

PIMアサートメカニズムについて

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[PIMアサートメカニズムとは何ですか。](#)

[シナリオ1. LHRの根拠](#)

[RFC 7761 セクション 4.2.2 からの抜粋.](#)

[シナリオ2.アサートパス選択](#)

[RFC 7761 セクション 4.6.3 からの抜粋.](#)

[要約](#)

概要

このドキュメントでは、Protocol Independent Multicast(PIM)アサートメカニズムについて説明し、PIMアサートの勝者の基準に集中し、特定のコーナーケースに深く割り込みます。

前提条件

要件

PIMアサートメカニズムに関する知識があることが推奨されます。

使用するコンポーネント

このドキュメントの情報は、Cisco CSR1000Vバージョン16.4.1に基づくものです

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

PIMアサートメカニズムとは何ですか。

共有セグメントに複数のPIM対応ルータがある場合、これらのルータで重複するマルチキャストトラフィックが発生する可能性があります。これは、同じ共有セグメント上の2つ以上のルータが、同じ送信元IP/宛先グループの共有セグメントに向けて発信インターフェイスを設定する有効な(S、G)または(*、G)エントリを持っている可能性があるためです。

PIMアサートメカニズムは、共有セグメントでのマルチキャストトラフィックの重複を検出して排除するために使用されます。このメカニズムでは、このストリームに対して単一のフォワードを選択するこのメカニズムをアクティブ化するために、マルチキャストトラフィックの重複がトリガーとして使用されるので、このメカニズムでは重複が防止されないことに注意してください

。

共有セグメントでマルチキャストトラフィックが重複している場合、共有セグメントで同じ(S、G)または(*、G)を送信するルータが複数あると仮定できます。このストリームを効果的に転送するルータを1つ選択すると、重複が排除されます。

PIMは、発信インターフェイスリスト(OIL)でマルチキャストパケットを受信したときにトリガーされるPIMアサートメッセージを利用します。これらのアサートメッセージにはメトリックが含まれており、このメトリックを使用して、アサートの勝者になるユーザを計算します。LAN上のダウンストリームルータもPIMアサートメッセージを受信します。次に、これらのメッセージはダウンストリームデバイスによって使用され、アサート選択を獲得したアップストリームルータに適切なJoin/Pruneメッセージを送信します。

シナリオ1. LHRの根拠

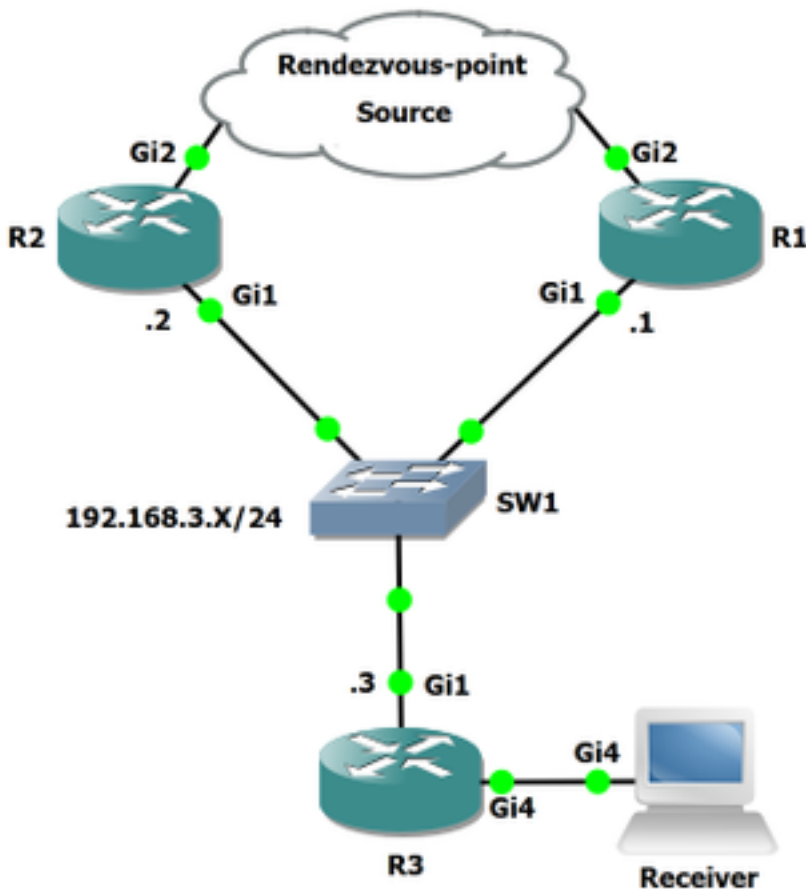


図 1.

ネットワークダイアグラムでは、R3がラストホップルータ(LHR)であり、R3は共有セグメントを介してR2とR1の両方に接続します。

レシーバからInternet Group Management Protocol(IGMP)レポートを受信すると、R3はRPに対するRPFネイバーが誰であるかをチェックします。トポロジでは、R1がRPに対するRPFネイバーであるため、R3はR1に対して(*,G)参加を送信します。R1がストリームをプルダウン(グループがアクティブであると仮定)すると、R3はソースに対して(S,G)参加を送信します。R2はソースツリーに向かうRPFネイバーで、R3はR2に(S,G)参加を送信します。R3はRPとソースの両方に向かう同じRPFインターフェイスを持っています。ここでは、グループ239.1.1.1のR3 mrouteテーブルが表示されます。

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
```

R3では、(*,G) RPFネイバーが192.168.3.1で、(S,G)に対するRPFネイバーが192.168.3.2であることがわかります。この結果、R1とR2の両方がR3に対して有効なOILになります。次のエントリをを確認します。

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

前述のように、有効なOILが共有セグメントに設定されている2つのアップストリームルータが存在する場合、アサートがトリガーされる可能性があります。R1とR2の両方に有効なOILがあるため、パケットキャプチャにアサートメカニズムがあるかどうかを確認します。

このパケットキャプチャは、SW1に対するR3インターフェイスGi1でキャプチャされました。

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

このパケットキャプチャでは、R1、R2、およびR3間の共有セグメントで重複を作成するための前提条件が存在していても、アサートパケットが表示されません。(S,G)ストリームがアクティブ化されたときにPIMアサートパケットが表示されないのはなぜですか。

RFC 7761では、これらの質問に対する回答が保持されている可能性があります。

RFC 7761 セクション 4.2.2 からの抜粋.

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) SPTbitは、順方向オン(*,G)またはオン(S,G)状態を区別するために使用されます。RPツリーからソースツリーに切り替えると、アップストリーム(S,G)状態が確立されている間に、データがアップストリーム(*,G)状態によって到着する遷移期間が発生します。このとき、ルータは(*,G)状態でのみ転送を続けます。これにより、アップストリーム(S,G)状態が確立される前に Prune(S,G,rpt)を送信することによって発生する一時的なブラックホールが防止されます。

このシナリオは上記の最後のポイントに関連付けられると思われませんが、RPFインターフェイスがRPとSで同じ場合、ただし、RPF'(S,G)とRPF'(*,G)は異なります。これは、(S,G)状態のアップストリームルータがSPTが完了したことを示すアサート(S,G)を待ちます。

アサートをトリガーするには、セグメント上の同じ送信元IPまたは宛先グループに対して、ルータがすでに入力されているOILで重複パケットを受信する必要があります。R3もLHRであり、(*,G)からパケットを受信したときに(*,G)からSPT(S,G)に切り替えるように指定されます。

パケットキャプチャでは、アサートがトリガーされないことがわかります。最初のICMPエコーを受信した直後にプルーンが送信されていることがわかります。

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)

PIM Options (pim.option), 30 bytes

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

ご覧のように、R3インターフェイスG1で最初のインターネット制御メッセージプロトコル

(ICMP)要求パケットが受信されると、(*,G) SRビットプルーニングがアップストリームのネイバー192.168.3.1に送信されます。このプルーニング(*,G)は、定義された特定の送信元です。

次のフラグも設定されています。(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.

2番目のPIMパケット番号14では、R3が(S,G)ツリーへの参加を試みることがわかります。

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

最初のデータプレーンが受信されると、パケットR3は(*,G)をプルーニングし、(S,G)を構築します。これが、PIMアサートパケットが表示されない理由です。この特定のシナリオは、(S,G)と(*,G)に対して同じRPFインターフェイスを持つLHRがある場合に有効です。この動作はRFC 7761とは若干異なる場合がありますが、問題は発生しません。

次に、シナリオ2を続けます。このシナリオの図を次に示します。

シナリオ2.アサートパス選択

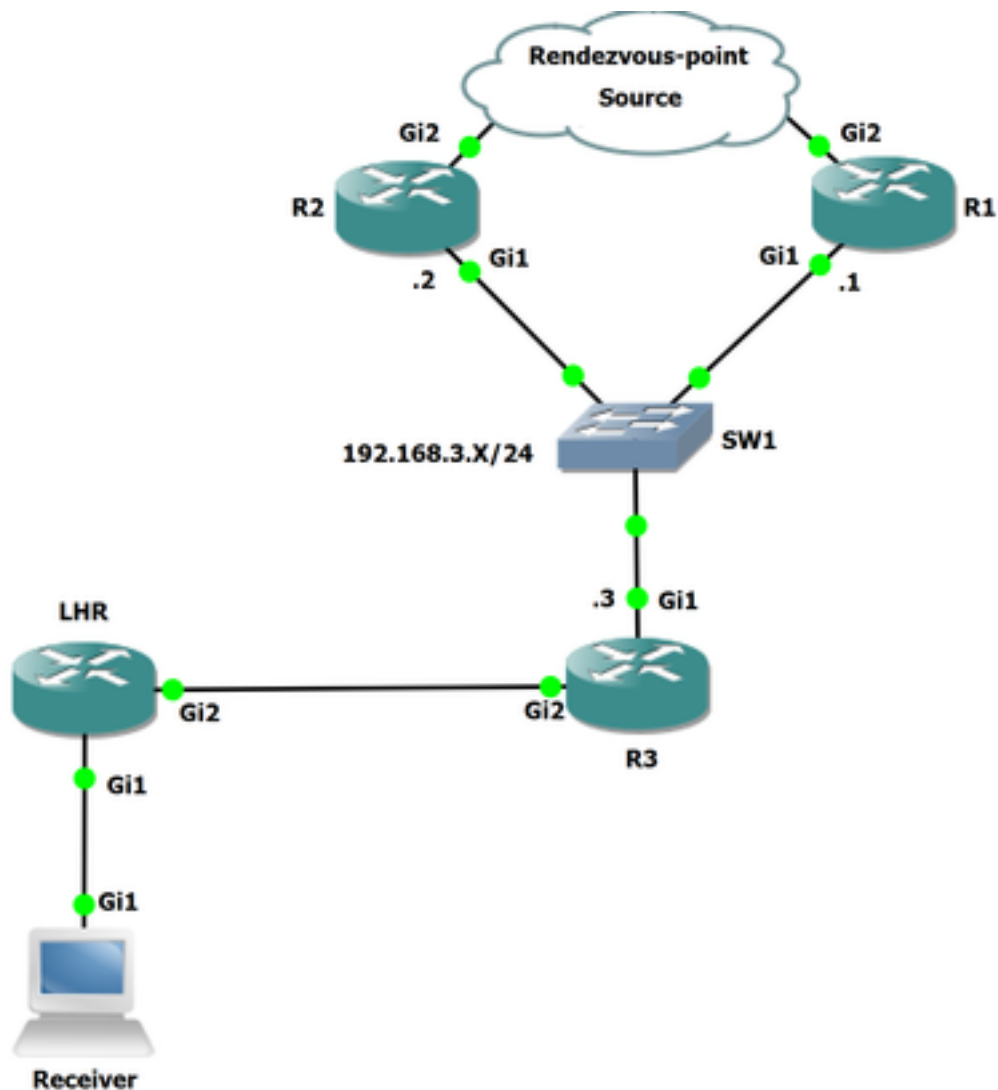


図 2 :

このトポロジでは、R3に接続された別のルータ(LHR)があります。LHRはレシーバに直接接続します。送信元とRPはどちらもR2とR1の上にあります。R3のRPに対するRPFネイバーはR1で、送信元に対するRPFネイバーはR2です。

送信元とRPの両方のRPFネイバーを確認します。

次に、RPに対するRPFネイバーを示します。192.168.0.100は192.168.3.1です。

```
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
```

次に、送信元へのRPFネイバーを示します。10.0.0.2は192.168.3.2です。

```
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

```

送信元をアクティブ化する前に、グループ239.1.1.1にすでに(*,G)が存在するため、R3のmrouteテーブルを確認します。これは、LHRに接続されたレシーバが指定されたグループに対して既に要求しているためです。

```

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

```

ここで、R3インターフェイスGi1の送信元パケットとキャプチャパケットをアクティブにします。

The screenshot displays a Wireshark capture of network traffic on interface Gi1. The packet list pane shows the following entries:

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 packet details pane for packet 11 shows the following structure:

- 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

このパケットキャプチャからわかるように、PIMアサートのパケットがすでに存在しています。

Frame 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
```

Frame 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
```

これらのパケットを調べると、アサートの勝者は誰かを判断できるはずですが、次に、PIMアサートフォワーダの選択について説明します。

メトリックのプリファレンスは、アドミニストレーティブディスタンス(AD)です。これは、ルーティングテーブルにルートを実インストールするルーティングプロトコルのアドミニストレーティブディスタンスを指します。このディスタンスは送信元IPアドレスの検索に使用され、メトリックはルートのコストです。

アサートの勝者を決定するために使用される他の属性もあります。これらの詳細は、RFC 7761で確認できます。

RFC 7761 セクション 4.6.3 からの抜粋.

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.

これらのフィールドを定義し、パスを選択することで、このシナリオでアサートの勝者を決定できます。アサートパケットをもう一度見ると、`rpt_bit_flag`という最初の選択基準に基づいてメトリック設定が決定されるため、メトリック設定が比較されないことがわかります。

このシナリオでは、R1とR2を比較します。両方のルータは、以前に見られたアサートメッセージを送信し、両方のデバイスが互いのアサートメッセージを見た後に、勝者を決定するために互いのメトリックを比較できます。

R2はRPツリーでアサートメッセージを送信するため：Falseの値は0で、R1がRPツリーで送信した値よりも実際に低くなります。Trueの場合、値は1になります。RPツリービットは0または1に設定されます。

1に設定されている場合のRPツリービットは、現在の共有ツリーであることを意味します。RPTビットがクリアされている場合は、アサートの送信元がインターフェイスで(S,G)フォワーディング状態を持っていることを示します。

(S,G)アサートが(*,G)アサートよりも優先されるため、R2がアサートの勝者である必要があります。「I am Assert Winner」状態に移行します。RFC 7761で前述したように、低い値が優先されます。

R1とR2の両方を見て、アサートの勝者が誰かを確認してみましょう。

```
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
```

この出力では、R2の(S,G)がアサートの勝者であることを示すAフラグがOILに設定されていることがわかります。ここで、R1には(S,G)にOILが設定されておらず、Pフラグが設定されている場合、特定の(S,G)がプルーニングされています。アサートの勝者じゃない。

注：アサートが共有セグメントに存在する場合、ダウンストリームのネイバーはJoin(*,G)お

よびJoin(S,G)の定期的なメッセージを適切なRPFネイバー (アサートプロセスによって変更されたRPFネイバー) に送信します。MRIBによって示されるように、RPFネイバーに必ずしも送信されるわけではありません。

```
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
```

R1とR2の両方のRPツリービットが1に設定されている場合は、最も低いADのルータを検討できません。等しい場合は、メトリックを確認します。両方のルータでRPツリービットがtrueの場合、メトリックはRP IPアドレスに対して比較されます。RPツリービットが0の場合、メトリックはマルチキャストストリームのソースに対して比較されます。

これらの値がすべて同じ場合、最も大きいIPアドレスソーシングアサートメッセージが勝者になります。

要約

シナリオ1では、アサートパケットは観察されていませんでしたが、RFCによるとトリガーされるはずでした。前述したように、これは、(S,G)のコントロールプレーンが構築される前にR3がプルニング(*,G)していたためです。

シナリオ2では、アサートパケットが表示されます。最初のパケットがLHRで受信されると、R3に(S,G)join/pruneを送信して、送信元/グループをプルします。その後、R3は同じ送信元/グループのR2に対してjoin/pruneパケットを送信します。これにより、R1とR2の両方に有効なOILが入力されます。R3は、TフラグがR3s(S,G)状態に設定されている場合、RPビットが設定されたプルニング(S,G)のみを実行します。そのためには、共有セグメントから別のデータプレーンパケットを受信する必要があります。コントロールプレーンはすでに(S,G)用に構築されているため、アサートメッセージをトリガーする共有セグメント上で重複が発生します。