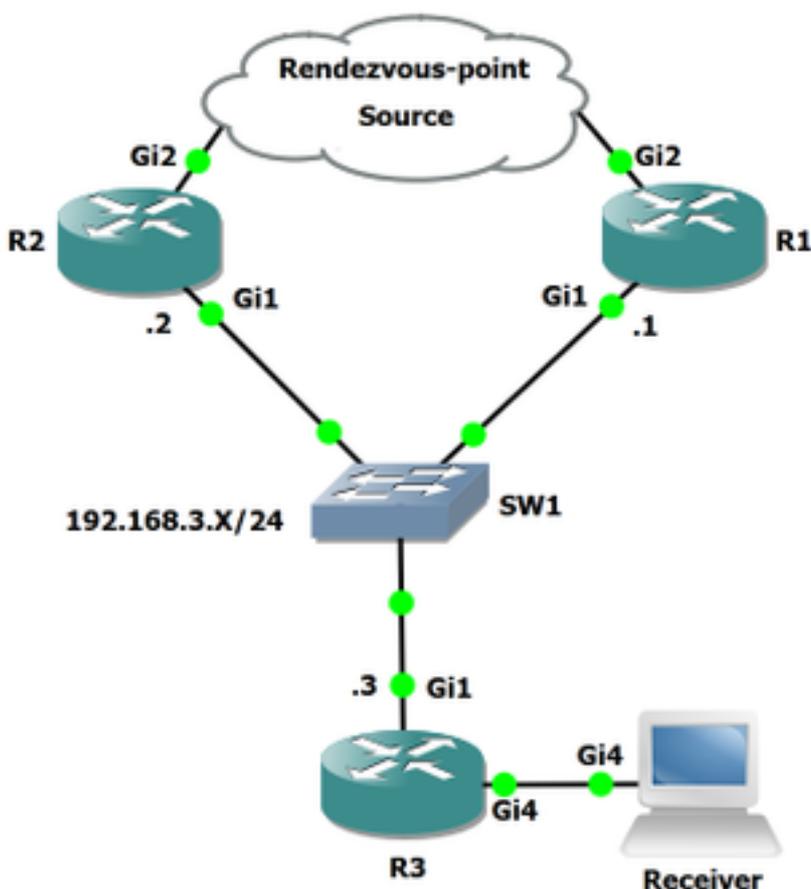


Abbildung 1:

Im Netzwerkdiagramm ist R3 der Last-Hop-Router (LHR). R3 ist über ein gemeinsam genutztes Segment mit R2 und R1 verbunden.

Wenn Sie vom Empfänger einen IGMP-Bericht (Internet Group Management Protocol) erhalten, prüft R3, wer der RPF-Nachbarn zum RP ist. In der Topologie ist R1 der RPF-Nachbarn zum RP, daher sendet R3 eine (*,G)-Verbindung zu R1. Sobald R1 den Stream abrufen (unter der Annahme, dass die Gruppe aktiv ist), sendet R3 eine (S,G)-Verbindung an die Quelle und zieht den Quellbaum nach unten. R2 ist der RPF-Nachbar zum Source Tree, d. h. R3 sendet die (S,G)-Join-

Nachricht an R2. R3 verfügt über dieselbe RPF-Schnittstelle sowohl zum RP als auch zur Quelle. Hier sehen Sie die R3 mroute-Tabelle für die Gruppe 239.1.1.1.

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: SJPCL
  Incoming interface: GigabitEthernet1, RPF nbr 192.168.3.1
```

Wie Sie auf R3 sehen können, ist der RPF-Nachbar (*,G) 192.168.3.1 und der RPF-Nachbar in Richtung (S,G) 192.168.3.2. Nun sollte dies sowohl R1 als auch R2 dazu führen, dass ein gültiges OIL für R3 vorhanden ist. Sehen wir uns die folgenden Einträge an:

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), 00:15:02/00:02:33, 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

Wie bereits erwähnt, kann Assert ausgelöst werden, wenn zwei Upstream-Router über eine gültige OIL verfügen, die in einem gemeinsam genutzten Segment enthalten ist. Da sowohl R1 als auch R2 über eine gültige OIL verfügen, prüfen Sie, ob bei der Paketerfassung ein Assertionsmechanismus vorhanden ist.

Diese Paketerfassung wurde auf der R3-Schnittstelle Gi1 zu SW1 erfasst.

The screenshot shows a Wireshark capture on interface SW1 Ethernet2 to R3 Gi1. The packet list pane displays 25 captured packets. The first 12 packets are OSPF Hello packets from various sources (192.168.3.1, 192.168.3.2, 192.168.3.3) to destination 224.0.0.5. Packets 13-14 are PIMv2 Join/Prune messages. Packets 15-16 are OSPF Hello packets. Packets 17-18 are ICMP Echo (ping) requests to 239.1.1.1. Packets 19-22 are more ICMP Echo requests. Packets 23-24 are OSPF Hello packets. Packet 25 is an OSPF Hello packet. The packet bytes pane shows the hex and ASCII representation of the first packet: 0000 01 00 5e 00 00 05 00 15 e5 9c 3a 00 08 00 45 c0 ..^.....E..

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

Bei dieser Paketerfassung werden keine Assert-Pakete angezeigt, obwohl alle Voraussetzungen für die Erstellung von Duplikaten auf dem gemeinsam genutzten Segment zwischen R1, R2 und R3 gegeben sind. Warum werden bei Aktivierung des (S,G)-Streams keine PIM-Assert-Pakete angezeigt?

Es scheint, dass RFC 7761 die Antwort auf diese Fragen enthalten könnte.

Abstract von RFC 7761, Abschnitt 4.2.2.

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 $RPF'(S,G) != \text{NULL}$ 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.

Das (S,G) SPTbit wird verwendet, um zu unterscheiden, ob ein (*,G)- oder ein (S,G)-Zustand weitergeleitet werden soll. Wenn Sie von der RP-Struktur zur Quellstruktur wechseln, gibt es eine Übergangsperiode, in der Daten aufgrund des Upstream-Status (*,G) eintreffen, während der Upstream-Status (S,G) eingerichtet ist. Zu diesem Zeitpunkt sollte der Router nur im Status (*,G) weiterleiten. Dies verhindert temporäre schwarze Löcher, die durch das Senden eines Prune(S,G,rpt) verursacht werden, bevor der Upstream-Status (S,G) fertig gestellt ist.

Obwohl es scheint, dass das Szenario mit dem letzten oben erwähnten Punkt korrelieren kann. Wenn die RPF-Schnittstelle für den RP und für S identisch ist, Allerdings unterscheiden sich $RPF'(S,G)$ und $RPF'(*,G)$. Wir warten auf einen Assert(S,G), der anzeigt, dass der Upstream-Router mit dem Status (S,G) der Meinung ist, dass die SPT abgeschlossen wurde.

Um die Anforderung auszulösen, muss der Router ein doppeltes Paket auf seiner bereits ausgefüllten OIL für dieselbe Quell-IP/Zielgruppe im Segment erhalten. R3 ist auch ein LHR, d. h., es ist für den Wechsel von (*,G) zu SPT (S,G) vorgesehen, wenn ein Paket von (*,G) empfangen wird.

Bei der Paketerfassung stellen wir fest, dass keine Assertionen ausgelöst werden. Obwohl ein Prune sofort nach dem Empfang des ersten ICMP-Echos gesendet wird.

*Standard input [SW1 Ethernet2 to R3 Gi1]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> 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 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)

PIM Options (pim.option), 30 bytes

Packets: 25 · Displayed: 25 (100.0%) · Dropped: 0 (0.0%) Profile: Default

Wie Sie sehen können, wird nach dem Empfang des ersten ICMP-Anforderungspakets (Internet Control Message Protocol) an die R3-Schnittstelle G1 ein (*,G) SR-Bit-Prune an den Upstream-Nachbarn 192.168.3.1 gesendet. Diese Abstriche (*,G) für die spezifische definierte Quelle.

Folgende Markierungen werden ebenfalls festgelegt: (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.

Im zweiten PIM-Paket Nr. 14 sehen Sie, dass R3 versucht, dem (S,G)-Tree beizutreten.

*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

Es wird beobachtet, dass das Paket R3 nach dem Empfang der ersten Datenebene die (*,G) löscht und die (S,G) erstellt. Aus diesem Grund werden PIM-Assertpakete nicht angezeigt. Dieses Szenario gilt, wenn Sie über einen LHR verfügen, der über dieselbe RPF-Schnittstelle für (S,G) und (*,G) verfügt. Obwohl dieses Verhalten leicht von RFC 7761 abweichen kann, sollte es keine Probleme verursachen.

Fahren wir nun mit Szenario 2 fort. Das Diagramm dieses Szenarios ist hier zu sehen:

Szenario 2. Pfadauswahl übernehmen

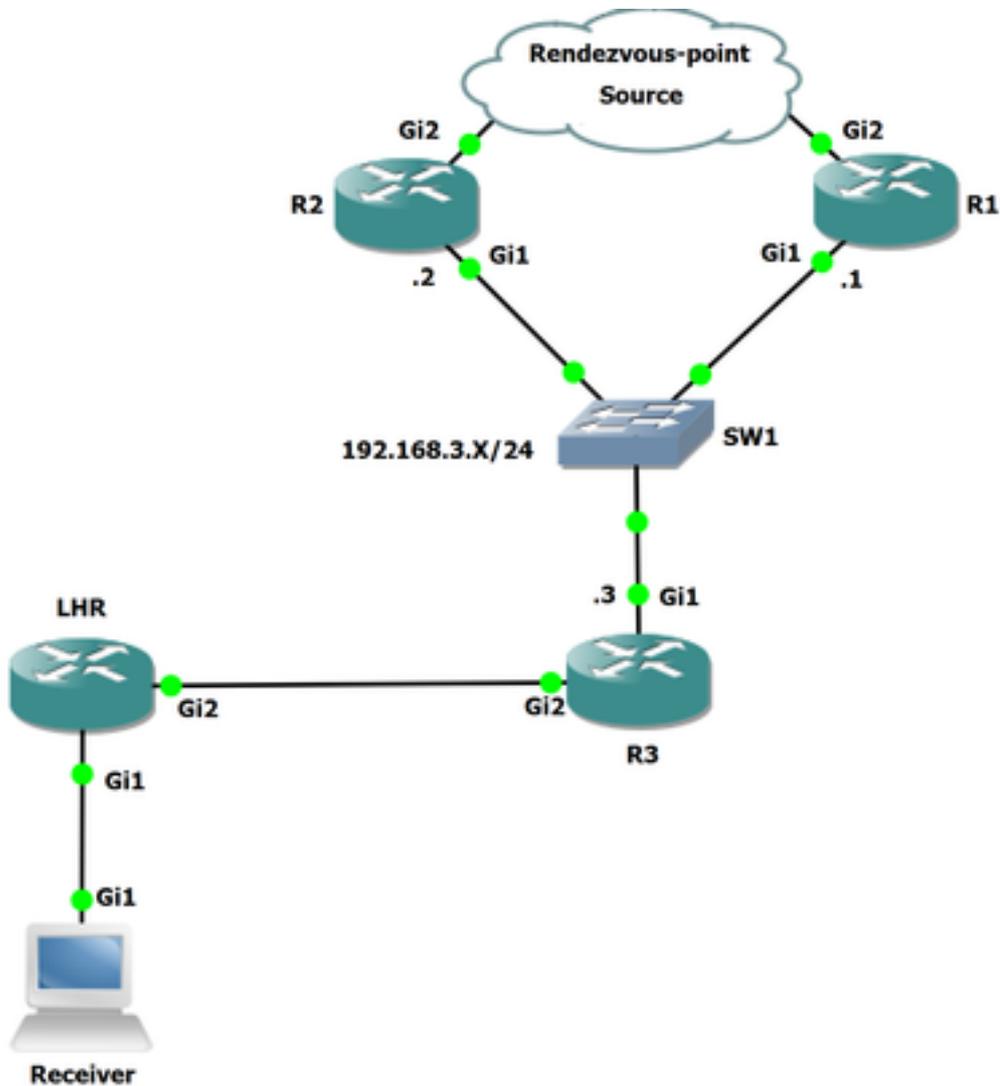


Abbildung 2:

In dieser Topologie ist auf R3 ein weiterer Router angeschlossen, der LHR ist. Der LHR wird direkt mit dem Empfänger verbunden. Quelle und RP sind beide über R2 und R1. Der RPF-Nachbar auf R3 zum RP ist R1, und der RPF-Nachbar zur Quelle ist R2.

Sehen wir uns nun den RPF-Nachbarn an, um sowohl die Quelle als auch den RP zu prüfen.

Hier sehen Sie den RPF-Nachbarn zum RP: 192.168.0.100 ist 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
```

Hier sehen Sie den RPF-Nachbarn zur Quelle: 10.0.0.2 ist 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

```

Bevor wir die Quelle aktivieren, schauen wir uns die mroute-Tabelle auf R3 an, wie Sie sehen können, dass es bereits (*,G) für die Gruppe 239.1.1.1 gibt. Der Grund hierfür ist, dass der mit LHR verbundene Empfänger bereits für die angegebene Gruppe angefordert wurde.

```

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

```

Aktivieren Sie jetzt die Quelle, und erfassen Sie Pakete auf der R3-Schnittstelle Gi1.

The screenshot shows a Wireshark capture of network traffic on interface 0. The main packet list table is as follows:

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

The detailed view for Frame 11 (Packet 11) is shown below:

```

> 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

```

Wie Sie in dieser Paketerfassung sehen, bestätigt PIM, dass Pakete bereits vorhanden sind.

Bild 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
▼ 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 0000 0110 1110 = Metric Preference: 110
  Metric: 2
```

Bild 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
▼ Protocol Independent Multicast
  0010 .... = Version: 2
  .... 0101 = Type: Assert (5)
  Reserved byte(s): 00
  Checksum: 0xde6a [correct]
  [Checksum Status: Good]
▼ 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
```

Wenn Sie sich diese Pakete ansehen, sollten Sie in der Lage sein zu bestimmen, wer der Gewinner ist. Sehen wir uns nun die Auswahl der PIM Assert Forwarder an.

Die metrische Voreinstellung ist die administrative Distanz (AD). Dies bezieht sich auf die administrative Distanz des Routing-Protokolls, das die Route in der Routing-Tabelle installiert. Diese Route wird zum Nachschlagen der Quell-IP-Adresse verwendet, und der Metric ist die Kosten der Route.

Es gibt auch andere Attribute, die verwendet werden, um zu bestimmen, wer der Gewinner ist. Sie können diese Details in RFC 7761 sehen.

Abstract von RFC 7761, Abschnitt 4.6.3.

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.

Durch die Verwendung dieser Felder und die Pfadauswahl können Sie bestimmen, wer der Gewinner des Asserts in diesem Szenario sein wird. Wenn Sie sich die Assert-Pakete erneut ansehen, sehen Sie, dass die metrische Präferenz nicht verglichen wird, da die Entscheidung für die ersten Auswahlkriterien getroffen wird, die `rpt_bit_flag` sind.

In diesem Szenario werden R1 und R2 verglichen. Beide Router senden Assert-Meldungen, die zuvor angezeigt wurden, und sobald beide Geräte die Assert-Meldungen des jeweils anderen Geräts sehen, können sie die Metriken miteinander vergleichen, um festzustellen, wer der Gewinner ist.

Da R2 eine Assert-Nachricht mit der RP-Struktur sendet: False, der den Wert 0 hat, ist tatsächlich niedriger als der Wert, den R1 mit einer RP-Struktur gesendet hat: True mit dem Wert 1. Das Bit der RP-Struktur ist auf 0 oder 1 eingestellt.

Wenn Sie das Bit für die RP-Struktur auf 1 setzen, bedeutet dies, dass Sie sich derzeit im Shared Tree befinden. Das gelöschte RPT-Bit gibt an, dass der Absender des Asserts den Weiterleitungsstatus (S,G) auf einer Schnittstelle hatte.

Wie (S,G) behauptet, hat die Priorität vor (*,G), sollte R2 der Gewinner sein. Übergang zum Status "Ich bin Assert Winner" Wie bereits in der früheren Anweisung in RFC 7761 erwähnt, wird der niedrigere Wert eher bevorzugt.

Schauen wir uns nun R1 und R2 an, um zu sehen, wer der Gewinner ist.

```
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
```

In dieser Ausgabe können Sie sehen, dass auf dem (S,G) auf R2 das A-Flag auf dem OIL gesetzt ist, das anzeigt, dass es der Gewinner ist. Hier auf R1 befindet sich kein OIL auf dem (S,G) und die P-Markierung ist gesetzt, was bedeutet, dass das jeweilige (S,G) in diesem Fall abgeschnitten

wurde: Es ist nicht der Gewinner des Festivals.

Anmerkung: Wenn Assertion auf einem gemeinsam genutzten Segment vorhanden ist, senden Downstream-Nachbarn periodische Meldungen über Join(*,G) und Join(S,G) an den entsprechenden RPF-Nachbarn, d. h. den RPF-Nachbarn in der durch den Assert-Prozess geänderten Fassung. Sie werden nicht immer, wie vom MRIB angegeben, an den RPF-Nachbarn gesendet.

```
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
```

Wenn es der Fall ist, dass sowohl R1 als auch R2 RP-Tree-Bit auf 1 festgelegt sind. Sie können dann den Router mit dem niedrigsten AD in Betracht ziehen. Wenn gleich, dann schauen Sie sich die Metrik an. Wenn das RP-Tree-Bit auf beiden Routern true ist, wird die Metrik mit der RP-IP-Adresse verglichen. Wenn das RP-Tree-Bit 0 ist, wird die Metrik mit der Quelle des Multicast-Streams verglichen.

Wenn alle diese Werte identisch sind, ist die höchste Bestätigungsmeldung zur IP-Adressenbeschaffung der Gewinner.

Zusammenfassung

In Szenario eins haben Sie keine Assert-Pakete beobachtet, aber sie hätten pro RFC ausgelöst werden sollen. Wie bereits erwähnt, lag dies daran, dass R3 vor dem Bau der Kontrollebene für (S,G) abgeschnitten wurde (*,G).

In Szenario zwei sehen Sie Assert-Pakete. Wenn das erste Paket auf LHR empfangen wurde, sendet es ein (S,G)-Join/Prune zu R3, um die Quelle/Gruppe abzuziehen. R3 sendet dann ein join/prune Paket an R2 für dieselbe Quelle/Gruppe. Dies würde dann dazu führen, dass sowohl R1 als auch R2 gültige ÖLs enthalten. R3 löscht jetzt nur noch (S,G) mit einem RP-Bit, wenn das T-Flag in den R3s (S,G)-Status eingetragen ist. Dazu müssen Sie ein weiteres Datenebenenpaket vom gemeinsam genutzten Segment erhalten. Da die Kontrollebene bereits für (S,G) erstellt wurde, führt dies zu Doppelungen im gemeinsam genutzten Segment, wodurch Assert-Meldungen ausgelöst werden.