Entender o mecanismo PIM Assert

Contents

Introduction

Este documento descreve o mecanismo de asserção do Protocol Independent Multicast (PIM), concentra-se nos critérios vencedores da asserção PIM e mergulha mais profundamente em determinados casos de canto.

Prerequisites

Requirements

A Cisco recomenda que você tenha conhecimento do mecanismo de declaração PIM.

Componentes Utilizados

As informações neste documento são baseadas no Cisco CSR1000V versão 16.4.1

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

O que é o mecanismo PIM Assert?

Quando há vários roteadores habilitados para PIM em um segmento compartilhado, é possível que esses roteadores encontrem tráfego multicast duplicado. Esse pode ser o caso porque dois ou mais roteadores no mesmo segmento compartilhado podem ter uma entrada válida (S,G) ou (*,G) que preenche a interface de saída para o segmento compartilhado para o mesmo IP de origem/grupo de destino.

O mecanismo PIM assert é usado para detectar e eliminar a duplicação de tráfego multicast em um segmento compartilhado. É importante observar que esse mecanismo não evita a duplicação de ocorrida, em vez disso, ele usa a duplicação do tráfego multicast como um disparador para

ativar esse mecanismo que escolhe um único encaminhador para esse fluxo.

Quando você tem a duplicação de tráfego multicast em um segmento compartilhado, pode supor que há vários roteadores que enviam o mesmo (S,G) ou (*,G) em um segmento compartilhado. Se você escolher um roteador para encaminhar esse fluxo de forma eficaz, ele eliminará a duplicação.

O PIM aproveita as mensagens de declaração PIM que são disparadas quando você recebe um pacote multicast na Lista de Interface de Saída (OIL). Essas mensagens assertivas contêm métricas que são usadas para calcular quem será o vencedor assertivo. Os roteadores de downstream na LAN também recebem mensagens de asserção PIM. Essas mensagens são então usadas por dispositivos downstream para enviar mensagens Join/Prune apropriadas ao roteador upstream que ganhou a eleição assert.

Cenário 1. Razão da LHR

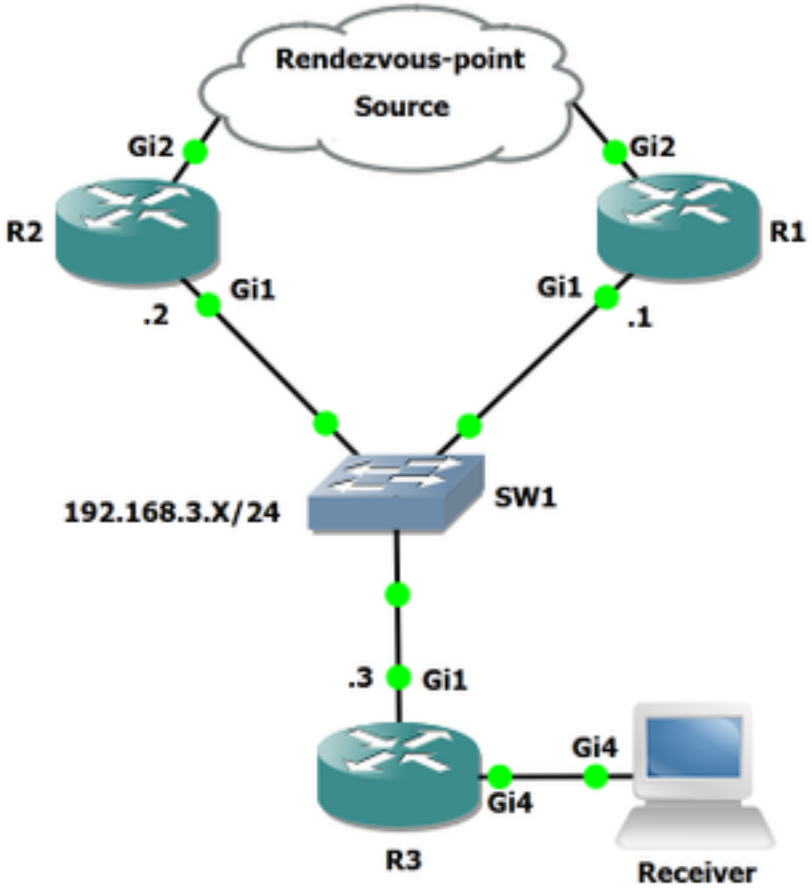


Figura 1.

No diagrama de rede, R3 é o LHR (Last Hop Router), R3 se conecta a R2 e R1 por meio de um segmento compartilhado.

Quando você recebe um relatório do Internet Group Management Protocol (IGMP) do receptor, o R3 verifica quem é o vizinho RPF em direção ao RP. Na topologia, R1 é o vizinho RPF em direção ao RP, portanto, R3 envia uma união (*,G) em direção ao R1. Quando R1 puxa o fluxo (suponha que o grupo esteja ativo), R3 envia uma junção (S,G) para a origem e puxa a árvore de origem para baixo. R2 é o vizinho RPF em direção à árvore de origem, o que significa que R3 enviará a junção (S,G) em direção a R2. R3 tem a mesma interface de RPF para RP e origem. Aqui você pode ver a tabela mroute de R3 para o grupo 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
```

Como você pode ver em R3, o vizinho RPF (*,G) é 192.168.3.1 e o vizinho RPF em direção ao (S,G) é 192.168.3.2. Agora, **isso deve resultar em R1 e R2 para ter um OIL válido em direção a R3**. Vamos ver essas entradas:

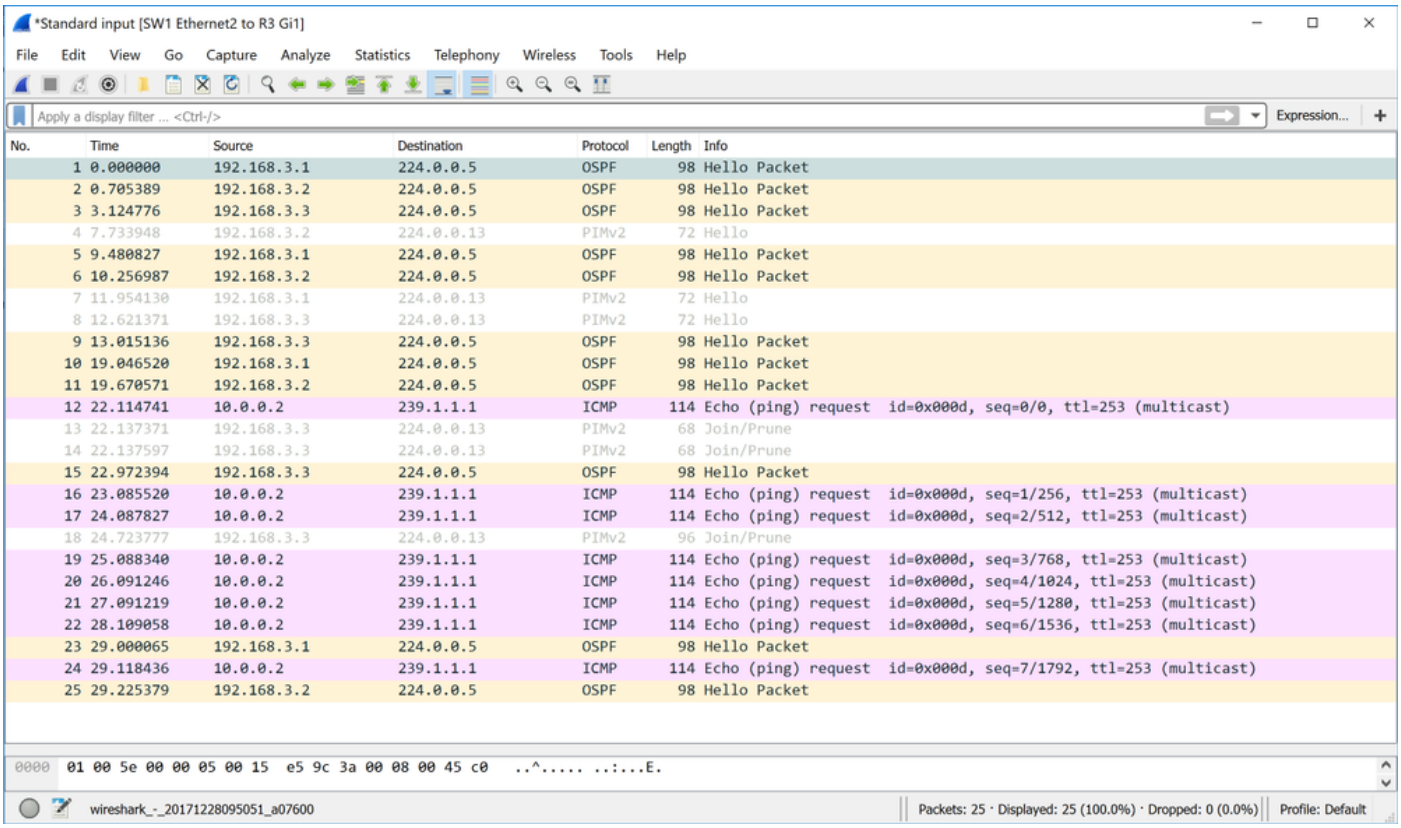
```
R1#show ip mroute
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 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
```

Como mencionado, a asserção pode ser disparada quando há dois roteadores upstream que têm um OIL válido preenchido em um segmento compartilhado. Como R1 e R2 têm um OIL válido, verifique se há um mecanismo assert na captura de pacotes.

Essa captura de pacote foi capturada na interface Gi1 de R3 em direção a SW1.



Nesta captura de pacote, você não vê nenhum pacote assert, mesmo que haja todos os pré-requisitos para criar duplicação no segmento compartilhado entre R1, R2 e R3. Por que você não vê nenhum pacote PIM assert quando o fluxo (S,G) foi ativado?

Parece que o RFC 7761 pode ter a resposta para essas perguntas.

Resumo do RFC 7761 Seção 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.
```

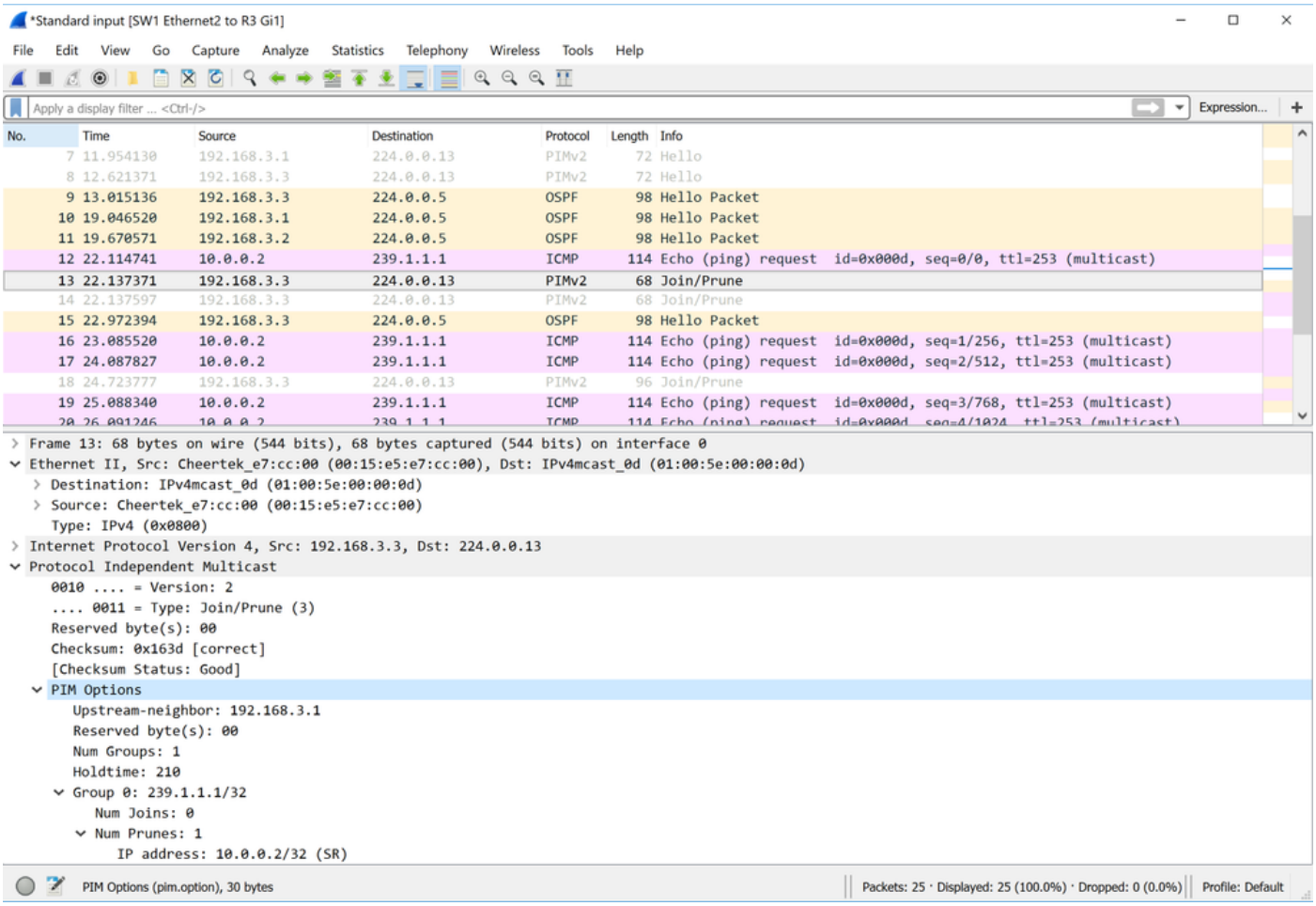
```
   4.  RPF'(S,G) == RPF'(*,G).  In this case, the router will never be
       able to tell if the SPT has been completed, so it should just
       switch immediately.  The RPF'(S,G) != NULL check ensures that the
       SPTbit is set only if the RPF neighbor towards S is valid.

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.
```

O (S,G) SPTbit é usado para distinguir se deve encaminhar no estado (*,G) ou no estado (S,G). Quando você muda da árvore RP para a árvore de origem, há um período de transição quando os dados chegam devido ao estado upstream (*,G) enquanto o estado upstream (S,G) está estabelecido, nesse momento, o roteador deve continuar encaminhando somente no estado (*,G). Isso evita buracos negros temporários que seriam causados pelo envio de uma(S,G,rpt) ameça antes que o estado upstream (S,G) tenha terminado de ser estabelecido.

Embora pareça que o cenário pode correlacionar-se com o último ponto mencionado acima. No caso de a interface RPF ser a mesma para o RP e para o S,
mas RPF'(S,G) e RPF'(*,G) diferem, esperamos por um Assert(S,G), que indica que o roteador upstream com estado (S,G) acredita que o SPT foi concluído.

Para que a asserção seja disparada, o roteador deve receber um pacote duplicado em seu OIL já preenchido para o mesmo IP de origem/grupo de destino no segmento. R3 também é um LHR, o que significa que ele é designado para mudar de (*,G) para SPT (S,G) quando um pacote é recebido de (*,G).

Na captura de pacotes, observamos que nenhuma asserção é disparada. Apesar de vermos uma ameixa enviada imediatamente após o primeiro eco ICMP ser recebido.

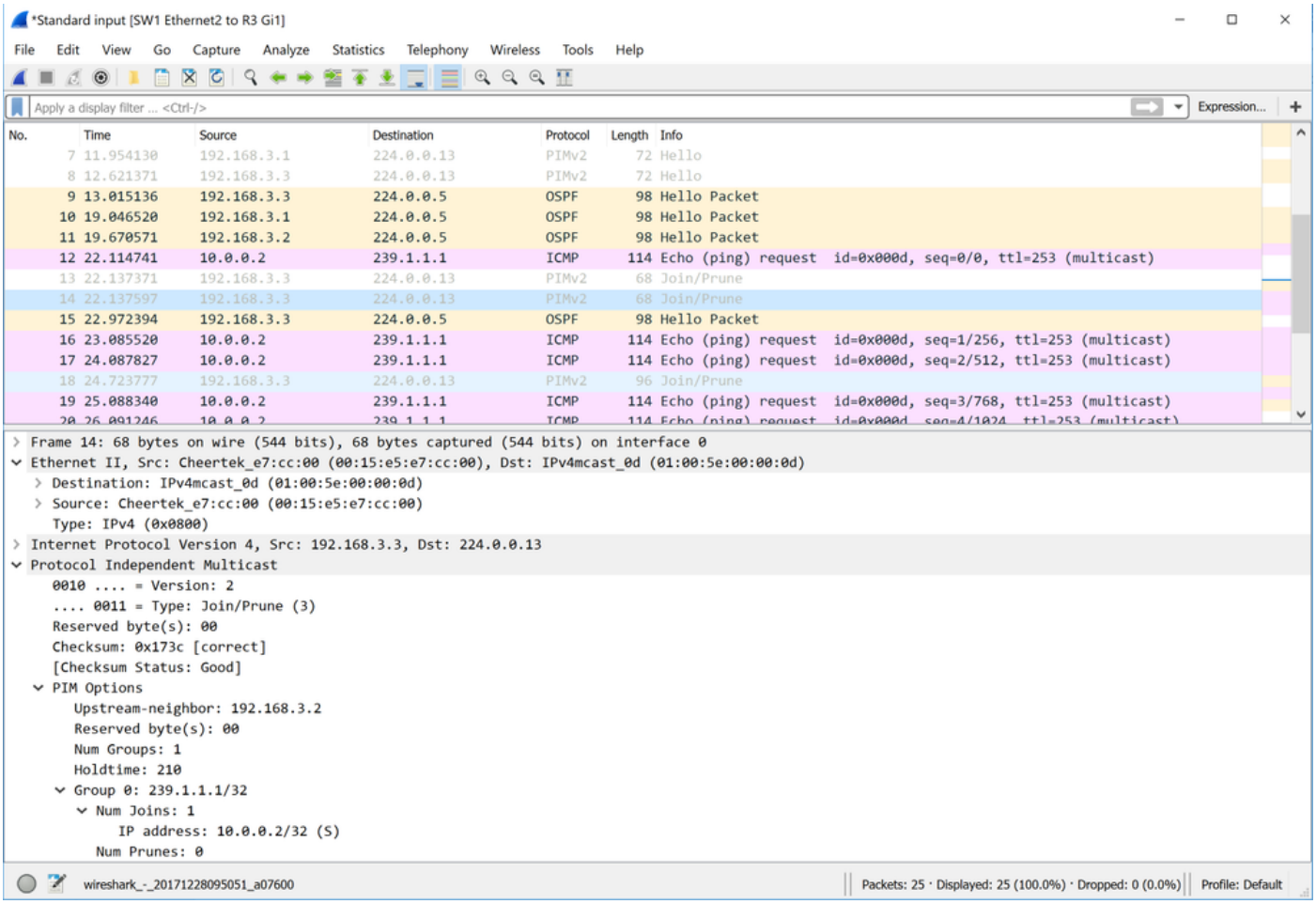


Como você pode ver, uma vez que o primeiro pacote de solicitação do Internet Control Message Protocol (ICMP) é recebido na interface G1 de R3, um (*,G) SR-bit prune é enviado para o vizinho upstream 192.168.3.1. Esta poda (*,G) para a fonte específica definida.

Você pode ver estes sinalizadores também definidos: (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.
```

No segundo pacote PIM nº 14, você pode ver que R3 tenta ingressar na árvore (S,G).



Observe-se que, assim que o primeiro plano de dados é recebido, o pacote R3 elimina o (*,G) e cria o (S,G). Essa é a razão pela qual você não vê pacotes PIM assert. Esse determinado cenário está em vigor quando você tem um LHR que tem a mesma interface RPF para (S,G) e (*,G). Embora esse comportamento possa ser um pouco diferente do RFC 7761, ele não deve causar nenhum problema.

Agora, vamos continuar com o cenário 2. O diagrama desse cenário pode ser visto aqui:

Cenário 2. Seleção de caminho de asserção

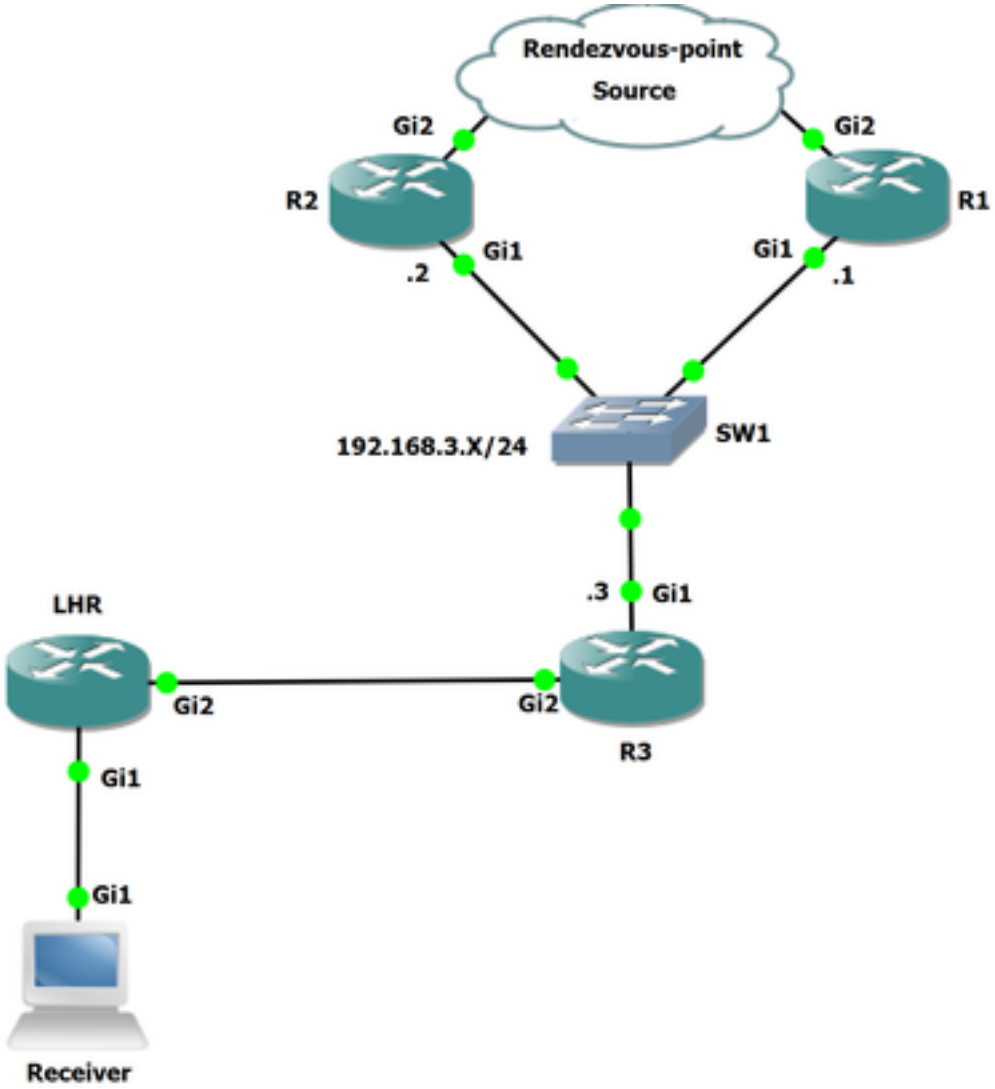


Figura 2.

Nessa topologia, há outro roteador conectado em R3 que é o LHR. O LHR se conecta diretamente ao receptor. A origem e o RP estão acima de R2 e R1. O vizinho RPF em R3 em direção ao RP é R1 e o vizinho RPF em direção à origem é R2.

Vamos verificar o vizinho RPF tanto para a origem quanto para o RP.

Aqui você vê o vizinho RPF em direção ao RP: 192.168.0.100 é 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
```

Aqui você vê o vizinho RPF em direção à Origem: 10.0.0.2 é 192.168.3.2.

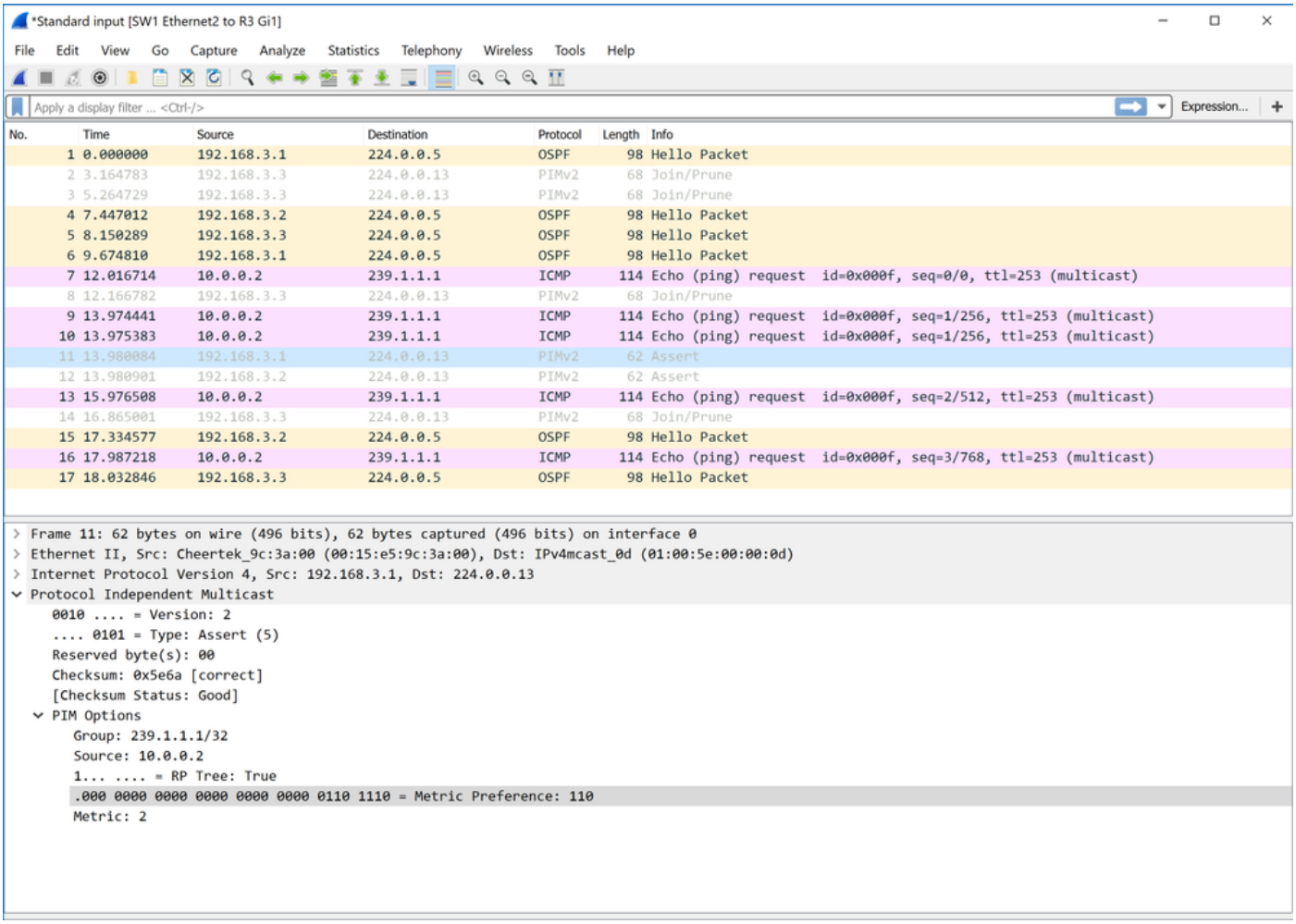
```
R3#show ip rpf 10.0.0.2
RPF information for ? (10.0.0.2)
  RPF interface: GigabitEthernet1
```

```
  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
```

Antes de ativarmos a origem, vamos observar a tabela mroute em R3, como você pode ver que já existe (*,G) para o grupo 239.1.1.1. Isso ocorre porque o receptor conectado ao LHR já solicitou para o grupo especificado.

```
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
```

Agora, ative a origem e capture os pacotes na interface Gi1 de R3.



Como você pode ver nesta captura de pacote, o PIM afirma que os pacotes já estão presentes.

Quadro 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
```

Quadro 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
```

Ao examinar esses pacotes, você deve ser capaz de determinar quem é o vencedor da asserção. Agora, vamos dar uma olhada na seleção do PIM assert forwarding.

A preferência da métrica é a Distância Administrativa (AD). Isso se refere à distância administrativa do protocolo de roteamento que instala a rota na tabela de roteamento, que é usada para pesquisar o endereço IP origem e a métrica é o custo da rota.

Há também outros atributos que são usados para determinar quem é o vencedor da asserção. Você pode ver esses detalhes no RFC 7761.

Resumo do RFC 7761 Seção 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.
```

Com o uso desses campos definidos e a seleção do caminho, você pode determinar quem será o vencedor da asserção neste cenário. Se você observar os pacotes assert novamente, poderá ver que a preferência métrica não é comparada, já que a decisão é tomada no primeiro critério de seleção, que é rpt_bit_flag.

Neste cenário, comparar R1 e R2 é comparado. Ambos os roteadores enviam mensagens assertivas que foram vistas anteriormente e quando ambos os dispositivos veem as mensagens assertivas um do outro, eles podem comparar métricas entre si para determinar quem é o vencedor.

Como R2 envia uma mensagem assert com a árvore RP: Falso que tem um valor 0, é realmente inferior ao que R1 enviou com uma árvore RP: Verdadeiro que tem um valor de 1. O bit de árvore RP é definido como 0 ou 1.

bit de árvore RP quando definido como 1 significa que você está atualmente na árvore compartilhada; o bit RPT limpo indica que o remetente da asserção tinha o estado de encaminhamento (S,G) em uma interface.

Como (S,G) afirmações têm prioridade sobre (*,G) afirmações, R2 deve ser o vencedor da asserção. Transição para o estado "Sou Vencedor de Asserto". Como mencionado na instrução anterior no RFC 7761, o valor mais baixo é preferido.

Vamos ver o R1 e o R2 para ver quem é o vencedor da asserção.

```
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
```

Nessa saída, você pode ver que o (S,G) em R2 tem o sinalizador A definido no OIL, o que indica que ele é o vencedor do assert. Aqui em R1, você não tem um OIL no (S,G) e o flag P está definido, o que significa que o particular (S,G) foi removido neste caso: não é o vencedor do acerto.

> **Note**: Quando a asserção está presente em um segmento compartilhado, os vizinhos downstream enviam mensagens periódicas Join(*,G) e Join(S,G) para o vizinho RPF apropriado, isto é, o vizinho RPF conforme modificado pelo processo de asserção. Nem sempre são enviados para o vizinho RPF conforme indicado pelo MRIB.

```
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
```

Se for o caso de que R1 e R2 têm o bit de árvore RP definido como 1. você pode considerar o roteador com o AD mais baixo; se igual, observe a métrica. Se o bit de árvore RP for verdadeiro em ambos os roteadores, a métrica será comparada ao endereço IP RP. Se o bit de árvore RP for 0, a métrica será comparada à origem do fluxo multicast.

Se todos esses valores forem os mesmos, a mensagem assert de origem de endereço IP mais alta será a vencedora.

Summary

No cenário um, você não observou pacotes assert, no entanto, por RFC eles devem ter sido disparados. Como mencionado, foi porque R3 estava podando (*,G) antes de o plano de controle para (S,G) ser construído.

Enquanto no cenário dois, você vê pacotes assert. Quando o primeiro pacote foi recebido no LHR, ele envia uma junção/remoção (S,G) em direção a R3 para puxar a origem/o grupo. Em seguida, R3 enviará um pacote de junção/remoção para R2 para a mesma origem/grupo. Isso faria com que R1 e R2 tivessem OILs válidos preenchidos. Agora, R3 apenas puxa (S,G) com bit RP definido quando o flag T é preenchido no estado R3s (S,G). Para que isso ocorra, você precisa receber outro pacote de plano de dados do segmento compartilhado. Como o plano de controle já foi criado para (S,G), isso leva à duplicação no segmento compartilhado, disparando mensagens assert.