Réflexion du service de multidiffusion à l'aide de PIM-SM sur IOS-XE : Multidiffusion vers monodiffusion

Contenu

[Introduction](#)
[Conditions préalables](#)
[Conditions requises](#)
[Components Used](#)
[Configuration](#)
[Diagramme du réseau](#)
[Configurations](#)
[Vérification](#)

Introduction

L'objectif de cet article est de vous donner une idée du fonctionnement de base de MSR (Multicast Service Replication) à l'aide de plates-formes IOS-XE, sous la forme d'un guide de travaux pratiques de configuration.

Conditions préalables

Conditions requises

Compréhension de base de PIM-SM

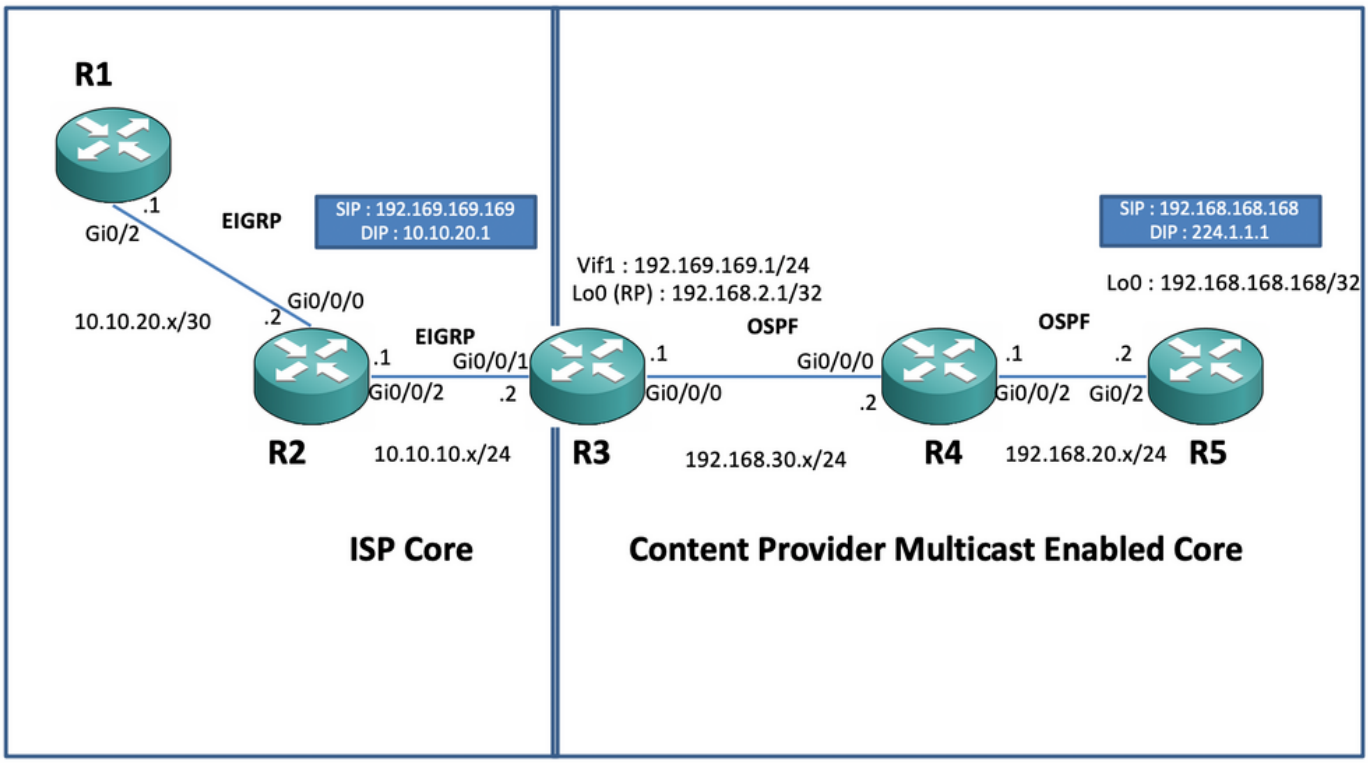
Components Used

ASR1000 (R2&R4), ISR4300 (R3), ISR2900 (R1&R5)

Configuration

Nous afficherions les configurations de bout en bout en fonction du scénario schématique ci-dessous pour la traduction de la multidiffusion.

Diagramme du réseau



Configurations

Dans le schéma ci-dessus, le noeud R1 agit en tant que récepteur qui est censé obtenir uniquement le flux de données de multidiffusion monodiffusion de la source de multidiffusion.

Le noeud R5 agit en tant que source de multidiffusion qui génère du trafic ICMP multicast provenant de son interface de bouclage 0.

Le noeud R2 se trouve sous le domaine principal de multidiffusion des fournisseurs de contenu et exécute PIM-SM avec la sous-couche du protocole OSPF.

Le noeud R3 agit comme le routeur qui exécute l'application de réplication de service multidiffusion et est dans ce cas le routeur périphérique multidiffusion à partir duquel le trafic de données multidiffusion est censé être traduit en un paquet de données monodiffusion vers le récepteur. Il utilise respectivement OSPF et EIGRP avec le fournisseur de contenu et le FAI et héberge le RP (point Rendezvouz) sur son interface de bouclage dans le domaine principal de multidiffusion.

Le noeud R4 est sous le contrôle de coeur de réseau du FAI et n'est pas activé pour la multidiffusion. Il comprend uniquement comment atteindre le noeud R3 à l'aide du routage EIGRP sous-jacent.

Vous trouverez ci-dessous les configurations pertinentes présentes sur les noeuds figurant dans le schéma de topologie ci-dessus :

R1:

```
! no ip domain lookup ip cef no ipv6 cef ! interface GigabitEthernet0/2 ip address 10.10.20.1
255.255.255.0 duplex auto speed auto end ! router eigrp 100 network 10.10.20.0 0.0.0.255 !
```

R2:

```
! interface GigabitEthernet0/0/0 ip address 10.10.20.2 255.255.255.0 negotiation auto ! interface GigabitEthernet0/0/2 ip address 10.10.10.1 255.255.255.0 negotiation auto ! router eigrp 100 network 10.10.10.0 0.0.0.255 network 10.10.20.0 0.0.0.255 !
```

R3:

```
! ip multicast-routing distributed ! interface Loopback0 ip address 192.168.2.1 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet0/0/0 ip address 192.168.30.1 255.255.255.0 ip pim sparse-mode ip ospf 1 area 0 negotiation auto ! interface GigabitEthernet0/0/1 ip address 10.10.10.2 255.255.255.0 negotiation auto ! interface Vif1 ip address 192.169.169.1 255.255.255.0 ip pim sparse-mode ip service reflect GigabitEthernet0/0/0 destination 224.1.1.0 to 10.10.20.0 mask-len 24 source 192.169.169.169 <<<< ip igmp static-group 224.1.1.1 ip ospf 1 area 0 ! router eigrp 100 network 10.10.10.0 0.0.0.255 ! router ospf 1 ! ip pim rp-address 192.168.2.1 !
```

R4:

```
! ip multicast-routing distributed ! interface GigabitEthernet0/0/0 ip address 192.168.30.2 255.255.255.0 ip pim sparse-mode ip ospf 1 area 0 negotiation auto ! interface GigabitEthernet0/0/2 ip address 192.168.20.1 255.255.255.0 ip pim sparse-mode ip ospf 1 area 0 negotiation auto ! router ospf 1 ! ip pim rp-address 192.168.2.1 !
```

R5:

```
! ip multicast-routing ip cef no ipv6 cef ! interface Loopback0 ip address 192.168.168.168 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet0/2 ip address 192.168.20.2 255.255.255.0 ip pim sparse-mode ip ospf 1 area 0 duplex auto speed auto ! router ospf 1 ! ip pim rp-address 192.168.2.1 !
```

Vérification

Nous pouvons valider les configurations en exécutant un test ping pour simuler le trafic de multidiffusion à partir du routeur R5 avec une source de son interface de bouclage 0 [192.168.168.168] destinée à l'adresse de multidiffusion 224.1.1.1. Vérifiez ensuite les entrées mroute sur le noeud qui exécute l'application MSR, c'est-à-dire R3 :

```
R5(config)#do ping 224.1.1.1 sou lo 0 rep 10000000 Type escape sequence to abort. Sending 10000000, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds: Packet sent with a source address of 192.168.168.168 ..........................

R3#sh ip mroute 224.1.1.1 IP Multicast Routing Table Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected, L - Local, P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet, X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement, U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel, z - MDT-data group sender, Y - Joined MDT-data group, y - Sending to MDT-data group, G - Received BGP C-Mroute, g - Sent BGP C-Mroute, N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed, Q - Received BGP S-A Route, q - Sent BGP S-A Route, V - RD & Vector, v - Vector, p - PIM Joins on route, x - VxLAN group, c - PFP-SA cache created entry Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join Timers: Uptime/Expires Interface state: Interface, Next-Hop or VCD, State/Mode (*, 224.1.1.1), 00:47:41/stopped, RP 192.168.2.1, flags: SJC Incoming interface: Null, RPF nbr 0.0.0.0 Outgoing interface list: Vif1, Forward/Sparse, 00:46:36/00:01:23 <<<< (192.168.168.168, 224.1.1.1), 00:00:20/00:02:43, flags: T Incoming interface: GigabitEthernet0/0/0, RPF nbr 192.168.30.2 Outgoing interface list: Vif1, Forward/Sparse, 00:00:20/00:02:39 <<<<

R3#sh ip mroute 224.1.1.1 count Use "show ip mfib count" to get better response time for a large number of mroutes. IP Multicast Statistics 3 routes using 2938 bytes of memory 2 groups, 0.50
```

```
average sources per group Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc) Group: 224.1.1.1, Source count: 1, Packets forwarded: 1455, Packets received: 1458 <<<< RP-tree: Forwarding: 1/0/100/0, Other: 1/0/0 Source: 192.168.168.168/32, Forwarding: 1454/1/113/0, Other: 1457/3/0 R3#sh ip mroute 224.1.1.1 count Use "show ip mfib count" to get better response time for a large number of mroutes. IP Multicast Statistics 3 routes using 2938 bytes of memory 2 groups, 0.50 average sources per group Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc) Group: 224.1.1.1, Source count: 1, Packets forwarded: 1465, Packets received: 1468 <<<< RP-tree: Forwarding: 1/0/100/0, Other: 1/0/0 Source: 192.168.168.168/32, Forwarding: 1464/1/113/0, Other: 1467/3/0
```

Vous pouvez également prendre des captures pour vérifier que les paquets sont effectivement traduits en adresse de destination de monodiffusion sur le noeud R2 à l'aide de la fonction EPC (Embedded Packet Capture) sur le routeur IOS-XE :

```
R2#mon cap TAC int gi 0/0/2 both match any R2#mon cap TAC buff siz 50 circular R2#mon cap TAC start Started capture point : TAC R2# *Aug 12 06:50:40.195: %BUFCAP-6-ENABLE: Capture Point TAC enabled. R2#sh mon cap TAC buff br | i ICMP 6 114 10.684022 192.169.169.169 -> 10.10.20.1 0 BE ICMP <<<< 7 114 10.684022 192.169.169.169 -> 10.10.20.1 0 BE ICMP <<<< 8 114 12.683015 192.169.169.169 -> 10.10.20.1 0 BE ICMP <<<< 9 114 12.683015 192.169.169.169 -> 10.10.20.1 0 BE ICMP <<<<
```

Ici, le point important à noter est que régulièrement lorsque vous exécutez des requêtes ping ICMP multicast dans des « environnements de travaux pratiques », vous vous attendez généralement à recevoir les paquets de réponse d'écho ICMP du côté récepteur vers la source, en supposant qu'il y ait une accessibilité complète entre les deux (source et récepteur). Cependant, dans ce scénario, il est important de noter que même si nous essayons d'annoncer l'adresse source NATted pour les paquets ICMP de multidiffusion, c'est-à-dire 192.169.169.169 jusqu'au destinataire, c'est-à-dire R1 via EIGRP, les réponses d'écho ICMP de monodiffusion ne traversent pas le routeur R3, puisque la NAT inverse non configuré sur le noeud d'application MSR. Nous pouvons le tester, en essayant d'exécuter l'annonce de route EIGRP de l'interface Vif 1 sur R3 dans EIGRP (routage de coeur de réseau ISP) :

```
ISR4351(config)#router eigrp 100 ISR4351(config-router)#network 192.169.169.0 0.0.0.255 <<<<
```

Maintenant, nous pouvons vérifier les captures prises sur le noeud R2 sur les réponses d'écho ICMP envoyées vers R3 :

```
R2#sh mon cap TAC buff br | i ICMP
```

Mais les requêtes ping échoueraient toujours, comme indiqué sur la source R5 :

```
R5(config)#do ping 224.1.1.1 sou lo 0 rep 10000000 Type escape sequence to abort. Sending 10000000, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds: Packet sent with a source address of 192.168.168.168
.....................................................................
.....................................................................
```

Maintenant, pour obtenir les réponses jusqu'à la source, nous pouvons configurer le transfert de port NAT sur le noeud d'application MSR R3 pour traduire le trafic destiné vers 192.169.169.169 à 192.168.168.168, en configurant NAT extensible :

```
R3(config)#int gi 0/0/1 R3(config-if)#ip nat out R3(config-if)#int gi 0/0/0 R3(config-if)#ip nat ins R3(config-if)#exit R3(config)#ip nat inside source static 192.168.168.168 192.169.169.169 extendable <<<<
```

Àprésent, lors de la vérification du noeud R5 source, nous pouvons voir la réponse revenir :

```
R5(config)#do ping 224.1.1.1 sou lo 0 rep 10000000 Type escape sequence to abort. Sending
10000000, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds: Packet sent with a source
address of 192.168.168.168
.............................................................
```

Les éléments ci-dessus viennent d'être exécutés pour expliquer le flux de paquets et comprendre comment établir le chemin/flux de monodiffusion inverse pour le trafic de données et le trafic de multidiffusion en aval. Puisque dans le scénario de production classique, vous ne rencontreriez généralement pas de cas ou d'instances où les applications de multidiffusion exécutées côté serveur/source nécessitent un accusé de réception inverse des paquets des récepteurs sous forme de monodiffusion.

Par les tests et validations ci-dessus, il aurait dû donner un bref aperçu sur la façon d'exécuter l'application de réplication de service de multidiffusion sur l'un des noeuds de périphérie de multidiffusion et sur la façon de déployer le même si le même exemple présenté ci-dessus devait être étendu à un déploiement à grande échelle.