Examiner les études de cas sur le protocole de passerelle frontalière

Table des matières

- Introduction
- Conditions préalables
  - Exigences
  - Composants utilisés
  - Conventions
- Études de cas BGP 1
  - Fonctionnement de BGP
  - eBGP et iBGP
  - Activer le routage BGP
  - Former des voisins BGP
  - BGP et interfaces de bouclage
  - Saut multiple eBGP
  - Saut multiple eBGP (équilibrage de charge)
  - Mise en correspondance de route
  - Commandes de configuration match et set
    - Exemple 1
    - Exemple 2
  - Commande network
  - Redistribution
  - Routes statiques et redistribution
  - iBGP
  - Algorithme de décision BGP
- Études de cas BGP 2
  - Attribut AS_PATH
  - Attribut origin
  - Attribut BGP next hop
    - Prochain saut BGP (réseaux multi-accès)
    - Prochain saut BGP (NBMA)
    - Commande next-hop-self
  - Porte dérobée BGP
  - Synchronization
    - Désactiver la synchronisation
  - Attribut weight
  - Attribut local preference
  - Attribut metric
  - Attribut community
- Études de cas BGP 3

- Filtre BGP
  - Filtre de routage
  - Filtre de chemin
- Expression régulière AS
  - Filtre de communauté BGP
- Voisins BGP et mises en correspondance de route
  - Utilisation de la commande set as-path prepend
  - Groupes d'homologues BGP

Études de cas BGP 4

- CIDR et adresses agrégées
  - Commandes d'agrégat
  - Exemple CIDR 1
  - Exemple CIDR 2 (as-set)
- Confédération BGP
- Réflecteurs de route
  - Plusieurs RR dans un cluster
  - RR et speakers BGP conventionnels
  - Éviter la boucle des informations de routage
- Atténuation de la déflexion de route
- Comment BGP sélectionne un chemin

Études de cas BGP 5

- Exemple de projet pratique

Informations connexes

Introduction

Ce document présente cinq études de cas concernant le protocole BGP (Border Gateway Protocol).

Conditions préalables

Exigences

Aucune exigence spécifique n'est associée à ce document.

Composants utilisés

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

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. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à Conventions relatives aux conseils techniques Cisco.

Études de cas BGP 1

Le protocole BGP , défini par RFC 1771 , vous permet de créer un routage interdomaine sans boucle entre des systèmes autonomes (AS). Un AS est un ensemble de routeurs nécessitant une administration technique simple. Les routeurs d'un AS peuvent employer plusieurs protocoles Interior Gateway Protocols (IGP) pour échanger des informations de routage au sein de l'AS. Les routeurs peuvent utiliser un protocole Exterior Gateway Protocol (EGP) pour router les paquets en dehors de l'AS.

Fonctionnement de BGP

BGP utilise TCP comme protocole de transport, sur le port 179. Deux routeurs BGP forment une connexion TCP entre eux. Ces routeurs sont des routeurs homologues. Les routeurs homologues échangent des messages pour ouvrir et confirmer les paramètres de connexion.

Les routeurs BGP échangent des informations sur l'accessibilité du réseau. Ces informations constituent principalement une indication des chemins d'accès complets qu'une route doit emprunter pour atteindre le réseau de destination. Les chemins sont des numéros d'AS BGP. Cette information aide à la construction d'un graphique des AS sans boucle. Le graphique montre également à quel niveau appliquer des règles de routage afin d'imposer quelques restrictions au comportement de routage.

Deux routeurs qui forment une connexion TCP pour échanger des informations de routage BGP sont des « homologues » ou des « voisins ». Les homologues BGP échangent initialement l'intégralité des tables de routage BGP. Après cet échange, les homologues envoient des mises à jour incrémentielles lorsque la table de routage change. BGP conserve un numéro de version de la table BGP. Le numéro de version est identique pour tous les homologues BGP. Le numéro de version change à chaque fois que BGP met à jour la table pour refléter les modifications des informations de routage. L'envoi des paquets keepalive garantit que la connexion entre les homologues BGP est active. Les paquets de notification sortent en réponse aux erreurs ou aux conditions spéciales.

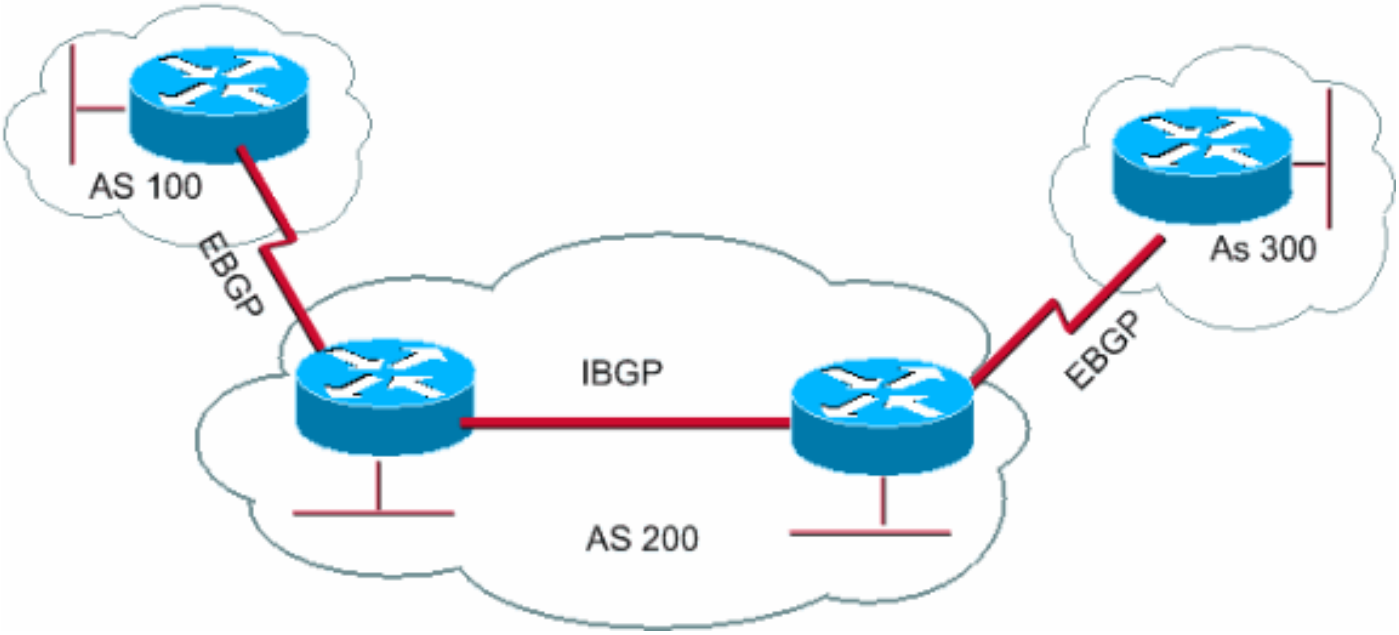
eBGP et iBGP

Si un AS comporte plusieurs speakers BGP, il peut servir de service de transit pour d'autres AS. Comme le montre le schéma suivant dans cette section, AS200 est un AS de transit pour AS100 et AS300.

Afin d'envoyer l'information aux AS externes, l'accessibilité des réseaux doit être garantie. Afin d'assurer l'accessibilité des réseaux, les processus suivants sont exécutés :

- Interconnexion BGP interne (iBGP) entre les routeurs au sein d'un AS
- Redistribution des informations BGP aux IGP qui s'exécutent dans l'AS

Quand BGP s'exécute entre des routeurs qui appartiennent à deux AS différents, on parle de BGP extérieur (eBGP). Quand BGP s'exécute entre des routeurs du même AS, on parle d'iBGP.



BGP s’exécute entre les routeurs dans le même AS

Activer le routage BGP

Complétez ces étapes afin d'activer et de configurer BGP.

Supposons que vous vouliez que deux routeurs, RTA et RTB, communiquent via BGP. Dans le premier exemple, RTA et RTB sont dans des AS différents. Dans le deuxième exemple, les deux routeurs appartiennent au même AS.

1. Définissez le processus de routage et le numéro de l'AS auquel les routeurs appartiennent.

   Émettez la commande suivante pour activer BGP sur un routeur :

   <#root>

   ```
   router bgp <autonomous-system>
   ```

   ```
   RTA#
   router bgp 100

   RTB#
   router bgp 200
   ```

   Ces instructions indiquent que RTA exécute BGP et appartient à AS100. RTB exécute BGP et appartient à AS200.

2. Définissez les voisins BGP.

   La formation de voisins BGP indique les routeurs qui essaient de communiquer via BGP. La section suivante explique ce processus.

Former des voisins BGP

Deux routeurs BGP deviennent voisins après avoir établi une connexion TCP entre eux. La connexion TCP est essentielle pour que les deux routeurs homologues commencent à échanger des mises à jour de routage.

Une fois la connexion TCP établie, les routeurs envoient des messages d'ouverture pour échanger des valeurs. Les valeurs que les routeurs s'échangent sont le numéro d'AS, la version de BGP exécutée par les routeurs, l'ID des routeurs BGP et le temps de maintien de keepalive. Après la confirmation et l'acceptation de ces valeurs, l'établissement de la connexion de voisinage s'effectue. N'importe quel état autre qu'Established indique que les deux routeurs ne sont pas devenus voisins et qu'ils ne peuvent pas échanger de mises à jour BGP.

Émettez cette neighbor commande pour établir une connexion TCP :

```
<#root>
```

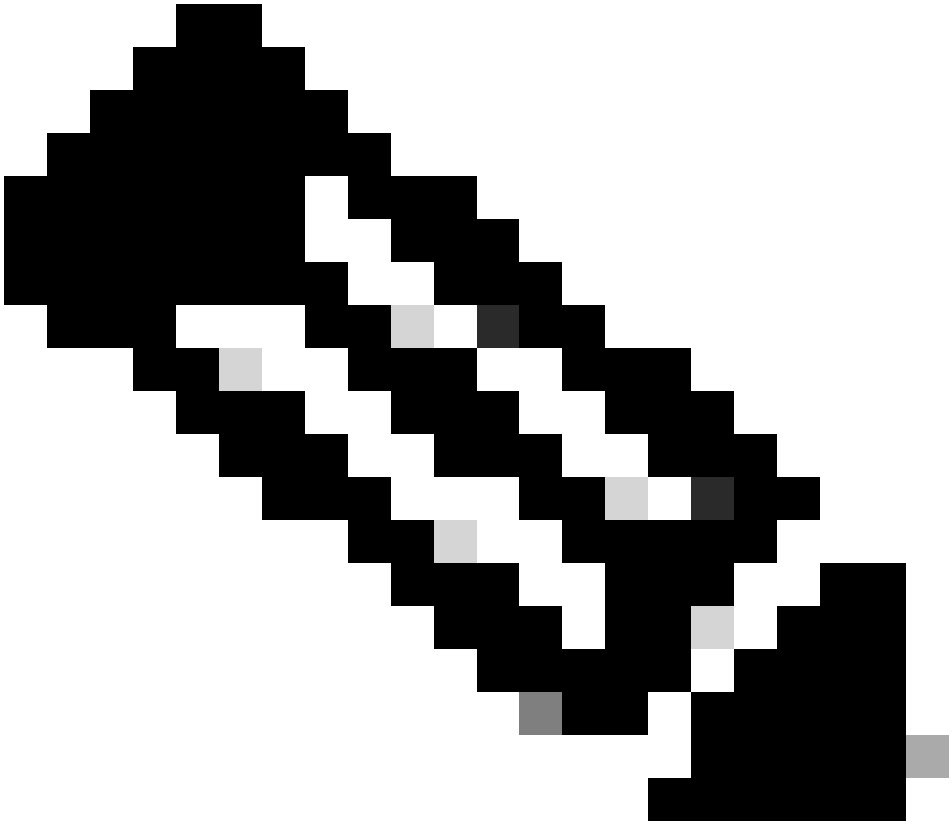
neighbor <ip-address> remote-as <number>

Le numéro dans la commande est le numéro d'AS du routeur auquel vous voulez vous connecter avec BGP. adresse-ip est l'adresse du prochain saut avec connexion directe pour eBGP. Pour iBGP, adresse-ip est n'importe quelle adresse IP sur l'autre routeur.

Les deux adresses IP que vous utilisez dans la neighbor commande des routeurs homologues *doivent* pouvoir se joindre. Une manière de vérifier l'accessibilité consiste à faire un test ping étendu entre les deux adresses IP. La commande ping étendue force le routeur à utiliser comme source l’adresse IP spécifiée par la commande ping neighbor . Le routeur doit utiliser cette adresse plutôt que l'adresse IP de l'interface à partir de laquelle le paquet est envoyé.

En cas de modifications de configuration de BGP, vous devez réinitialiser la connexion de voisinage pour permettre aux nouveaux paramètres d'entrer en vigueur. .

- clear ip bgp address



Remarque : Il s’agit de l’adresse du voisin

•

clear ip bgp *

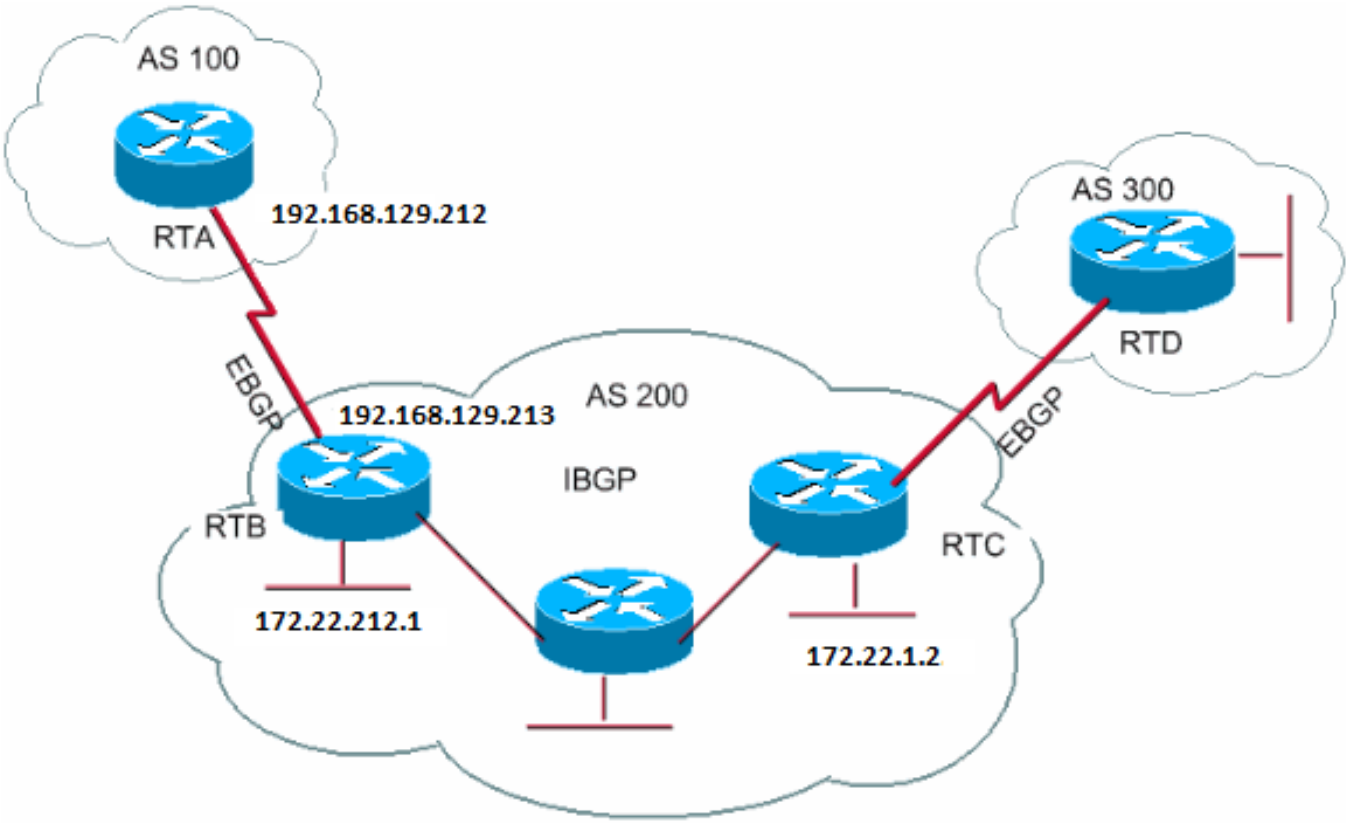
Cette commande efface toutes les connexions de voisinage.

Par défaut, les sessions BGP commencent par l'utilisation de BGP version 4 et négocient de manière descendante jusqu'aux versions antérieures, s'il y a lieu. Vous pouvez empêcher les négociations et forcer la version de BGP que les routeurs utilisent pour communiquer avec un voisin. Émettez la commande suivante en mode de configuration du routeur :

<#root>

neighbor {ip address | peer-group-name} version <value>

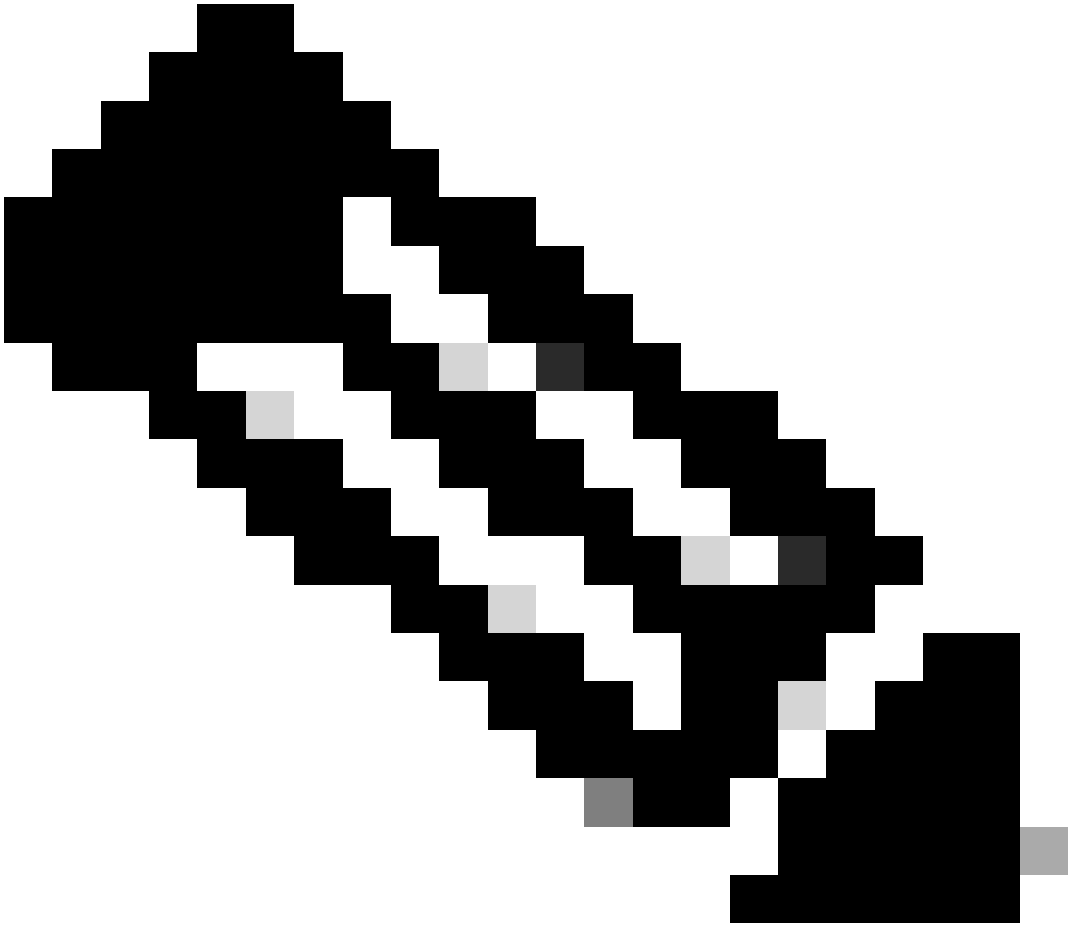
Voici un exemple de configuration de la neighbor commande :



```
RTA#
router bgp 100
 neighbor 192.168.129.213 remote-as 200

RTB#
router bgp 200
 neighbor 192.168.129.212 remote-as 100
 neighbor 172.22.1.2 remote-as 200

RTC#
router bgp 200
 neighbor 172.22.212.1 remote-as 200
```

Dans cet exemple, RTA et RTB exécutent eBGP. RTB et RTC exécutent iBGP. Le numéro de l'AS distant pointe vers un AS externe ou interne qui indique eBGP ou iBGP. De plus, les homologues eBGP ont une connexion directe, mais les homologues iBGP n’ont pas de connexion directe. Les routeurs iBGP n’ont pas besoin d’avoir une connexion directe. Cependant, un IGP doit s’exécuter pour permettre aux deux voisins de communiquer entre eux.

Cette section propose un exemple d'informations affichées par la commande show ip bgp neighbors.



Remarque : Portez une attention particulière à l’état du BGP. Hormis « Established » (établi), tout état indique que les homologues ne sont pas actifs. Notez également les éléments suivants :

- La version de BGP qui est la 4
- L'ID de routeur distant

Ce numéro est la plus haute adresse IP du routeur ou l'interface de bouclage la plus élevée, le cas échéant.

•

La version de la table

La version de la table indique l'état de la table. À chaque entrée de nouvelles informations, la table augmente la version. Une version qui continue d'être incrémentée indique une déflexion de route qui entraîne la mise à jour continue des routes.

<#root>

Router#

show ip bgp neighbors

```
  BGP neighbor is 192.168.129.213, remote AS 200, external link
  BGP version 4, remote router ID 172.22.12.1
```

BGP state = Established

```
, table version = 3, up for 0:10:59
  Last read 0:00:29, hold time is 180, keepalive interval is 60 seconds
  Minimum time between advertisement runs is 30 seconds
  Received 2828 messages, 0 notifications, 0 in queue
  Sent 2826 messages, 0 notifications, 0 in queue
  Connections established 11; dropped 10
```

BGP et interfaces de bouclage

L’utilisation d’une interface de boucle avec retour pour définir les voisins est courante avec iBGP, mais pas avec eBGP. Normalement, vous employez l'interface de bouclage pour vous assurer que l'adresse IP du voisin reste active et est indépendante du matériel qui fonctionne

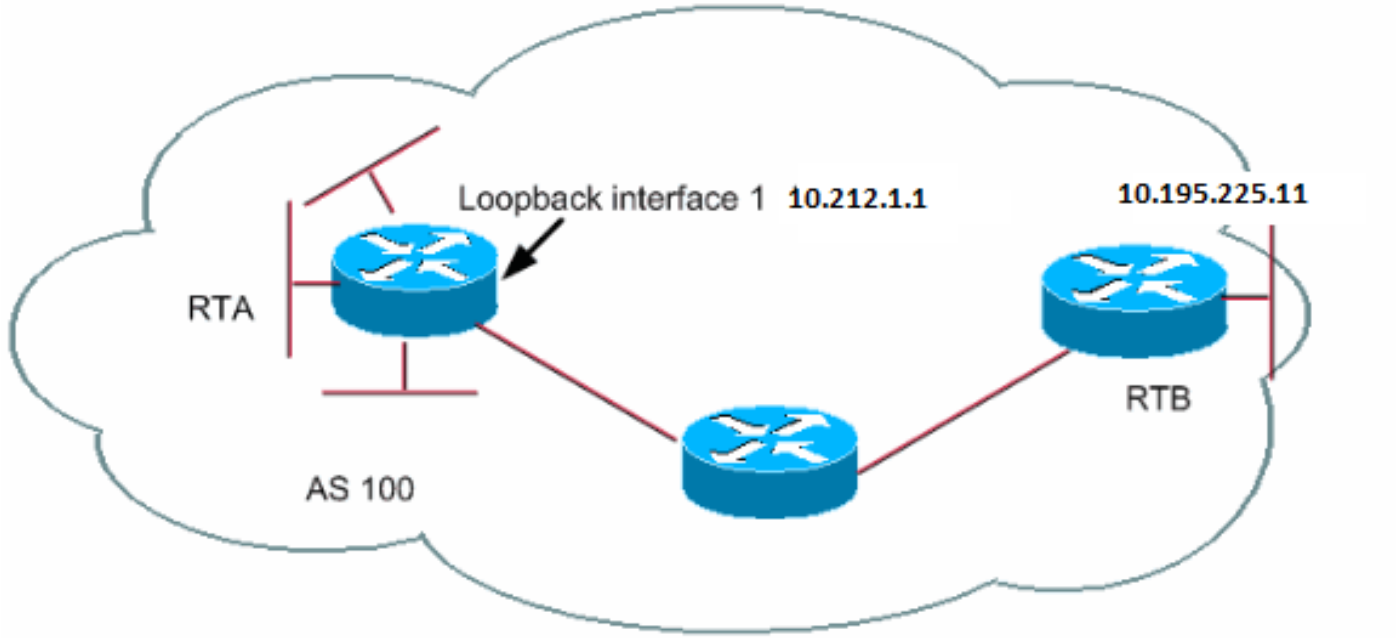
correctement. Dans le cas d'eBGP, les routeurs homologues utilisent fréquemment une connexion directe, et le bouclage ne s'applique pas.

Si vous utilisez l'adresse IP d'une interface de bouclage dans la neighbor commande, vous avez besoin d'une configuration supplémentaire sur le routeur voisin. Le routeur voisin doit informer BGP de l'utilisation d'une interface de bouclage plutôt qu'une interface physique pour initier la connexion TCP au voisin BGP. Pour indiquer une interface de bouclage, émettez la commande suivante :

<#root>

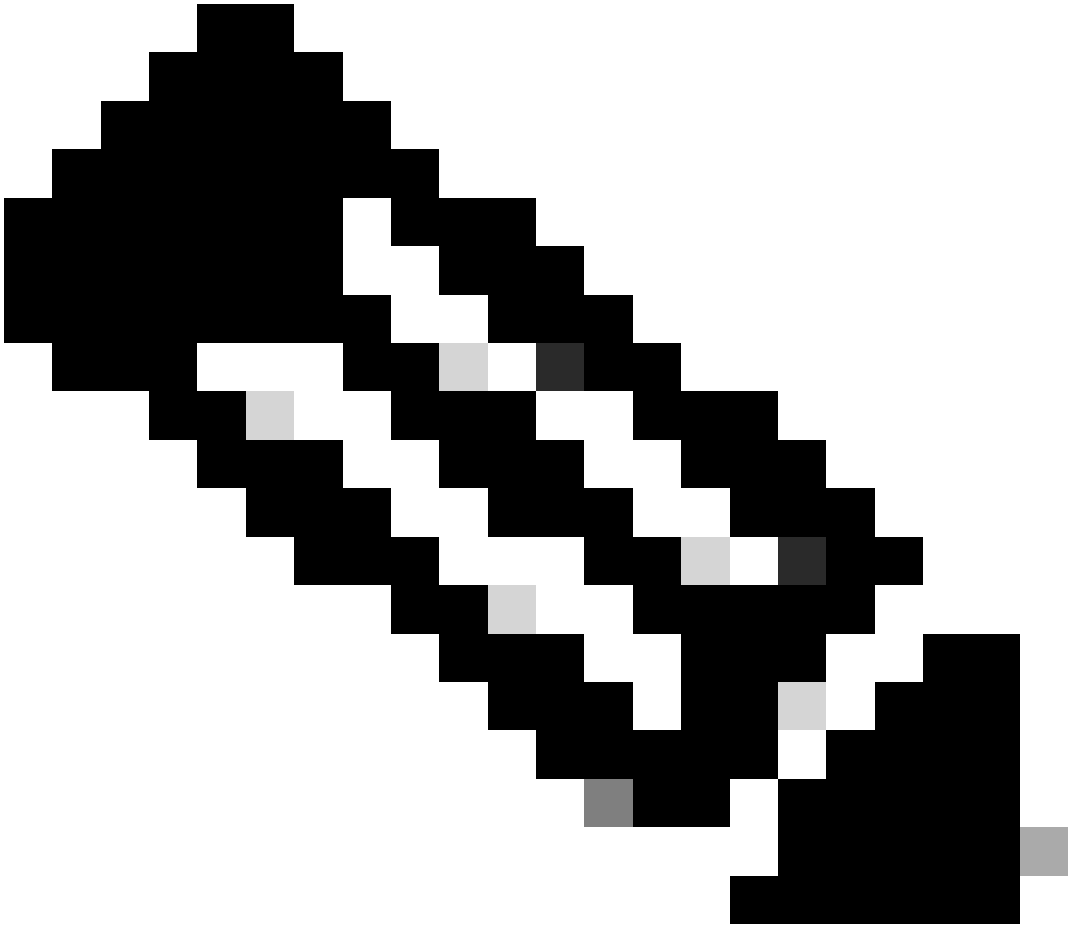
neighbor <ip-address> update-source <interface>

Cet exemple illustre l'utilisation de cette commande :




```
RTA#
router bgp 100
 neighbor 10.195.225.11 remote-as 100
 neighbor 10.195.225.11 update-source loopback 1

RTB#
router bgp 100
 neighbor 10.212.1.1 remote-as 100
```

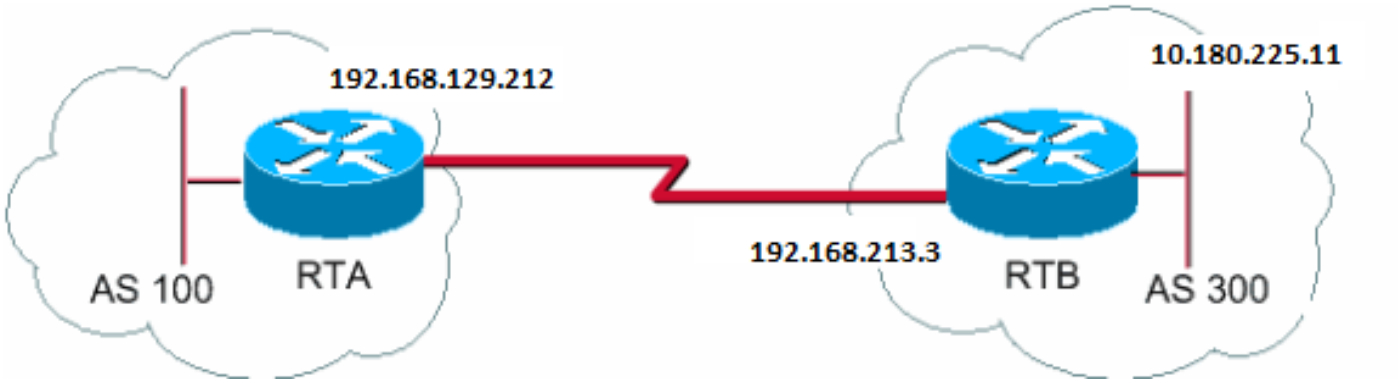
Dans cet exemple, RTA et RTB exécutent iBGP dans AS100. Dans la neighbor commande, RTB utilise l'interface de bouclage de RTA, 10.212.1.1. Dans ce cas, RTA doit forcer BGP à utiliser l'adresse IP de bouclage comme source dans la connexion de voisinage TCP. Afin de forcer cette action, RTA ajoute **update-source interface-type interface-number** afin que la commande soit neighbor 10.195.225.11 update-source loopback 1exécutée. Cette instruction force BGP à utiliser l'adresse IP de l'interface de bouclage quand BGP parle au voisin 10.195.225.11.



Remarque : Le RTA a utilisé l’adresse IP de l’interface physique de RTB, 10.195.225.11., en tant que voisin. L'utilisation de cette adresse IP explique pourquoi RTB n'a pas besoin de configuration spéciale. Référez-vous à l'exemple de configuration pour iBGP et eBGP avec ou sans adresse de bouclage pour obtenir un exemple de configuration de scénario réseau complet.

Saut multiple eBGP

Dans certains cas, un routeur Cisco peut exécuter eBGP avec un routeur tiers qui ne permet pas la connexion directe des deux homologues externes. Pour établir la connexion, vous pouvez utiliser le saut multiple eBGP. Le saut multiple eBGP permet d'établir une connexion de voisinage entre deux homologues externes qui n'ont pas de connexion directe. Le saut multiple eBGP s'applique seulement à eBGP et pas à iBGP. L'exemple suivant illustre le saut multiple eBGP :

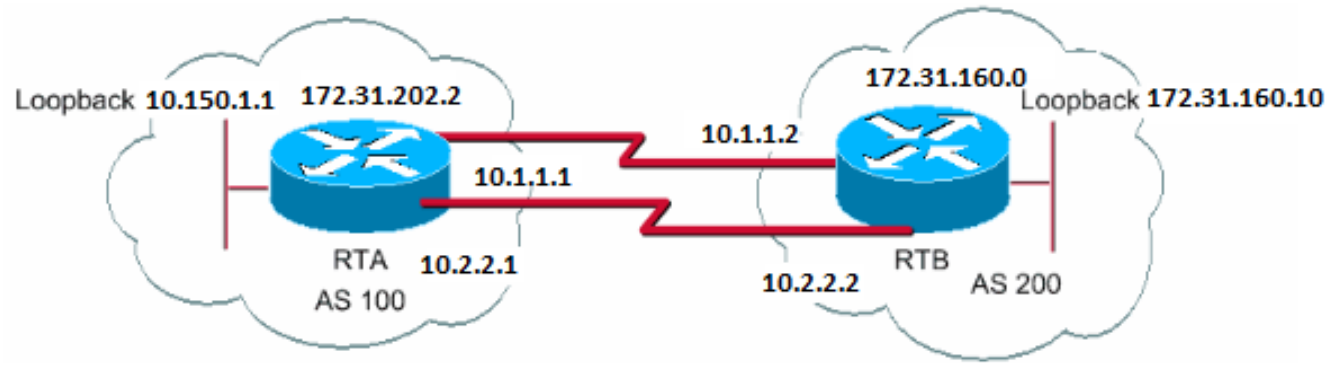



```
RTA#
router bgp 100
 neighbor 10.180.225.11 remote-as 300
 neighbor 10.180.225.11 ebgp-multihop

RTB#
router bgp 300
 neighbor 192.168.129.212 remote-as 100
```

RTA indique un voisin externe qui n'a pas de connexion directe. Le RTA doit indiquer s'il utilise la commande neighbor ebgp-multihop. D'un autre côté, le RTB indique un voisin qui a une connexion directe, soit 192.168.129.212. En raison de cette connexion directe, RTB n'a pas besoin de la neighbor ebgp-multihop commande. Vous devez également configurer un IGP ou un routage statique pour permettre aux voisins sans connexion de communiquer entre eux.

L'exemple de la section sur les sauts multiples eBGP (équilibrage de charge) montre comment réaliser l'équilibrage de charge avec BGP si vous avez un eBGP sur des lignes parallèles.

Saut multiple eBGP (équilibrage de charge)



```
RTA#
int loopback 0
 ip address 10.150.1.1 255.255.255.0

router bgp 100
 neighbor 172.31.160.10 remote-as 200
 neighbor 172.31.160.10 ebgp-multihop
 neighbor 172.31.160.10 update-source loopback 0
 network 172.31.202.2

ip route 172.31.160.0 255.255.0.0 10.1.1.2
ip route 172.31.160.0 255.255.0.0 10.2.2.2

RTB#
int loopback 0
 ip address 172.31.160.10 255.255.255.0

router bgp 200
 neighbor 10.150.1.1 remote-as 100
 neighbor 10.150.1.1 update-source loopback 0
 neighbor 10.150.1.1 ebgp-multihop
 network 172.31.160.0

ip route 172.31.202.2 255.255.0.0 10.1.1.1
ip route 172.31.202.2 255.255.0.0 10.2.2.1
```

Cet exemple illustre l'utilisation des interfaces de bouclage, update-source, et ebgp-multihop. L'exemple est une solution de contournement permettant d'équilibrer la charge entre deux speakers eBGP sur des lignes série parallèles. Dans des situations normales, BGP sélectionne une des lignes sur laquelle envoyer les paquets et l'équilibrage de charge ne se produit pas. Avec l'introduction des interfaces de bouclage, le prochain saut pour eBGP est l'interface de bouclage. Vous employez des routes statiques, ou un IGP, pour introduire deux chemins d'accès à coût égal pour atteindre la destination. Le RTA a deux choix pour atteindre le prochain saut 172.31.160.10 : un chemin qui passe par 10.1.1.2 et l’autre, par 10.2.2.2. RTB dispose des mêmes choix.

Mise en correspondance de route

BGP fait une utilisation intensive des mises en correspondance de route. Dans le contexte de BGP, la mise en correspondance de route est une méthode permettant de contrôler et de modifier les informations de routage. Le contrôle et la modification des informations de routage se produisent grâce à la définition des conditions de redistribution de routes d'un protocole de routage à l'autre. Le contrôle des informations de routage peut également s'effectuer à l'injection dans et hors de BGP. Voici le format de la carte de routage :

<#root>

route-map map-tag [[permit | deny] | [sequence-number]]

La balise map est simplement le nom que vous donnez à la mise en correspondance de route. Vous pouvez définir plusieurs instances d'une mise en correspondance de route, ou de la même balise de nom. Le numéro de séquence est simplement une indication de la position d'une nouvelle mise en correspondance de route dans la liste des mises en correspondance de route que vous avez déjà configurées avec le même nom.

Dans cet exemple, deux instances de mise en correspondance de route sont définies, avec le nom MYMAP. La première instance a le numéro de séquence 10, et la deuxième le numéro de séquence 20.

- **route-map MYMAP permit 10 (Le premier jeu de conditions s'affiche ici.)**

- **route-map MYMAP permit 20 (Le deuxième jeu de conditions s'affiche ici.)**

Quand vous appliquez la mise en correspondance de route MYMAP aux routes entrantes ou sortantes, le premier ensemble de conditions est appliqué par l'intermédiaire de l'instance 10. Si le premier jeu de conditions n'est pas respecté, vous passez à une instance plus élevée de la mise en correspondance de route.

Commandes de configuration match et set

Chaque route map est constituée d’une liste de commandes match et de set configuration. La correspondance spécifie un match critère et un jeu spécifie une set action si les critères appliqués par la match commande sont satisfaits.

Par exemple, vous pouvez définir une mise en correspondance de route qui vérifie les mises à jour sortantes. S'il existe une correspondance pour l'adresse IP 10.1.1.1, la métrique de cette mise à jour est définie sur 5. Les commandes suivantes illustrent l'exemple :

<#root>

match ip address 10.1.1.1

set metric 5

Maintenant, si les critères de correspondance sont satisfaits et que vous avez un permit, il y a une redistribution ou un contrôle des routes, comme l'action set le spécifie. Vous sortez de la liste.

Si les critères de correspondance sont satisfaits et que vous avez un deny, il n'y a aucune redistribution ou aucun contrôle de la route. Vous sortez de la liste.

Si les critères de correspondance ne sont pas remplis et que vous avez un permit ou deny , l'instance suivante de la carte de routage est vérifiée. Par exemple, l'instance 20 est vérifiée. Ce contrôle de l'instance suivante continue jusqu'à ce que vous sortiez de ou terminiez toutes les instances de la mise en correspondance de route. Si vous terminez la liste sans correspondance, la route est not accepted nor forwarded.

Dans les versions du logiciel Cisco IOS® antérieures à la version 11.2, lorsque vous utilisez des cartes de routage pour filtrer les mises à jour du BGP au lieu de les redistribuer entre les protocoles, vous *ne pouvez pas* appliquer de filtre entrant si vous utilisez une commande **match** sur l’adresse IP. Un filtre sur les données sortantes est acceptable. Le Logiciel Cisco IOS Version 11.2 et les versions postérieures n'ont pas cette restriction.

Les commandes associées à match sont :

- matchas-path
- match community
- matchclns
- match interface
- matchip address
- matchip nexthop

- matchip route-source
- matchmetric
- match route-type
- match tag

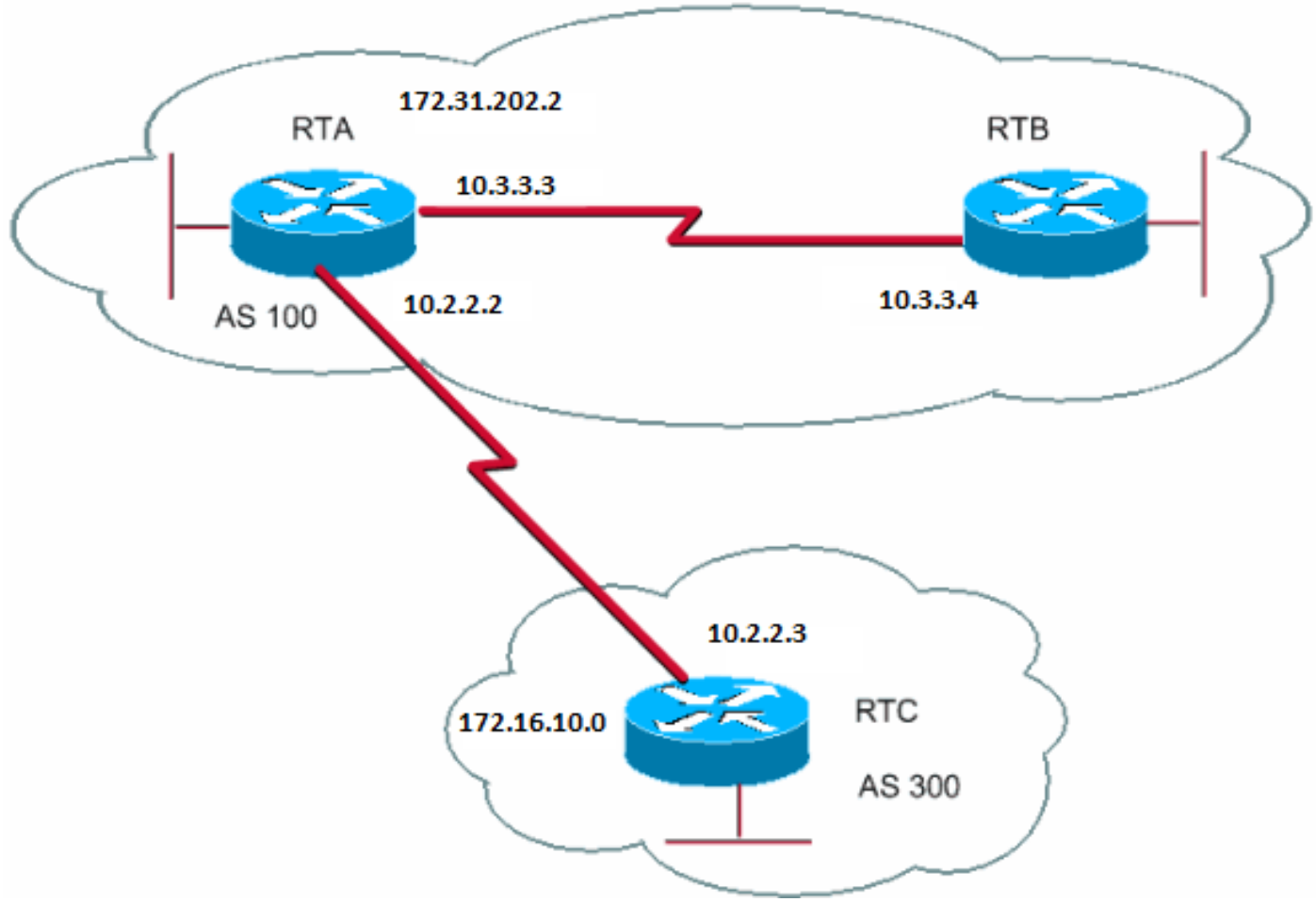
Les commandes associées à set sont :

- set as-path
- set clns
- set automatic-tag
- set community
- set interface
-

set default interface

- set ip default nexthop
- set level
- set local-preference
- set metric
- set metric-type
- set nexthop
- set origin
- set tag
- set weight

Voici quelques exemples de mises en correspondance de route :



Exemples de carte de routage

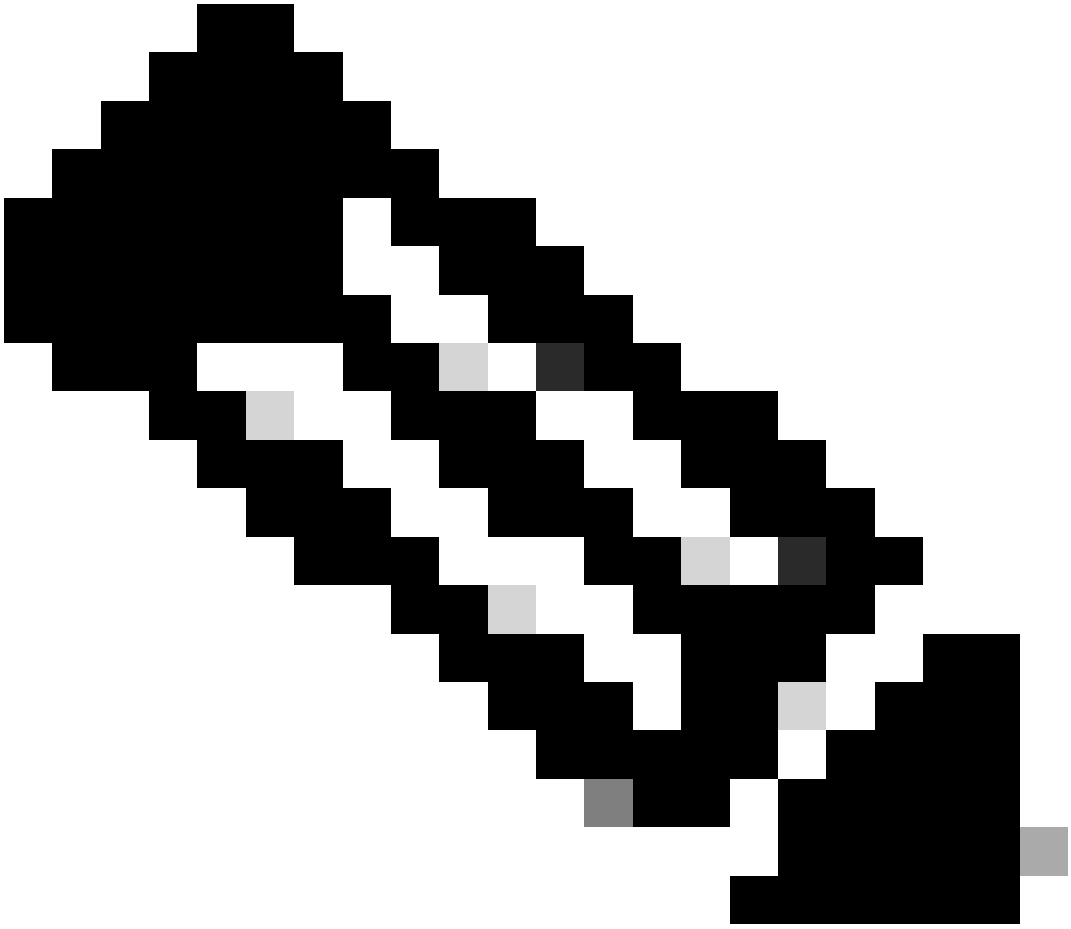
Exemple 1

Supposons que RTA et RTB exécutent le protocole d'informations de routage (RIP) et que RTA et RTC exécutent BGP. RTA obtient des mises à jour par l'intermédiaire de BGP et les redistribue à RIP. Supposons que le RTA veuille procéder à une redistribution vers les routes RTB relatives à 172.16.10.0 avec une mesure de 2 et toutes les autres routes avec une mesure de 5. Dans ce cas, vous pouvez utiliser cette configuration :

```
RTA#
router rip
 network 10.3.0.0
 network 10.2.0.0
 network 172.31.202.2
 passive-interface Serial0
 redistribute bgp 100 route-map SETMETRIC

router bgp 100
 neighbor 10.2.2.3 remote-as 300
 network 172.31.202.2

route-map SETMETRIC permit 10
 match ip-address 1
 set metric 2

route-map SETMETRIC permit 20
 set metric 5
```

```
access-list 1 permit 172.16.10.0 0.0.255.255
```

Dans cet exemple, si une route correspond à l'adresse IP 172.16.10.0, elle a une métrique de 2. Ensuite, vous sortez de la liste des mises en correspondance de route. S’il n’y a aucune correspondance, vous descendez dans la liste de cartes de routage, ce qui indique que tout le reste est défini sur la mesure 5.



Remarque : Posez toujours la question « Qu’advient-il des routes qui ne correspondent à aucune instruction de correspondance? ». Ces routes sont ignorées par défaut.

Exemple 2

Supposons que, dans l’exemple 1, vous ne souhaitez pas que l’AS100 accepte les mises à jour relatives à 172.16.10.0. Vous ne pouvez pas appliquer de mise en correspondance de route aux données entrantes lorsque vous établissez une correspondance avec une adresse IP en tant que base. Par conséquent, vous devez utiliser une mise en correspondance de route sortante sur RTC :

```
RTC#
router bgp 300
 network 172.16.10.0
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 route-map STOPUPDATES out

route-map STOPUPDATES permit 10
 match ip address 1

access-list 1 deny 172.16.10.0 0.0.255.255
access-list 1 permit 0.0.0.0 255.255.255.255
```

Maintenant que vous savez mieux comment démarrer BGP et définir un voisinage, découvrez comment démarrer l'échange des informations réseau.

Il existe plusieurs façons d'envoyer les informations réseau à l'aide de BGP. Les sections suivantes passent ces méthodes en revue une par une :

- Commande network
- Redistribution
- Routes statiques et redistribution

Commande network

Le format de la network commande est le suivant :

<#root>

network <network-number> mask <network-mask>

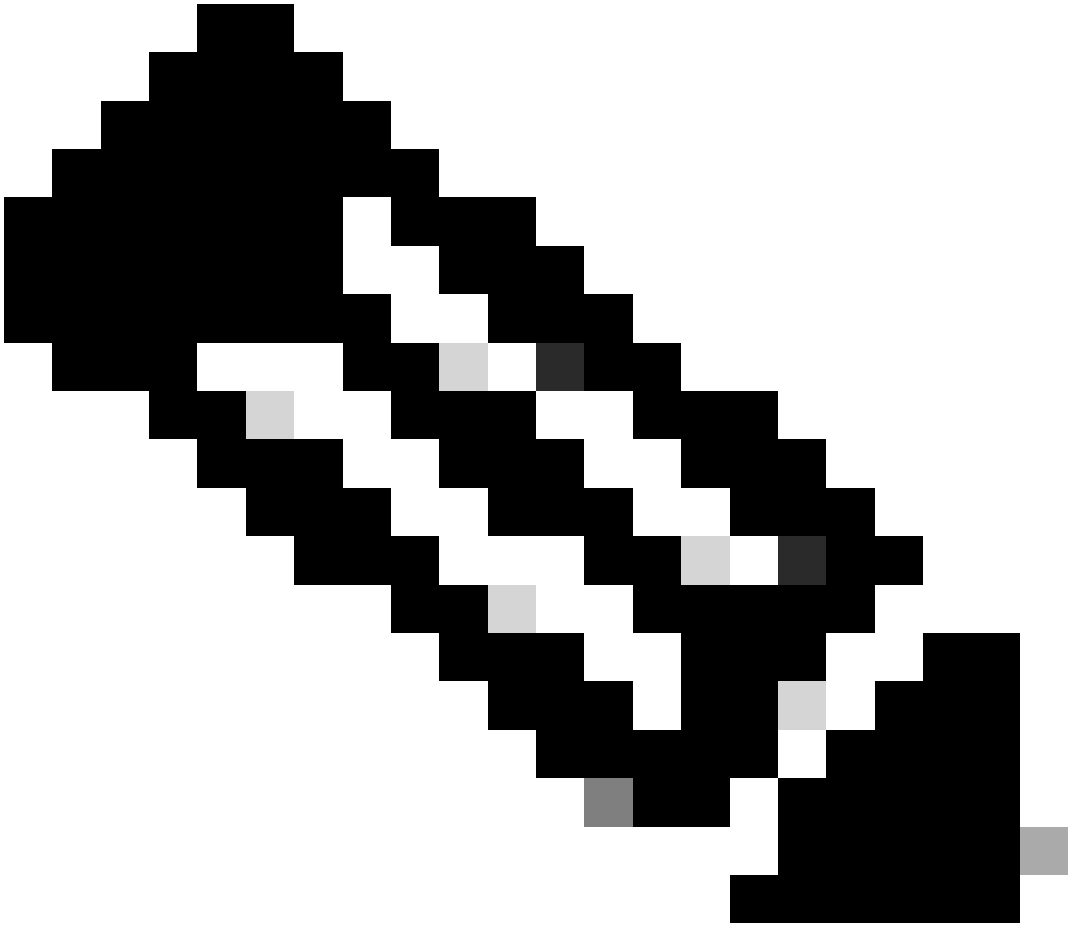
La network commande contrôle les réseaux qui proviennent de cette zone. Ce concept est différent de la configuration habituelle avec les protocoles Interior Gateway Routing Protocol (IGRP) et RIP. Avec cette commande, vous n'essayez pas d'exécuter BGP sur une interface donnée. Plutôt, vous essayez d'indiquer à BGP quels réseaux BGP doivent provenir de cette boîte. La commande utilise une partie de masque étant donné que BGP version 4 (BGP4) peut gérer les sous-réseaux et les super-réseaux. Un maximum de 200 entrées de la network commande est acceptable.

La network commande fonctionne si le routeur connaît le réseau que vous tentez d'annoncer, qu'il soit connecté, statique ou appris dynamiquement.

Voici un exemple de commande network :

```
RTA#
router bgp 1
 network 192.168.213.0 mask 255.255.0.0

ip route 192.168.213.0 255.255.0.0 null 0
```

Cet exemple montre que le routeur A génère une entrée réseau pour 192.168.213.0/16. /16 indique que vous utilisez un super-réseau d'adresse de classe C et que vous annoncez les deux premiers octets, ou les 16 premiers bits.

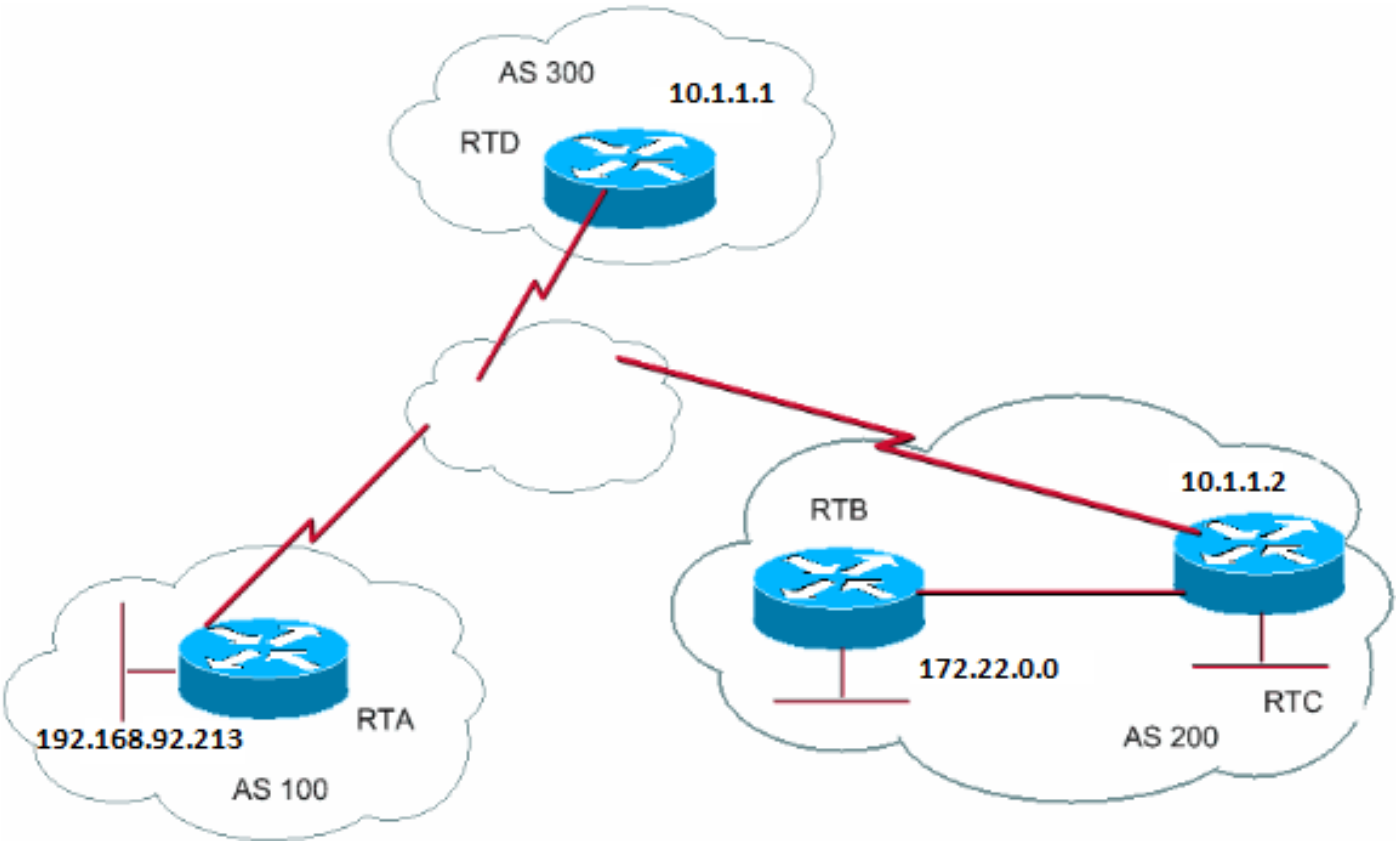


Remarque : Vous avez besoin de la route statique pour que le routeur génère 192.168.213.0, car la route statique insère une entrée correspondante dans la table de routage.

Redistribution

La network commande est une façon d'annoncer vos réseaux via BGP. Une autre méthode consiste à redistribuer votre IGP dans BGP. Votre IGP peut être le protocole IGRP, Open Shortest Path First (OSPF), RIP, Enhanced interior gateway routing protocol (EIGRP) ou un autre protocole. Cette redistribution peut sembler alarmante, car vous videz maintenant toutes vos routes internes dans BGP. Certaines d'entre elles peuvent avoir été obtenues par le BGP, et vous n'avez pas à les renvoyer. Lorsque vous filtrez, assurez-vous bien de faire l'envoi sur les routes Internet seulement que vous souhaitez annoncer, et non à toutes les routes dont vous disposez. Voici un exemple.

RTA annonce 192.168.92.213 et RTC annonce 172.22.0.0. Regardez la configuration RTC :



Si vous émettez la networkcommande, vous avez :

```
RTC#
router eigrp 10
 network 172.22.0.0
 redistribute bgp 200
 default-metric 1000 100 250 100 1500

router bgp 200
 neighbor 10.1.1.1 remote-as 300
 network 172.22.0.0 mask 255.255.0.0

!--- This limits the networks that your AS originates to 172.22.0.0.
```

Si vous utilisez la redistribution à la place, vous obtenez :

```
RTC#
router eigrp 10
network 172.22.0.0
redistribute bgp 200
default-metric 1000 100 250 100 1500

router bgp 200
neighbor 10.1.1.1 remote-as 300
redistribute eigrp 10
```

```
!--- EIGRP injects 192.168.92.213 again into BGP.
```

Cette redistribution entraîne la création de 192.168.92.213 par votre AS. Vous n’êtes pas la source de 192.168.92.213; AS100 est la source. Vous devez donc utiliser des filtres pour empêcher la source de sortir de ce réseau par votre AS. La configuration correcte est la suivante :

```
RTC#
router eigrp 10
 network 172.22.0.0
 redistribute bgp 200
 default-metric 1000 100 250 100 1500

router bgp 200
 neighbor 10.1.1.1 remote-as 300
 neighbor 10.1.1.1 distribute-list 1 out
 redistribute eigrp 10

access-list 1 permit 172.22.0.0 0.0.255.255
```

Vous utilisez la access-listcommande pour contrôler les réseaux qui proviennent d'AS200.

La redistribution d'OSPF vers BGP est légèrement différente de la redistribution pour d'autres IGP. La simple question du redistribute ospf 1sous-router bgpcontrôle ne fonctionne pas. Des mots-clés spécifiques tels que internal, external, et **nssa-external** sont nécessaires pour redistribuer les routes respectives. Consultez Understand the Redistribution of OSPF Routes into BGP (comprendre la redistribution des routes OSPF dans BGP) pour en savoir plus.

Routes statiques et redistribution

Vous pouvez toujours utiliser des routes statiques pour initier un réseau ou un sous-réseau. La seule différence est que BGP considère ces routes comme ayant une origine incomplète ou inconnue. Voici comment vous pouvez obtenir le même résultat que celui de l’exemple de la section sur la redistribution :

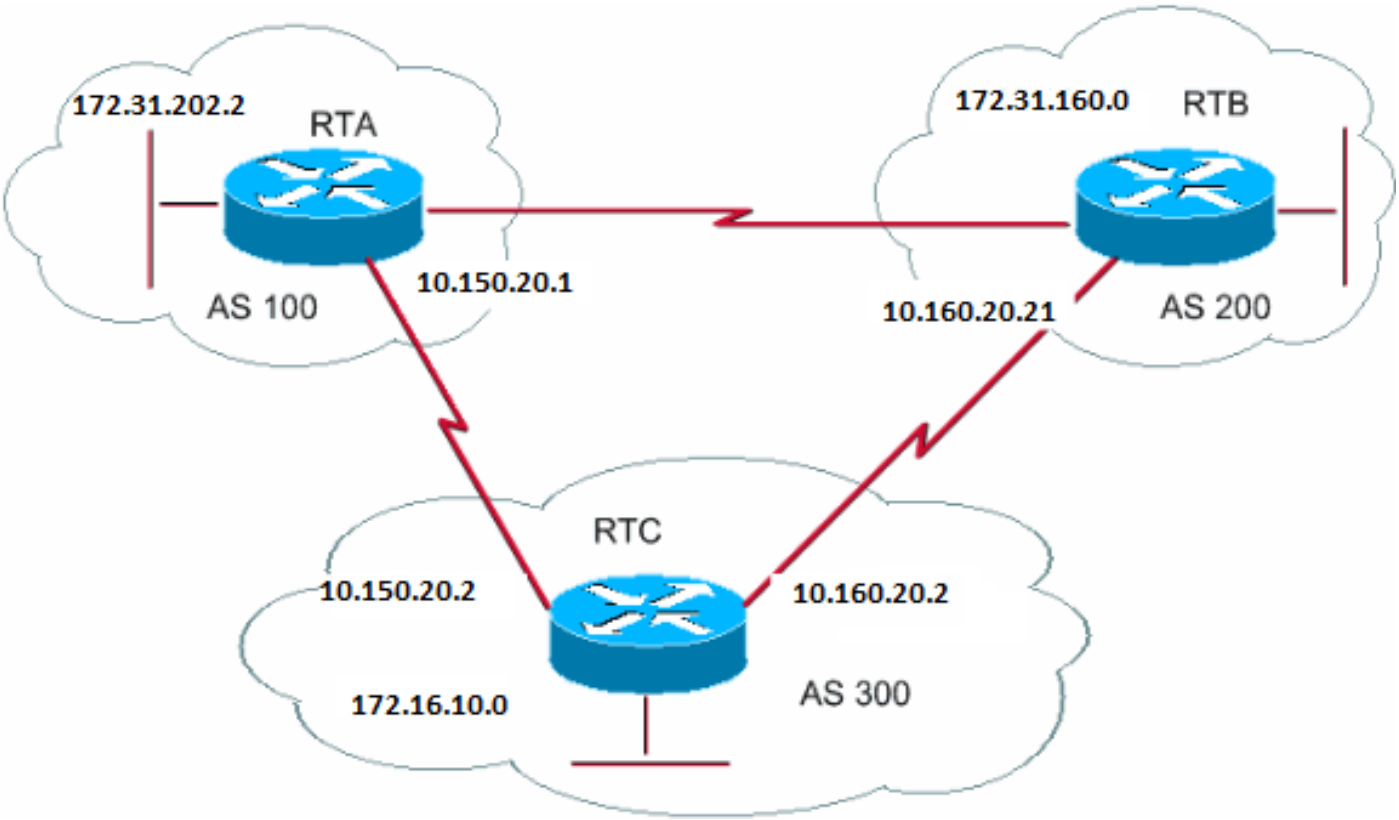
```
RTC#
router eigrp 10
 network 172.22.0.0
 redistribute bgp 200
 default-metric 1000 100 250 100 1500

router bgp 200
 neighbor 10.1.1.1 remote-as 300
 redistribute static

ip route 172.22.0.0 255.255.255.0 null0
```

L'null0 interface signifie ignorer le paquet. Alors, si vous obtenez le paquet et qu'il existe une correspondance plus précise que 172.22.0.0, le routeur envoie le paquet à cette correspondance. Autrement, le routeur ignore le paquet. Cette méthode permet d'annoncer facilement un super-réseau.

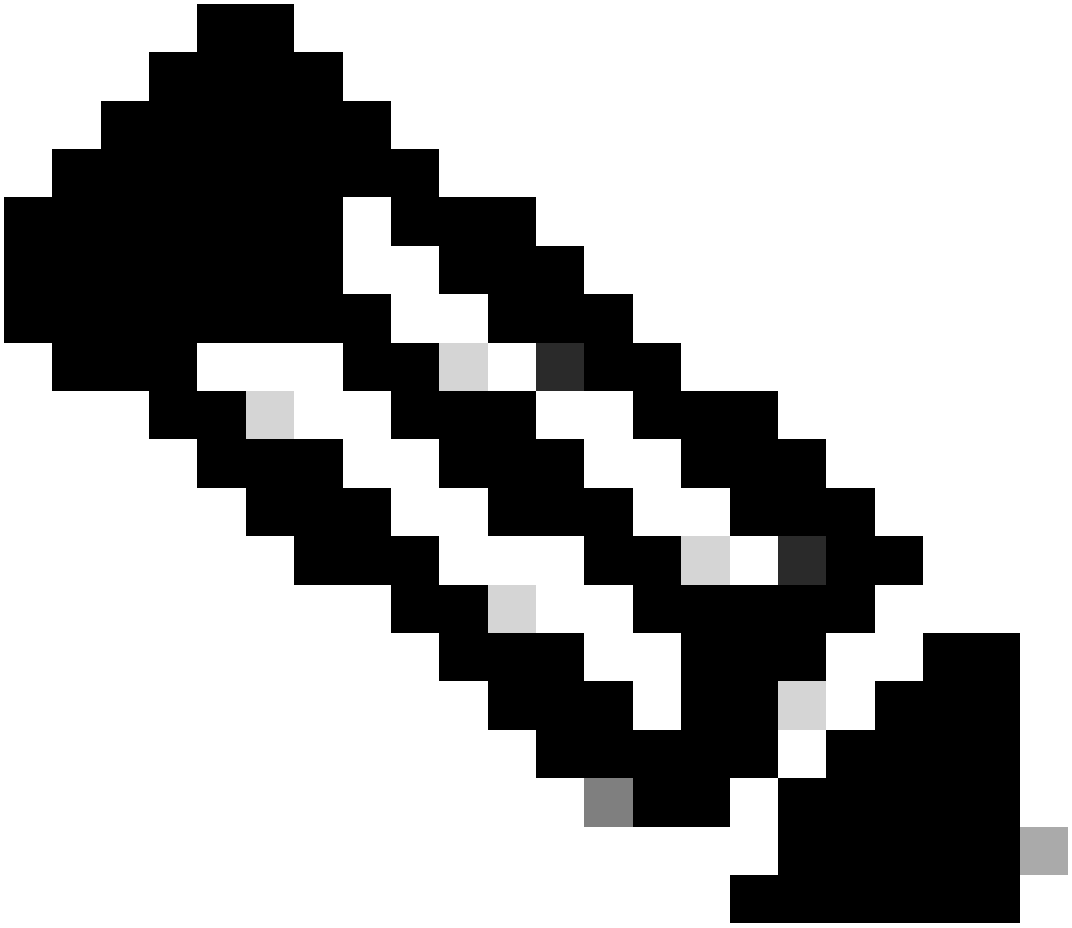
Ce document a présenté les différentes méthodes utilisées pour initier des routes à partir de votre AS. Rappelez-vous que ces routes sont générées en plus des autres routes BGP que BGP a apprises par l'intermédiaire des voisins, qu'elles soient internes ou externes. BGP transmet les informations recueillies par BGP auprès d'un homologue aux autres homologues. La différence est que les routes qui génèrent à partir de la network commande, redistribution ou statique indiquent que votre AS est l'origine de ces réseaux.

La redistribution est toujours la méthode utilisée pour l'injection de BGP dans IGP.

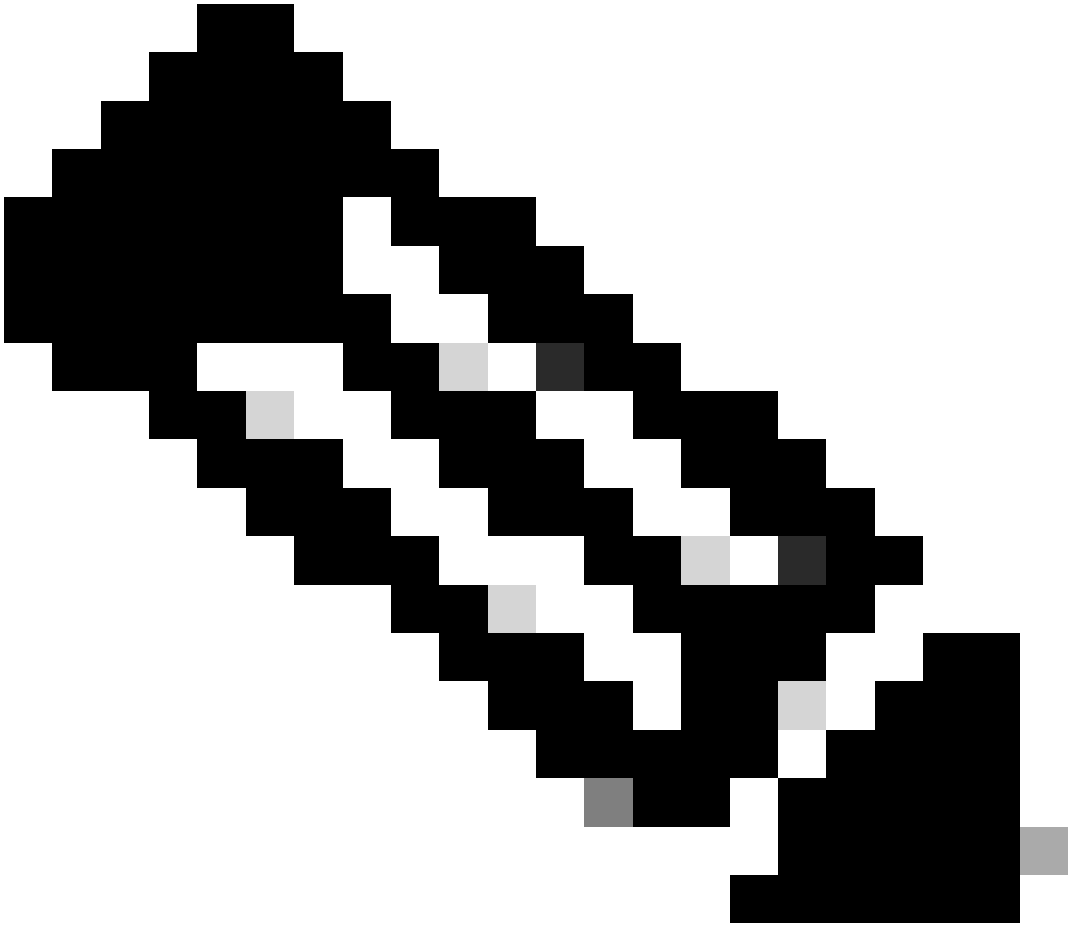
Voici un exemple :




```
RTA#
router bgp 100
 neighbor 10.150.20.2 remote-as 300
 network 172.31.202.2

RTB#
router bgp 200
 neighbor 10.160.20.2 remote-as 300
 network 172.31.160.0

RTC#
router bgp 300
 neighbor 10.150.20.1 remote-as 100
 neighbor 10.160.20.21 remote-as 200
 network 170.10.00
```



Remarque : Vous n’avez pas besoin du réseau 172.31.202.2 ou du réseau 172.31.160.0 dans le RTC, sauf si vous souhaitez que ce dernier génère ces réseaux et les transmette au fur et à mesure qu’ils arrivent d’AS100 et d’AS200. De nouveau, la différence est que la commande network ajoute une annonce supplémentaire pour ces réseaux qui indique qu'AS300 est également une origine pour ces routes.



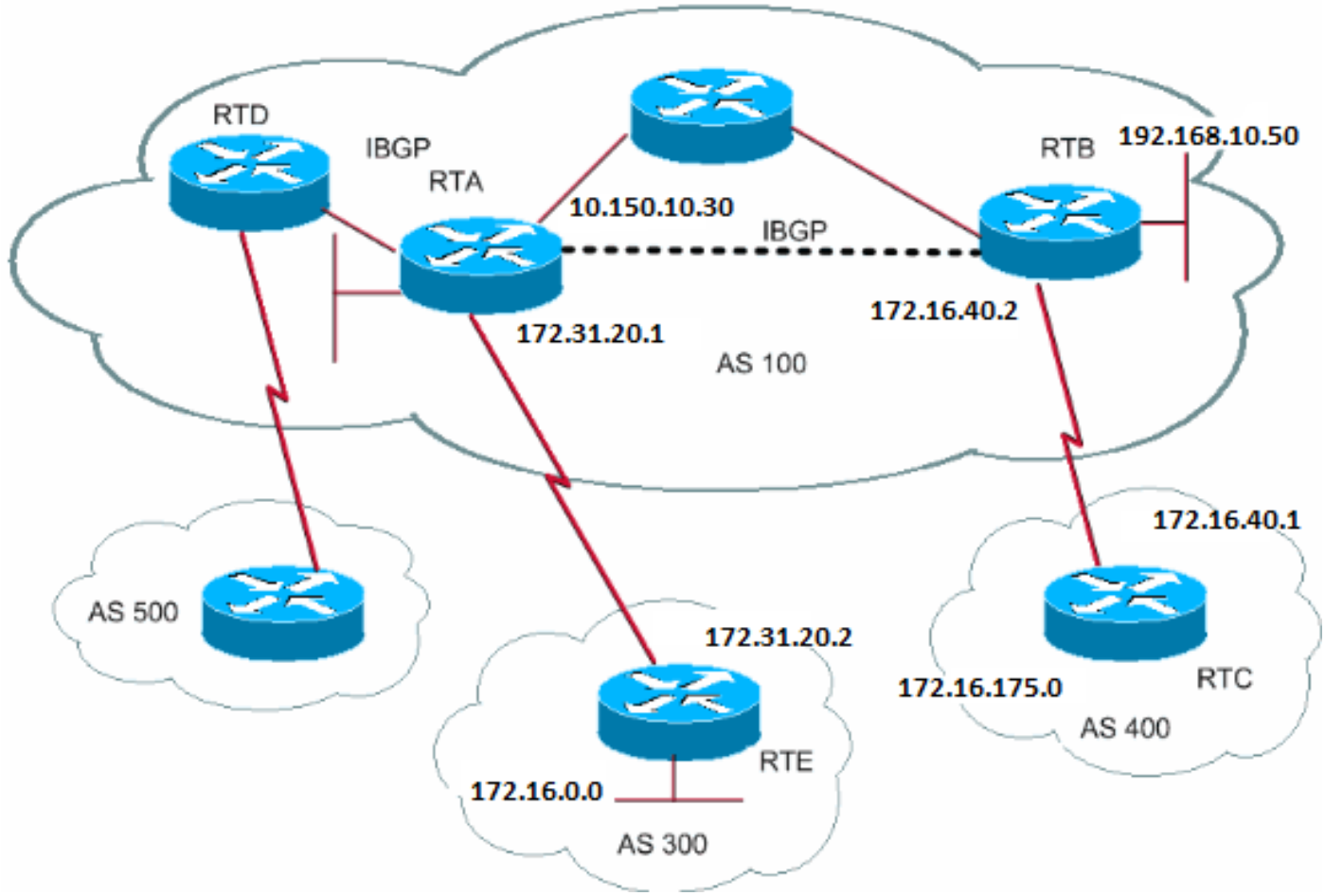
Remarque : Rappelez-vous que BGP n’accepte pas les mises à jour qui proviennent de son AS. Ce refus assure une topologie interdomaine sans boucle.

Par exemple, supposons qu'AS200 (dans l'exemple de cette section) dispose d'une connexion BGP directe à AS100. RTA génère une route 172.31.202.2 et l'envoie à AS300. Ensuite, RTC passe cette route à AS200 et conserve son origine comme étant AS100. RTB passe 172.31.202.2 à AS100 toujours avec l'origine AS100. RTA remarque que la mise à jour provient de son propre AS et l'ignore.

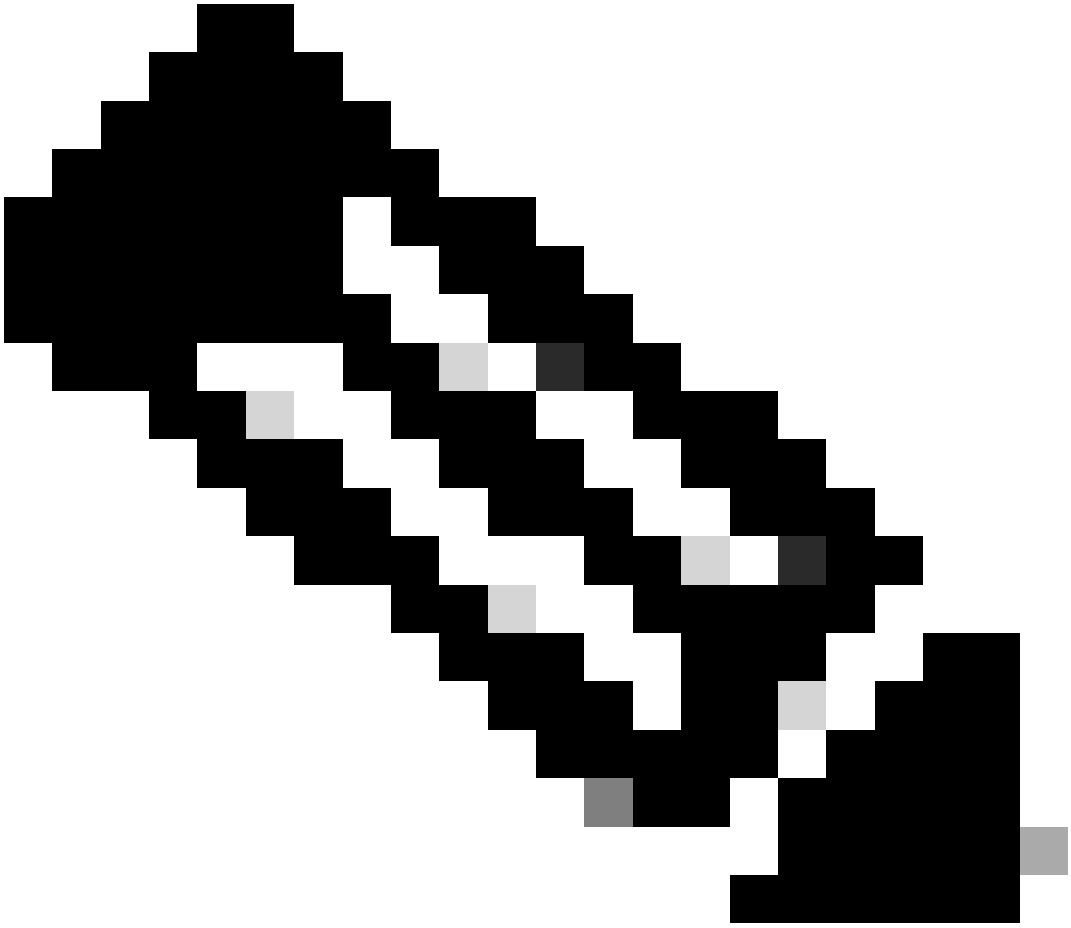
iBGP

Vous utilisez iBGP si un AS souhaite agir en tant que système de transit pour un autre AS. Vous pouvez faire la même chose si la détection se fait par eBGP et si la redistribution a lieu dans IGP, puis dans un autre AS. Toutefois, iBGP offre une plus grande flexibilité et des moyens plus efficaces d’échanger des renseignements au sein d’un AS. Par exemple, iBGP permet de contrôler de différentes manières le meilleur point de

sortie de l'AS à l'aide de la préférence locale. La section « Local Preference Attribute » (attribut des préférences locales) donne davantage d’information sur les préférences locales.




```
RTA#
router bgp 100
 neighbor 192.168.10.50 remote-as 100
 neighbor 172.31.20.2 remote-as 300
 network 172.31.202.2

RTB#
router bgp 100
 neighbor 10.150.10.30 remote-as 100
 neighbor 172.16.40.1 remote-as 400
 network 192.168.10.150

RTC#
router bgp 400
 neighbor 172.16.40.2 remote-as 100
 network 172.16.0.0
```



Remarque : N’oubliez pas que lorsqu’un haut-parleur BGP reçoit une mise à jour d’autres haut-parleurs BGP dans son propre AS (iBGP), le haut-parleur BGP qui reçoit la mise à jour ne redistribue pas l’information aux autres haut-parleurs BGP de son AS. Le speaker BGP qui reçoit la mise à jour redistribue l'information aux autres speakers BGP situés en dehors de son AS. Par conséquent, maintenez un maillage global entre les speakers iBGP au sein d'un AS.

Le RTA et le RTB exécutent iBGP. RTA et RTD exécutent également iBGP. Les mises à jour BGP entre RTB et RTA sont transmises à RTE qui se situe hors de l'AS. Les mises à jour ne sont pas transmises à RTD, qui se trouve dans l'AS. Par conséquent, effectuez une interconnexion iBGP entre RTB et RTD pour ne pas interrompre le flux des mises à jour.

Algorithme de décision BGP

Une fois que BGP a reçu des mises à jour au sujet de différentes destinations issues de différents systèmes autonomes, le protocole doit choisir

des chemins pour atteindre une destination spécifique. BGP choisit seulement un chemin unique pour atteindre une destination spécifique.

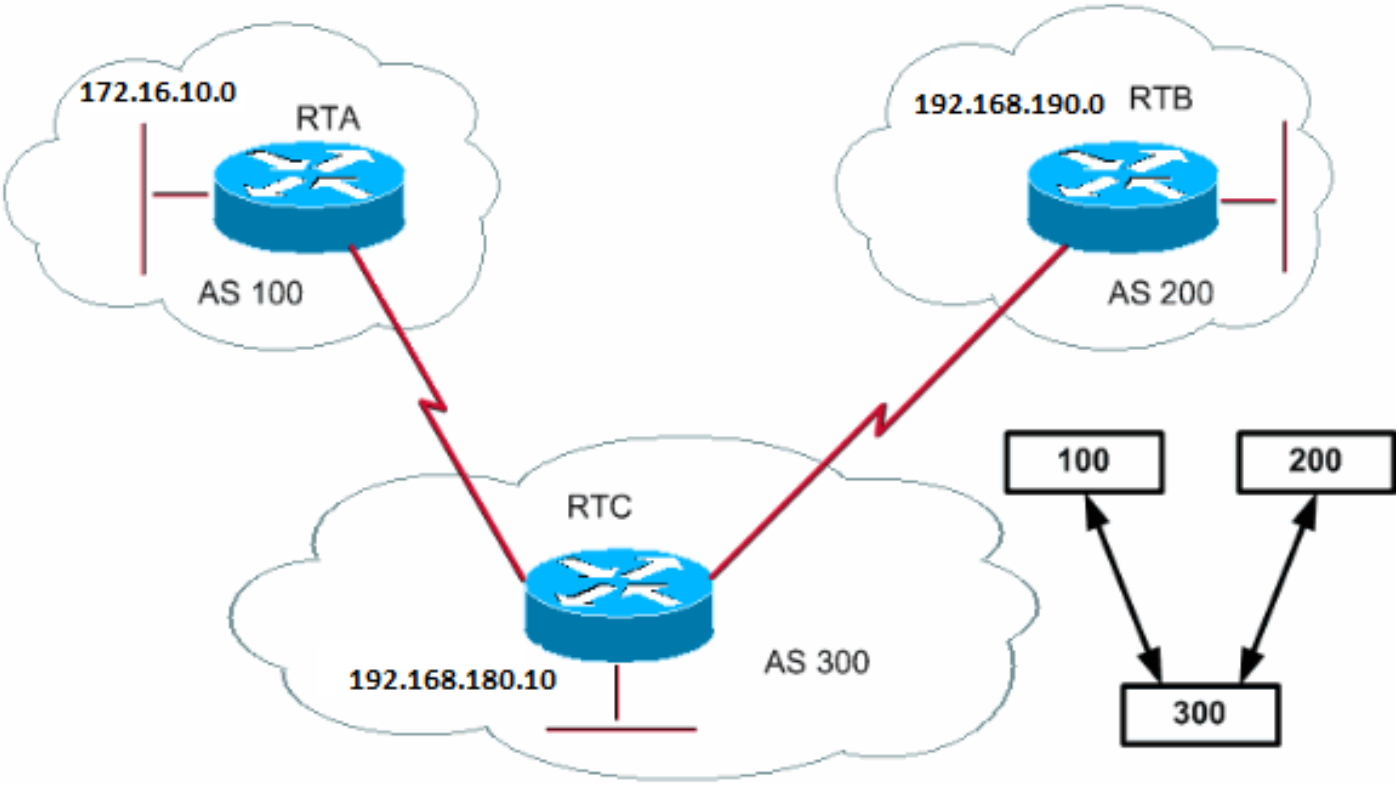
BGP base la décision sur différents attributesattributs, tels que le saut suivant, les pondérations administratives, la préférence locale, l'origine de la route, la longueur du chemin, le code d'origine, la métrique et d'autres attributs.

BGP propage toujours le meilleur chemin aux voisins. Consultez BGP Best Path Selection Algorithm (algorithme de sélection du meilleur chemin BGP) pour en savoir plus.

La section suivante explique ces attributs et leur utilisation.

Études de cas BGP 2

Attribut AS_PATH



À chaque fois qu'une mise à jour de route transite par un AS, le numéro de l'AS est préfixé à cette mise à jour. L'attribut AS_PATH est en fait la liste des numéros des AS qu'une route a traversés pour atteindre une destination. Un AS_SET est un ensemble mathématique ordonné {} de tous les AS qui ont été traversés. La section « CIDR Example 2 (as-set) » (exemple 2 du CIDR [as-set]) du présent document donne un exemple de la commande AS_SET.

Dans l'exemple de cette section, RTB annonce le réseau 192.168.190.0 dans AS200. Quand cette route traverse AS300, RTC ajoute son propre numéro d'AS au réseau. Lorsque 192.168.190.0 atteint RTA, deux numéros d'AS sont associés au réseau : d'abord le 200, puis le 300. Pour RTA, le chemin pour atteindre 192.168.190.0 est (300, 200).

Le même processus s'applique à 172.16.10.0 et à 192.168.180.10. Le RTB doit prendre le chemin (300, 100). Ensuite, RTB traverse AS300, puis AS100 pour atteindre 172.16.10.0. RTC doit traverser le chemin (200) afin d'atteindre 192.168.190.0 et le chemin (100) afin d'atteindre 172.16.10.0.

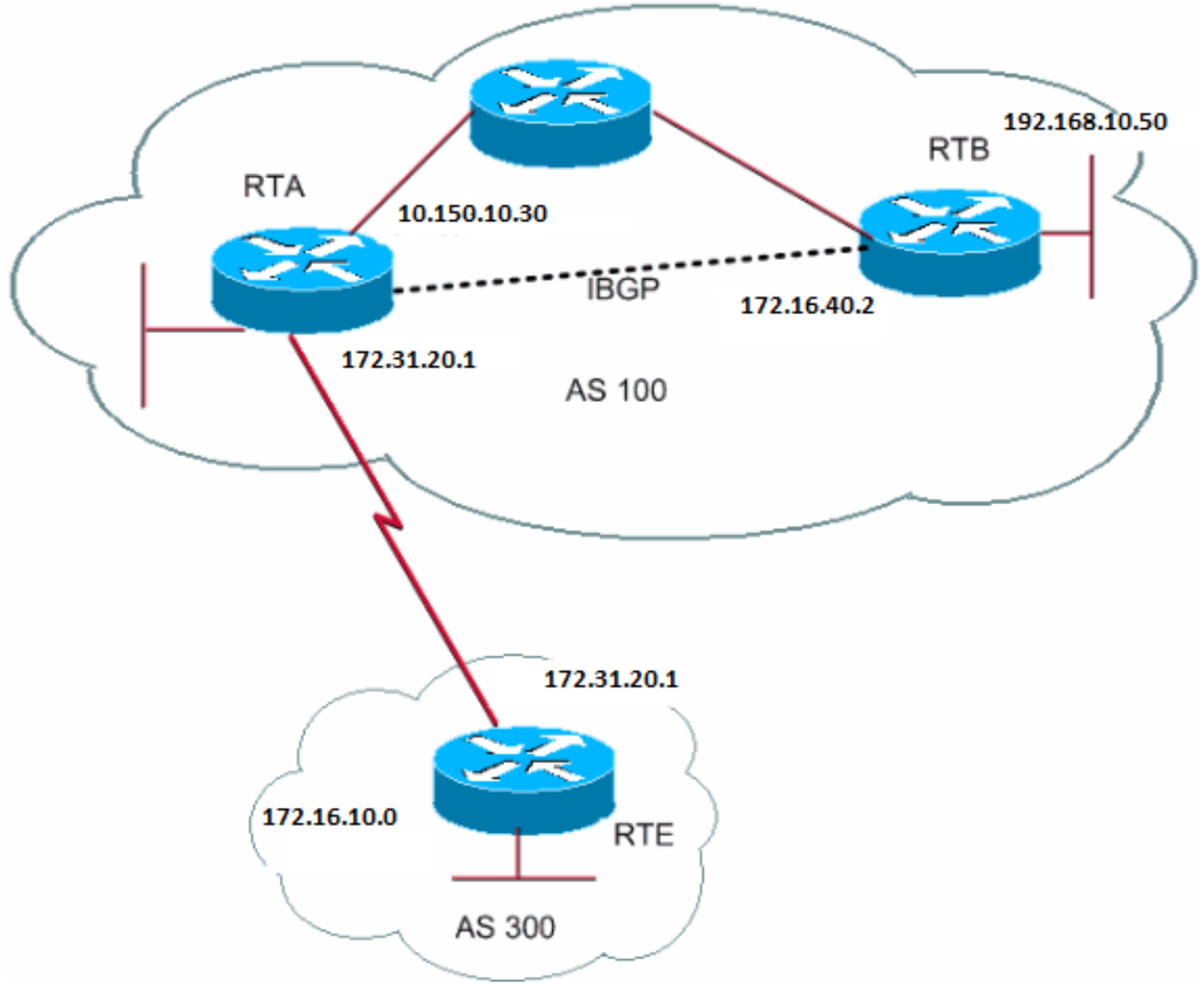
Attribut origin

L'origine est un attribut obligatoire qui définit l'origine des informations de chemin. L'attribut origin peut avoir trois valeurs :

- IGP : les informations d'accessibilité des couches réseau (NLRI) sont internes à l'AS d'origine. Cela se produit normalement lorsque vous émettez la **bgp network**commande . Un `i` dans le tableau BGP indique IGP.

- EGP : les informations NLRI sont apprises par l'intermédiaire de l'Exterior Gateway Protocol (EGP). Un `e` dans le tableau BGP indique EGP.

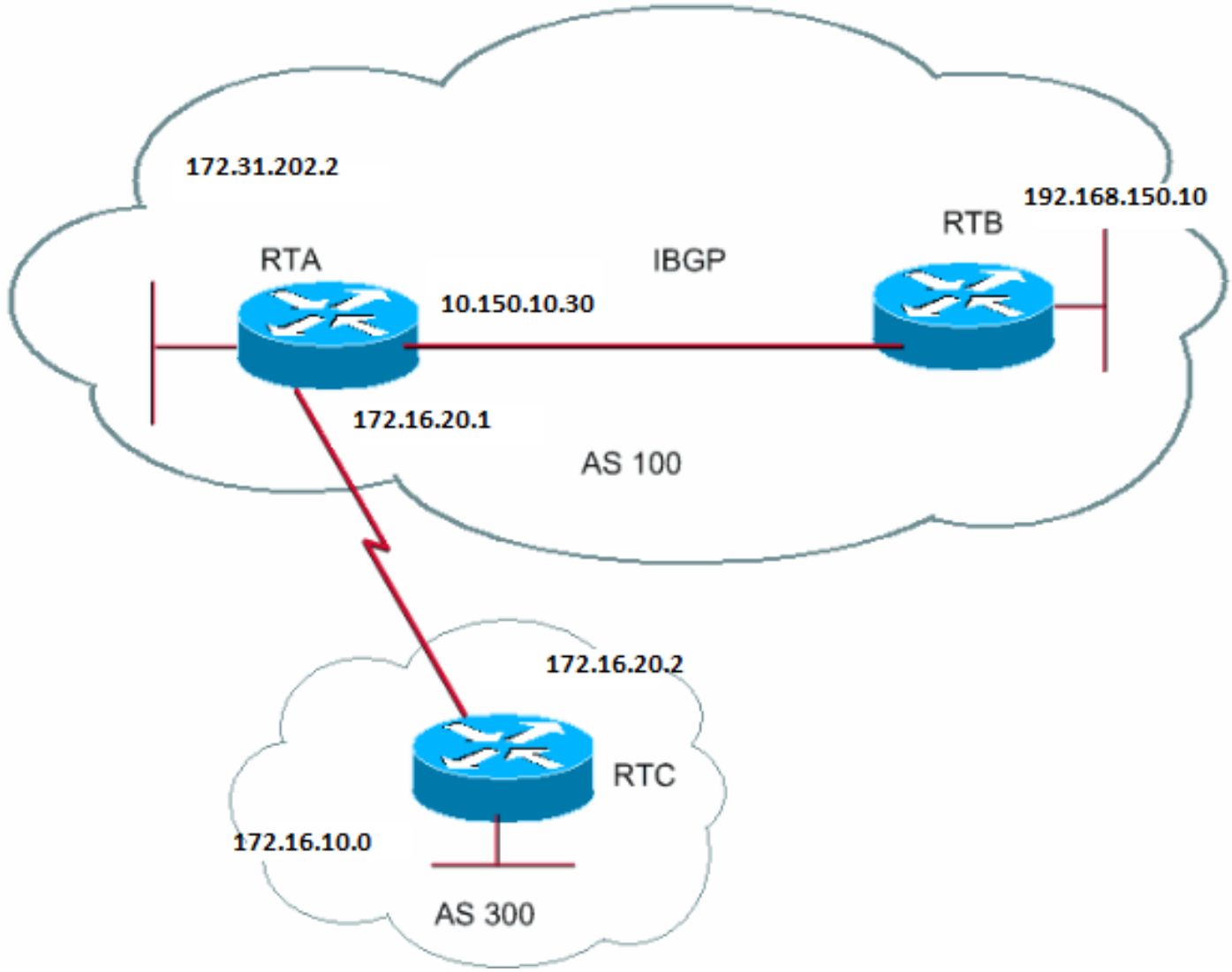
- INCOMPLETE : les informations NLRI sont inconnues ou apprises par un autre moyen. INCOMPLETE se produit habituellement quand vous redistribuez les routes d'autres protocoles de routage dans BGP et que l'origine de la route est incomplète. Un `?` dans le tableau BGP indique INCOMPLETE (incomplet).




```
RTA#
router bgp 100
 neighbor 192.168.10.50 remote-as 100
 neighbor 172.31.20.2 remote-as 300
 network 172.31.202.2
 redistribute static

ip route 192.168.190.0 255.255.0.0 null0

RTB#
router bgp 100
 neighbor 10.150.10.30 remote-as 100
 network 192.168.10.150

RTE#
router bgp 300
 neighbor 172.31.20.1 remote-as 100
 network 172.16.10.0
```

RTA atteint 172.16.10.0 par l'intermédiaire de 300 i. « 300 i » signifie que le chemin d'AS suivant est 300 et que l'origine de la route est IGP. RTA atteint également 192.168.10.150 par l'intermédiaire d'i. Ce « i » signifie que l'entrée se situe dans le même AS et que l'origine est IGP. RTE atteint 172.31.202.2 par l'intermédiaire de 100 i. « 100 i » signifie que l'AS suivant est 100 et que l'origine est IGP. RTE atteint également 192.168.190.0 par l'intermédiaire de 100 ?. La requête « 100 ? » signifie que le prochain AS est 100 et que l'origine est incomplète et provient d'une route statique.

Attribut BGP next hop



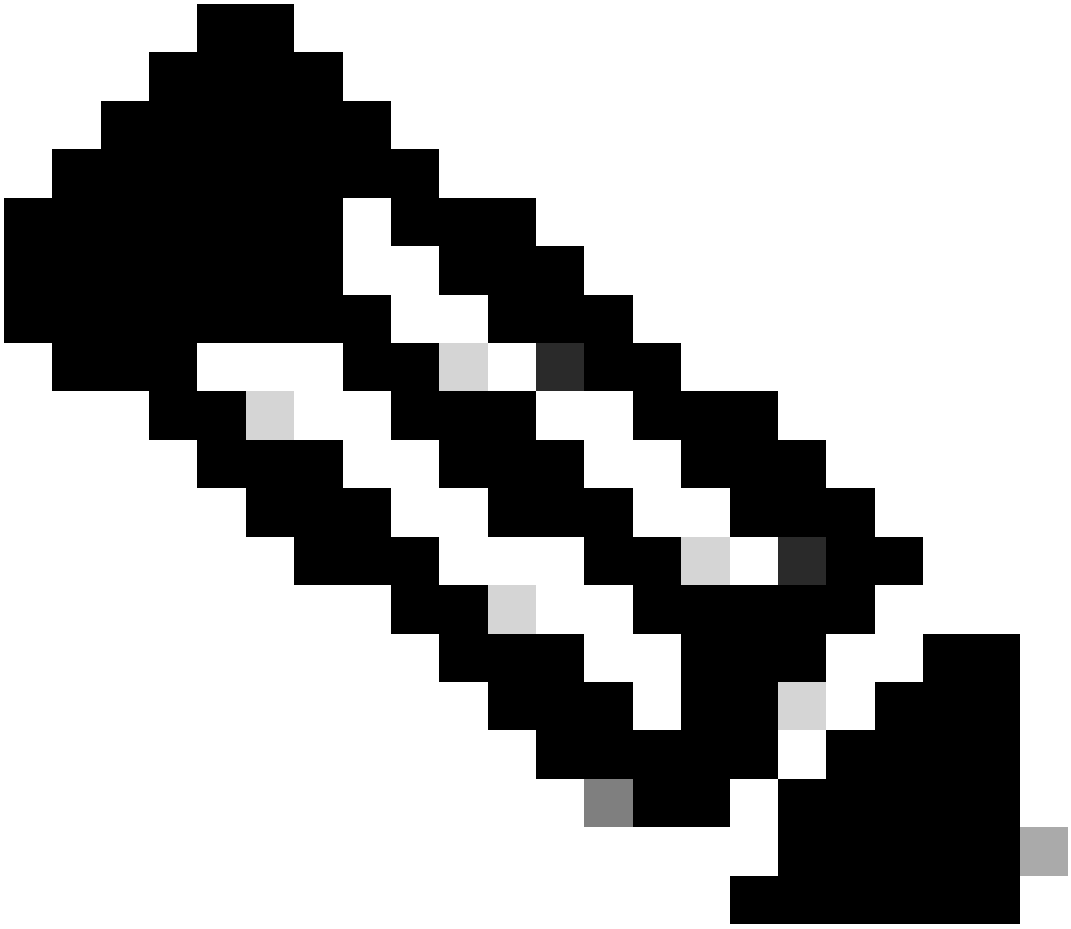
Attribut BGP next hop

L'attribut BGP next hop correspond à l'adresse IP du prochain saut à utiliser pour atteindre une destination donnée.

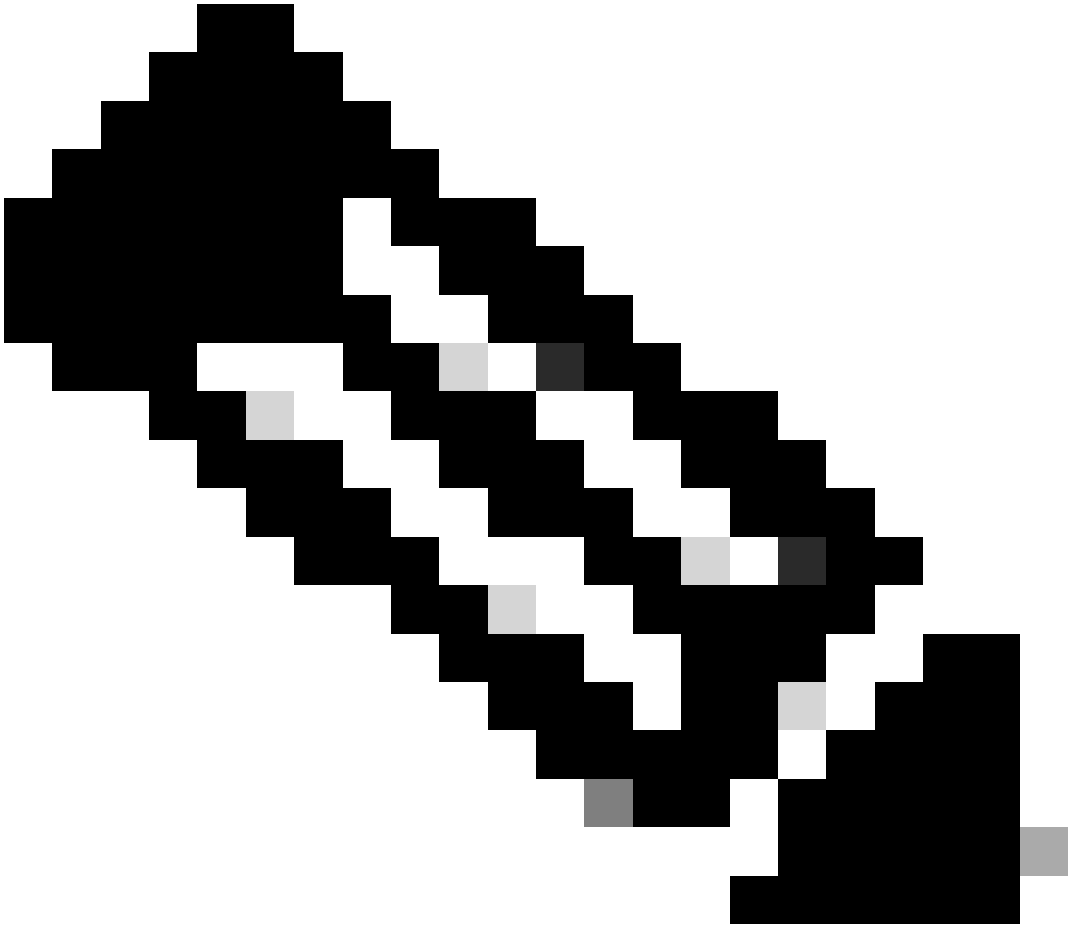
Pour eBGP, le saut suivant est toujours l'adresse IP du voisin spécifiée par la neighbor commande. Dans l'exemple de cette section, RTC annonce 172.16.10.0 à RTA avec un prochain saut de 172.31.20.2. RTA annonce 172.31.202.2 à RTC avec un prochain saut de 172.31.20.1. Pour iBGP, le protocole indique que le prochain saut annoncé par eBGP doit être acheminé dans iBGP. En raison de cette règle, RTA annonce 172.16.10.0 à son homologue iBGP RTB avec un prochain saut de 172.31.20.2. Selon le RTB, le prochain saut à atteindre 172.16.10.0 est 172.31.20.2 et *non* 10.150.10.30.

Assurez-vous que RTB peut atteindre 172.31.20.2 par l'intermédiaire d'IGP. Sinon, RTB ignore les paquets avec la destination 172.16.10.0 parce que l'adresse du prochain saut est inaccessible. Par exemple, si RTB exécute iGRP, vous pouvez également exécuter iGRP sur le réseau RTA 172.16.10.0. Vous voulez rendre iGRP passif sur le lien à RTC de sorte que BGP soit seulement échangé.

```
RTA#
router bgp 100
 neighbor 172.31.20.2 remote-as 300
 neighbor 192.168.150.10 remote-as 100
 network 172.31.202.2

RTB#
router bgp 100
 neighbor 10.150.10.30 remote-as 100

RTC#
router bgp 300
 neighbor 172.31.20.1 remote-as 100
 network 172.16.10.0
```



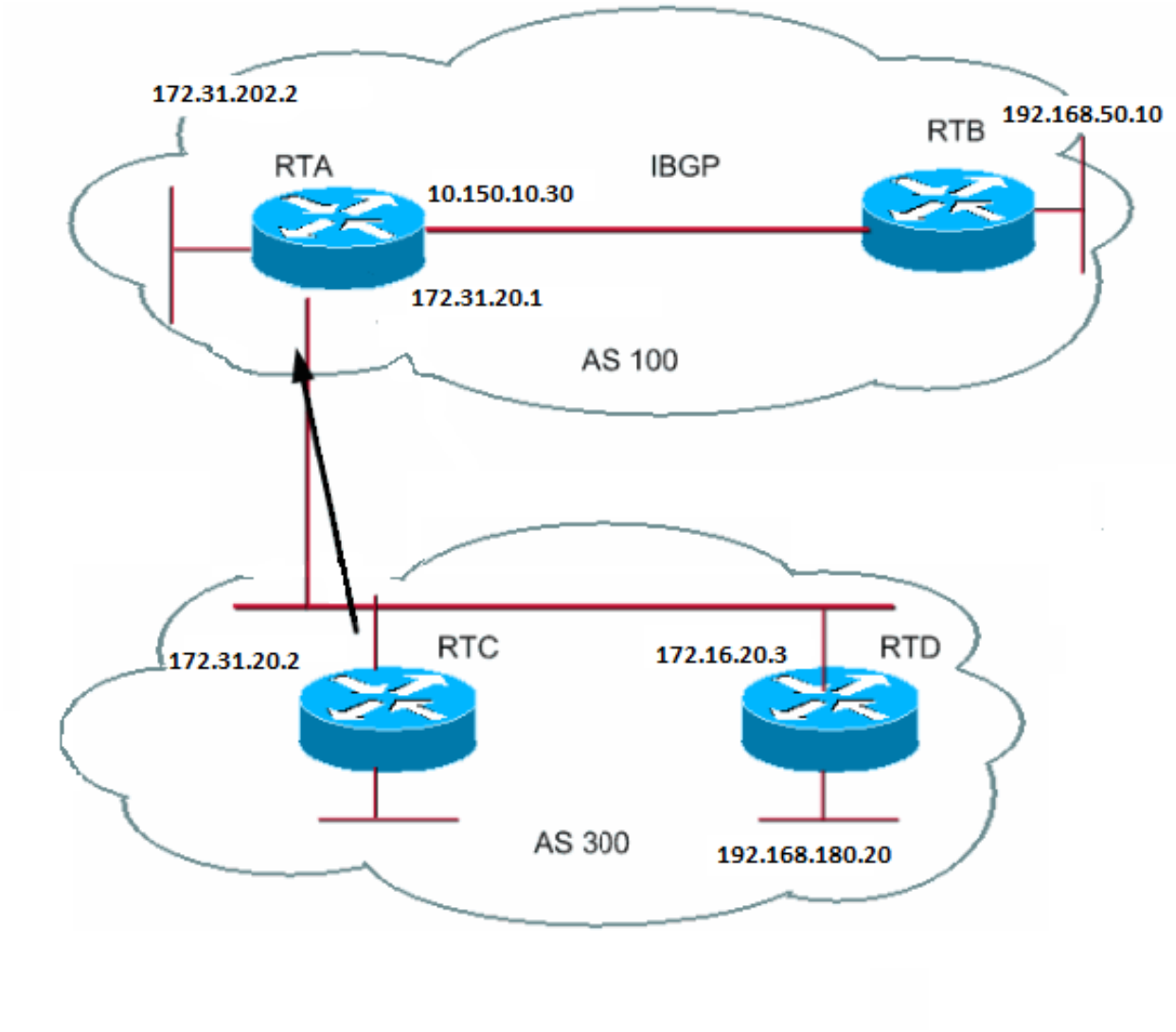
Remarque : RTC annonce 172.16.10.0 au RTA avec un prochain saut égal à 172.31.20.2.



Remarque : RTA annonce 172.16.10.0 au RTB avec un prochain saut égal à 172.31.20.2. Le prochain saut eBGP est effectué dans iBGP.

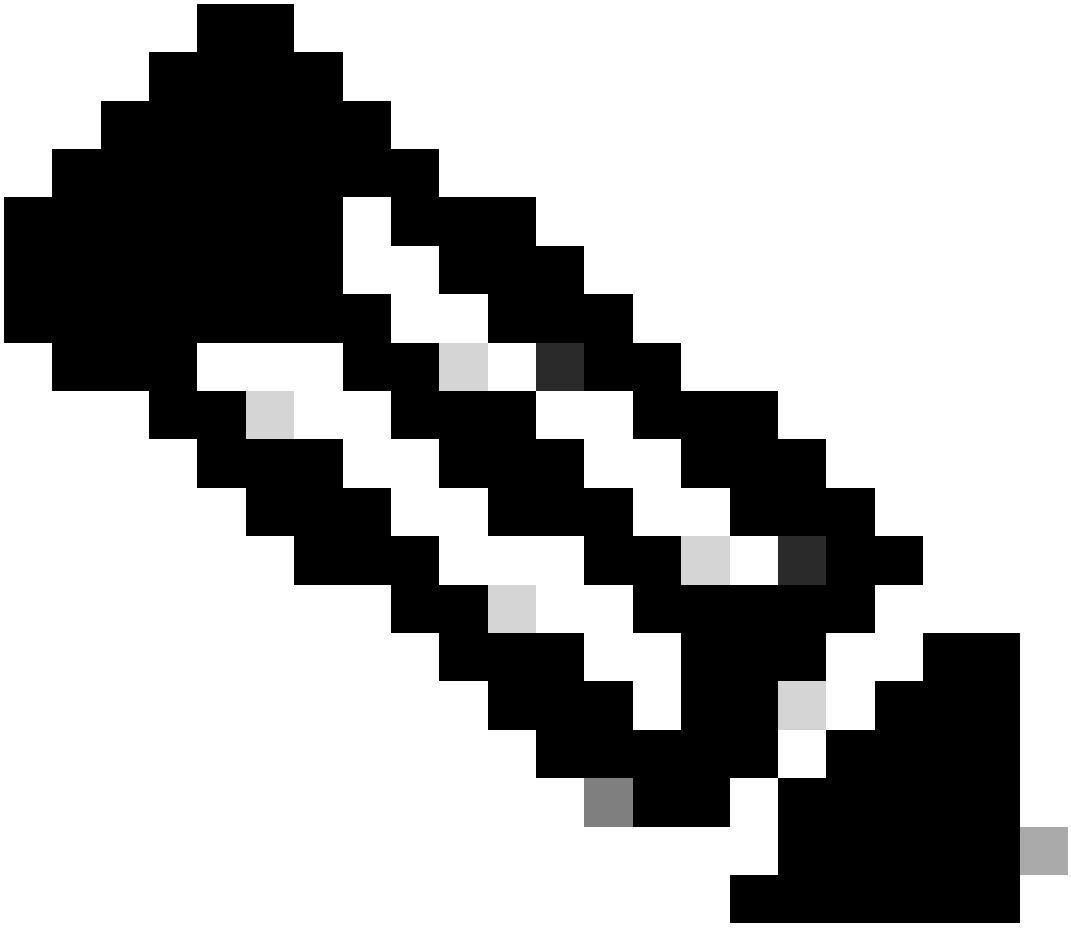
Faites particulièrement attention lorsque vous utilisez des réseaux multiaccès et des réseaux multiaccès sans diffusion (NBMA). Les sections sur le prochain saut BGP (réseaux multiaccès) et le prochain saut BGP (NBMA) donnent de plus amples renseignements à ce sujet.

Prochain saut BGP (réseaux multi-accès)



Cet exemple montre comment le prochain saut se comporte sur un réseau multi-accès tel qu'Ethernet.

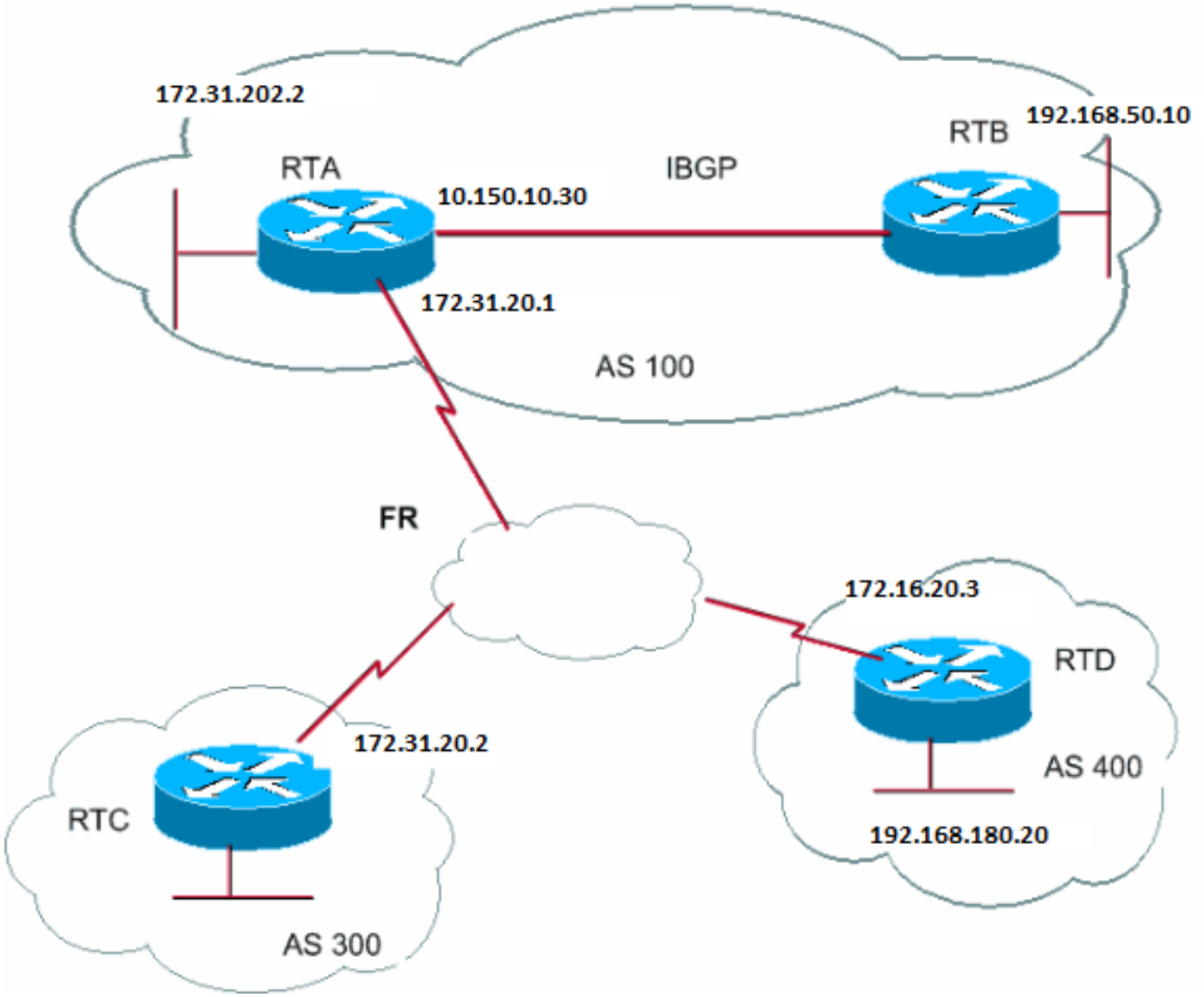
Supposons que RTC et RTD dans AS300 exécutent OSPF. RTC exécute BGP avec RTA. RTC peut atteindre le réseau 192.168.180.20 par l'intermédiaire de 172.16.20.3. Quand RTC envoie une mise à jour BGP à RTA concernant 192.168.180.20, RTC utilise comme prochain saut 172.16.20.3. RTC n'utilise pas sa propre adresse IP, 172.31.20.2. RTC utilise cette adresse parce que le réseau entre RTA, RTC et RTD est un réseau multi-accès. L'utilisation de RTA par RTD comme prochain saut pour atteindre 192.168.180.20 est plus raisonnable que le saut supplémentaire par l'intermédiaire de RTC.



Remarque : Le RTC annonce 192.168.180.20 au RTA avec un prochain saut 172.16.20.3.

Si le support commun à RTA, RTC, et RTD n'est pas de type multi-accès, mais NBMA, d'autres complications se produisent.

Prochain saut BGP (NBMA)



Le support commun apparaît sous la forme d'un nuage dans le diagramme. Si le support commun est un relais de trame ou n'importe quel nuage NBMA, le comportement exact est semblable à celui d'une connexion via Ethernet. RTC annonce 192.168.180.20 à RTA avec un prochain saut de 172.16.20.3.

Le problème est que RTA n'a pas un circuit virtuel permanent (PVC) à RTD et ne peut pas atteindre le prochain saut. Dans ce cas, le routage échoue.

La next-hop-selfcommande corrige cette situation.

Commande next-hop-self

Pour les situations avec le saut suivant, comme dans l'exemple BGP Next Hop (NBMA), vous pouvez utiliser la next-hop-self commande. La syntaxe est la suivante :

<#root>

```
neighbor {ip-address | peer-group-name} next-hop-self
```

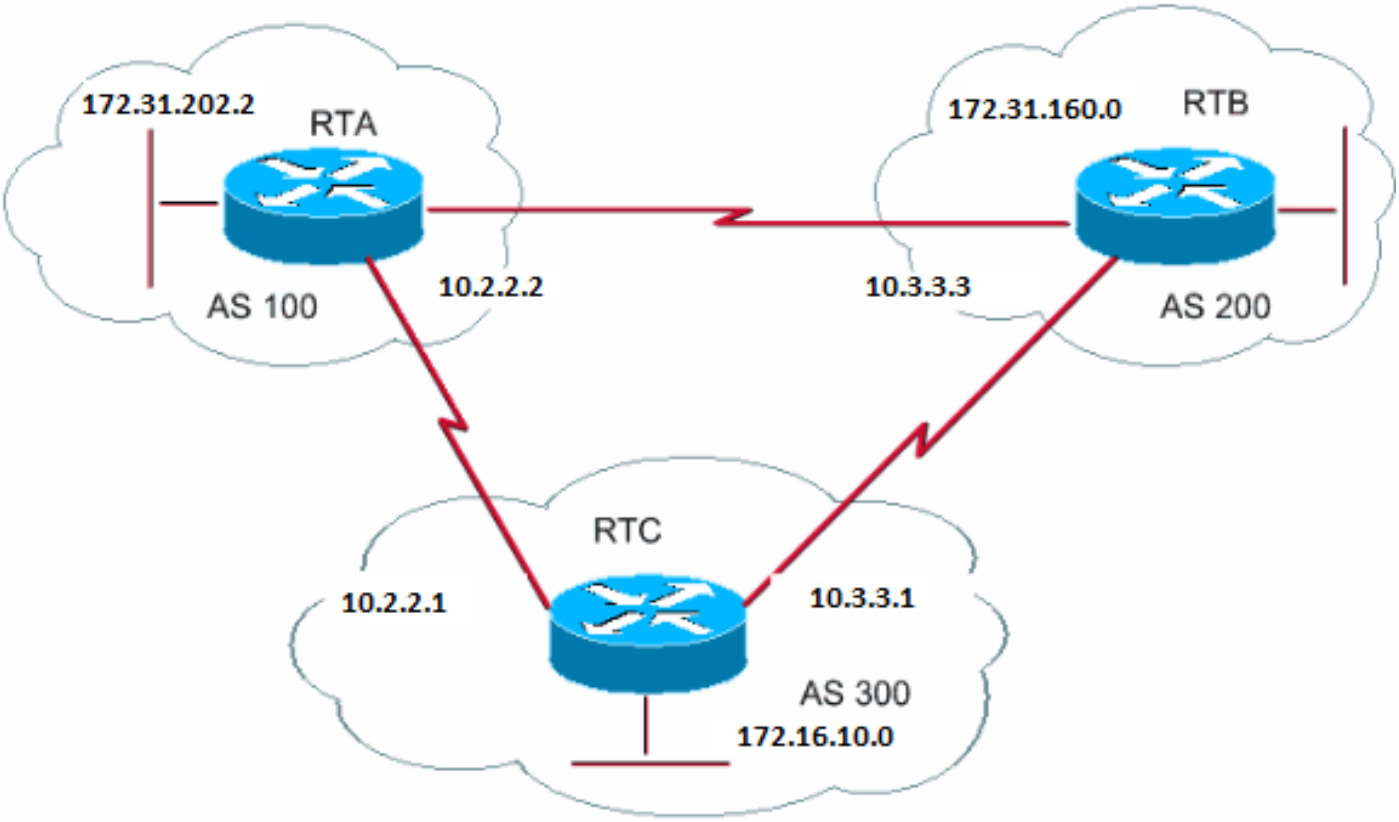
La next-hop-selfcommande vous permet de forcer BGP à utiliser une adresse IP spécifique comme tronçon suivant.

Pour l'exemple Prochain saut BGP (NBMA), cette configuration résout le problème :

```
RTC#
router bgp 300
 neighbor 172.31.20.1 remote-as 100
 neighbor 172.31.20.1 next-hop-self
```

RTA annonce 192.168.180.20 avec un prochain saut égal à 172.31.20.2.

Porte dérobée BGP



Dans le schéma précédent, RTA et RTC exécutent eBGP. RTB et RTC exécutent eBGP. RTA et RTB exécutent une sorte d'IGP (RIP ou IGRP)

ou un autre protocole. Par définition, les mises à jour eBGP ont une distance de 20, qui est inférieure aux distances IGP. Les distances par défaut sont :

- 120 pour RIP
- 100 pour IGRP
- 90 pour EIGRP
- 110 pour OSPF

RTA reçoit des mises à jour au sujet de 172.31.160.0 par l'intermédiaire de deux protocoles de routage :

- eBGP avec une distance de 20
- IGP avec une distance supérieure à 20

Par défaut, BGP utilise les distances suivantes :

- Distance externe - 20
- Distance interne - 200

- Distance locale - 200

Mais vous pouvez utiliser la distance commande pour modifier les distances par défaut :

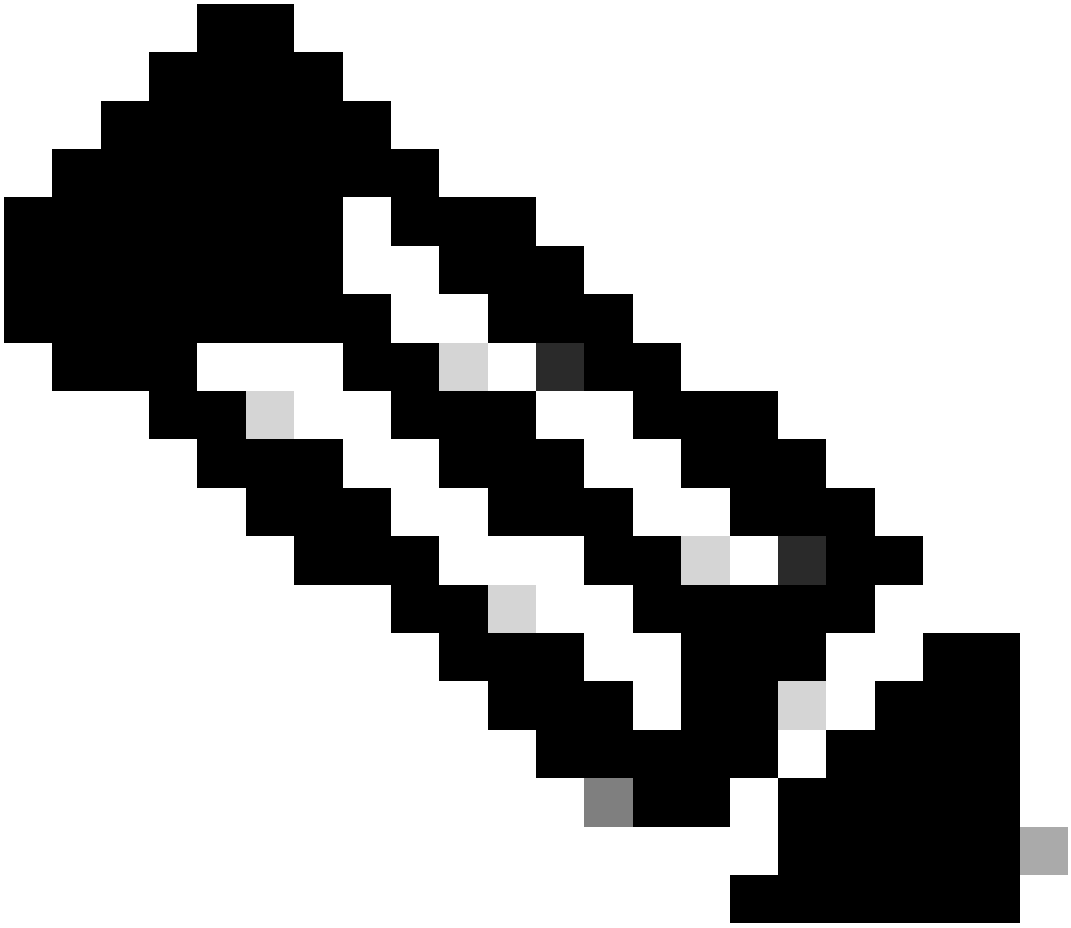
<#root>

distance bgp <external-distance> <internal-distance> <local-distance>

RTA sélectionne eBGP par l'intermédiaire de RTC en raison de sa distance inférieure.

Si vous voulez que RTA se renseigne sur 172.31.160.0 par l'intermédiaire de RTB (IGP), vous disposez de deux options :

- modifier la distance externe d'eBGP ou la distance IGP.



Remarque : Cette modification n’est pas recommandée.

-

Utilisez la porte dérobée BGP.

La porte dérobée BGP fait de la route IGP la route préférée.

Exécutez la commande **networkaddressbackdoor**.

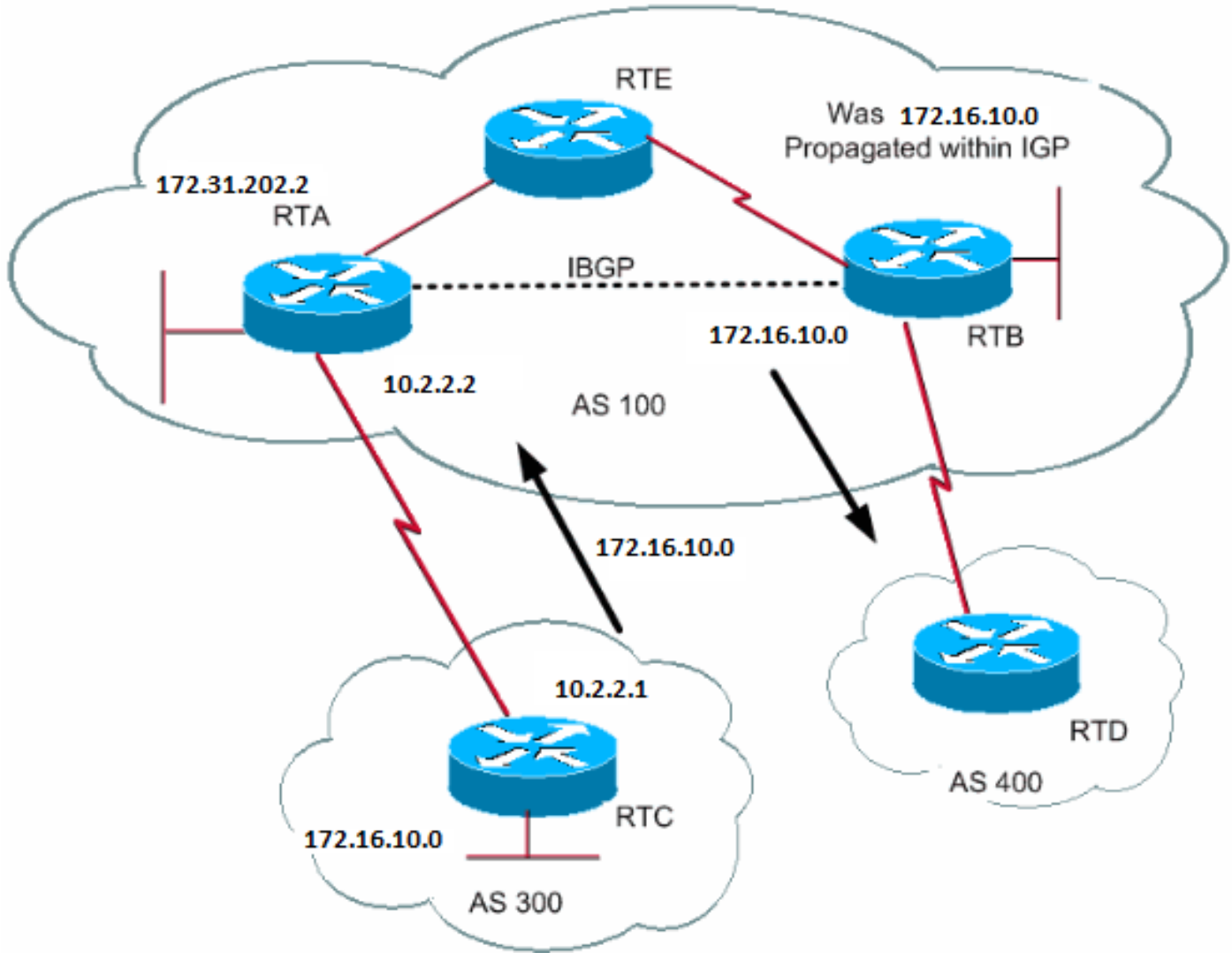
Le réseau configuré est le réseau qui vous voulez atteindre par l'intermédiaire d'IGP. Pour BGP, ce réseau bénéficie du même traitement qu'un réseau assigné localement, à ceci près que les mises à jour BGP n'annoncent pas ce réseau.

```
RTA#
router eigrp 10
 network 172.31.202.2

router bgp 100
 neighbor 10.2.2.1 remote-as 300
 network 172.31.160.0 backdoor
```

Le réseau 172.31.160.0 est traité comme une entrée locale, mais n’est pas annoncé comme une entrée de réseau normale.

RTA apprend 172.31.160.0 de RTB par l'intermédiaire d'EIGRP avec une distance de 90. RTA apprend également l'adresse de RTC par l'intermédiaire d'eBGP avec une distance de 20. Normalement, la préférence est eBGP, mais en raison de la commande **network backdoor**, EIGRP est à privilégier.

Synchronization



Avant la discussion sur la synchronisation, examinez ce scénario. RTC dans AS300 envoie des mises à jour au sujet de 172.16.10.0. RTA et RTB exécutent iBGP, par conséquent RTB obtient la mise à jour et peut atteindre 172.16.10.0 par l'intermédiaire du prochain saut 10.2.2.1. Rappelez-vous que le prochain saut est effectué par l'intermédiaire d'iBGP. Afin d'atteindre le prochain saut, RTB doit envoyer le trafic à RTE.

Supposons que RTA n'a pas redistribué le réseau 172.16.10.0 dans IGP. À ce niveau, RTE ne sait même pas que 172.16.10.0 existe.

Si le RTB commence à annoncer à AS400 que le RTB peut atteindre 172.16.10.0, le trafic qui passe du RTD au RTB avec la destination 172.16.10.0 est abandonné au RTE.

La synchronisation indique que, si votre AS achemine le trafic d’un autre AS vers un troisième AS, le BGP ne doit pas annoncer de route avant que tous les routeurs de votre AS aient détecté l’existence de la route par l’IGP. BGP attend qu'IGP ait propagé la route au sein de l'AS. Ensuite, BGP annonce la route aux homologues externes.

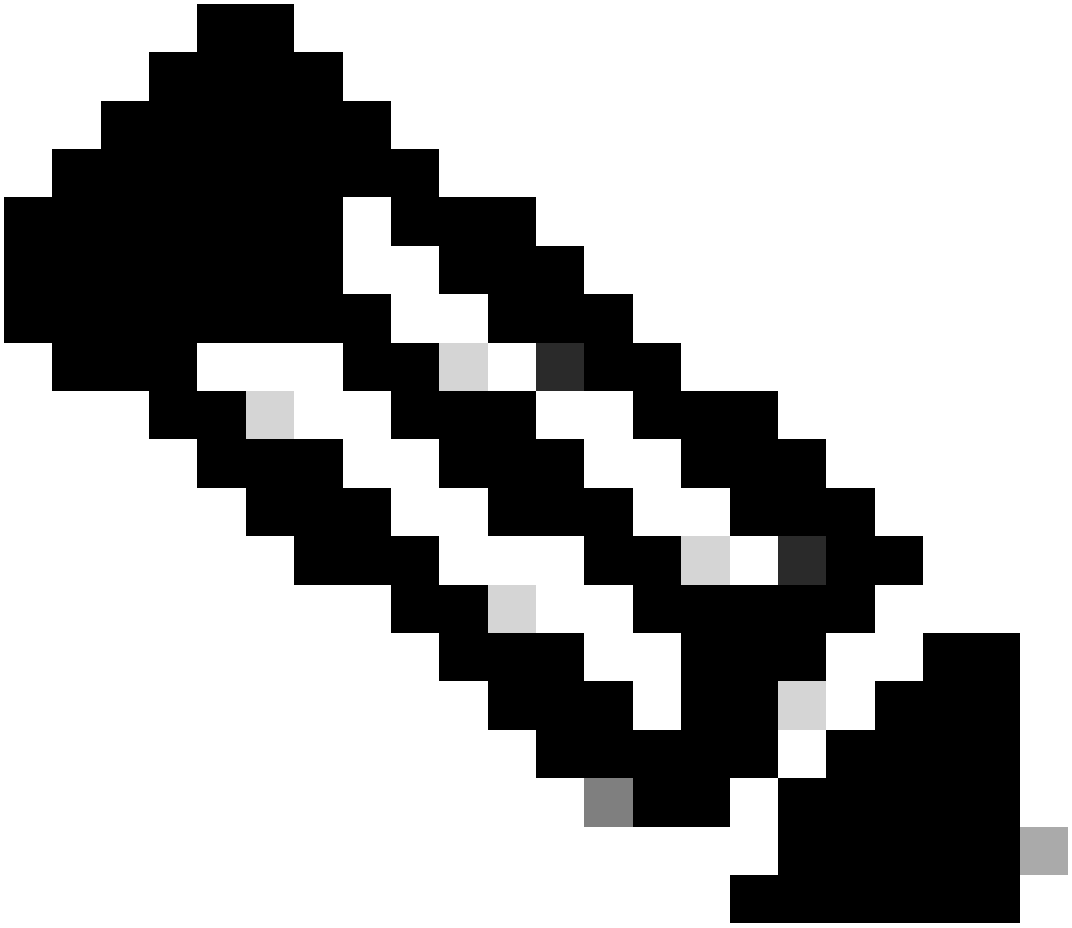
Dans l'exemple de cette section, RTB attend d'entendre parler de 172.16.10.0 par l'intermédiaire d'IGP. Ensuite, RTB commence à envoyer la mise à jour à RTD. Vous pouvez laisser croire à RTB qu'IGP a propagé l'information si vous ajoutez une route statique dans RTB qui pointe vers 172.16.10.0. Assurez-vous que les autres routeurs peuvent atteindre 172.16.10.0.

Désactiver la synchronisation

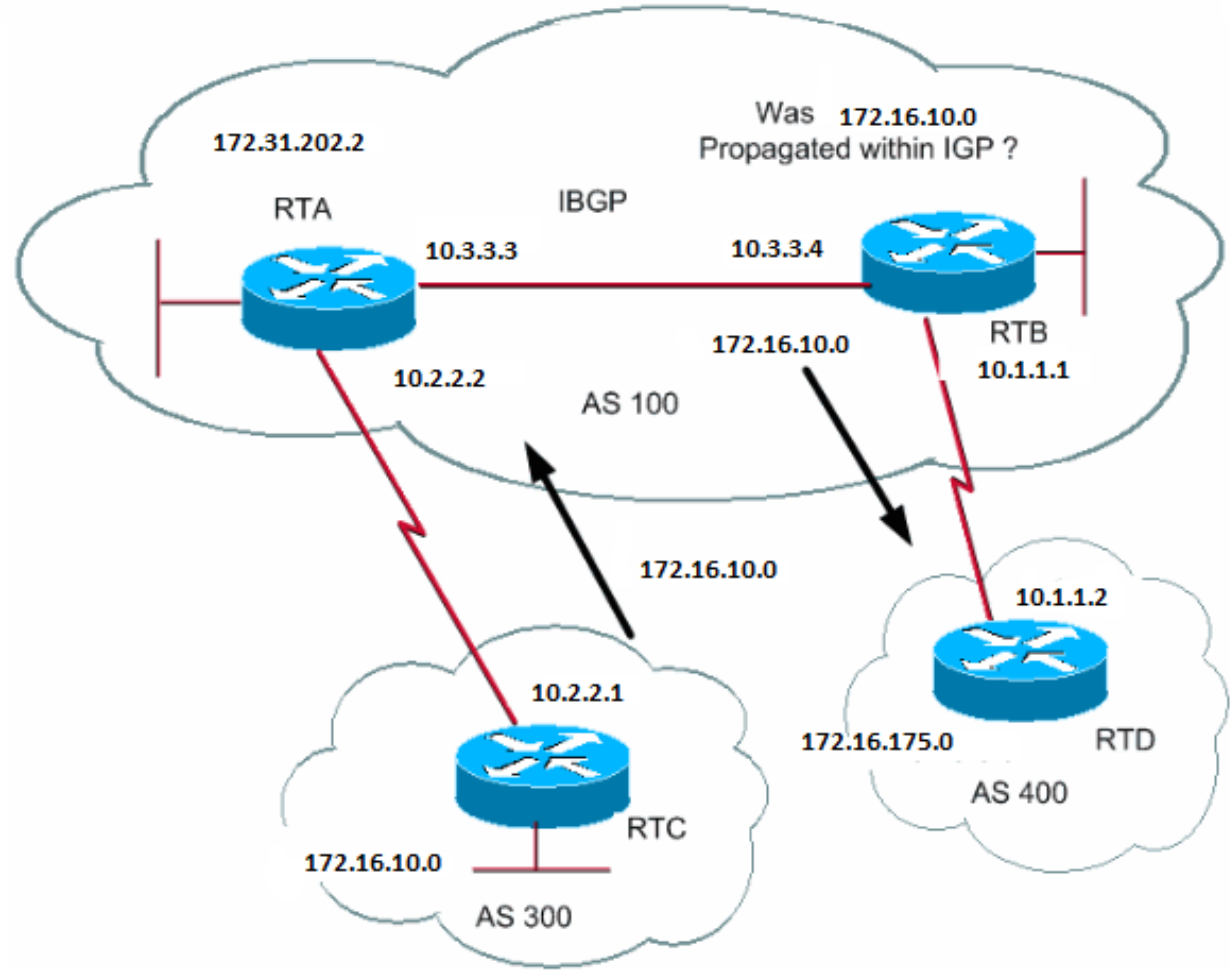
Dans certains cas, vous n'avez pas besoin de la synchronisation. Si aucun trafic issu d'un autre AS ne transite par votre AS, vous pouvez désactiver la synchronisation. Vous pouvez également désactiver la synchronisation si tous les routeurs de votre AS exécutent BGP. Grâce à la désactivation de cette fonctionnalité, vous pouvez gérer moins de routes dans votre IGP et permettre à BGP de converger plus rapidement.

La désactivation de la synchronisation n'est pas automatique. Si tous vos routeurs au sein de l'AS exécutent BGP et que vous n'exécutez pas du tout IGP, le routeur n'a aucun moyen de le savoir. Votre routeur attend indéfiniment une mise à jour IGP pour une route donnée avant d'envoyer la route aux homologues externes. Vous devez désactiver la synchronisation manuellement dans ce cas de sorte que le routage puisse fonctionner correctement :

```
router bgp 100
 no synchronization
```

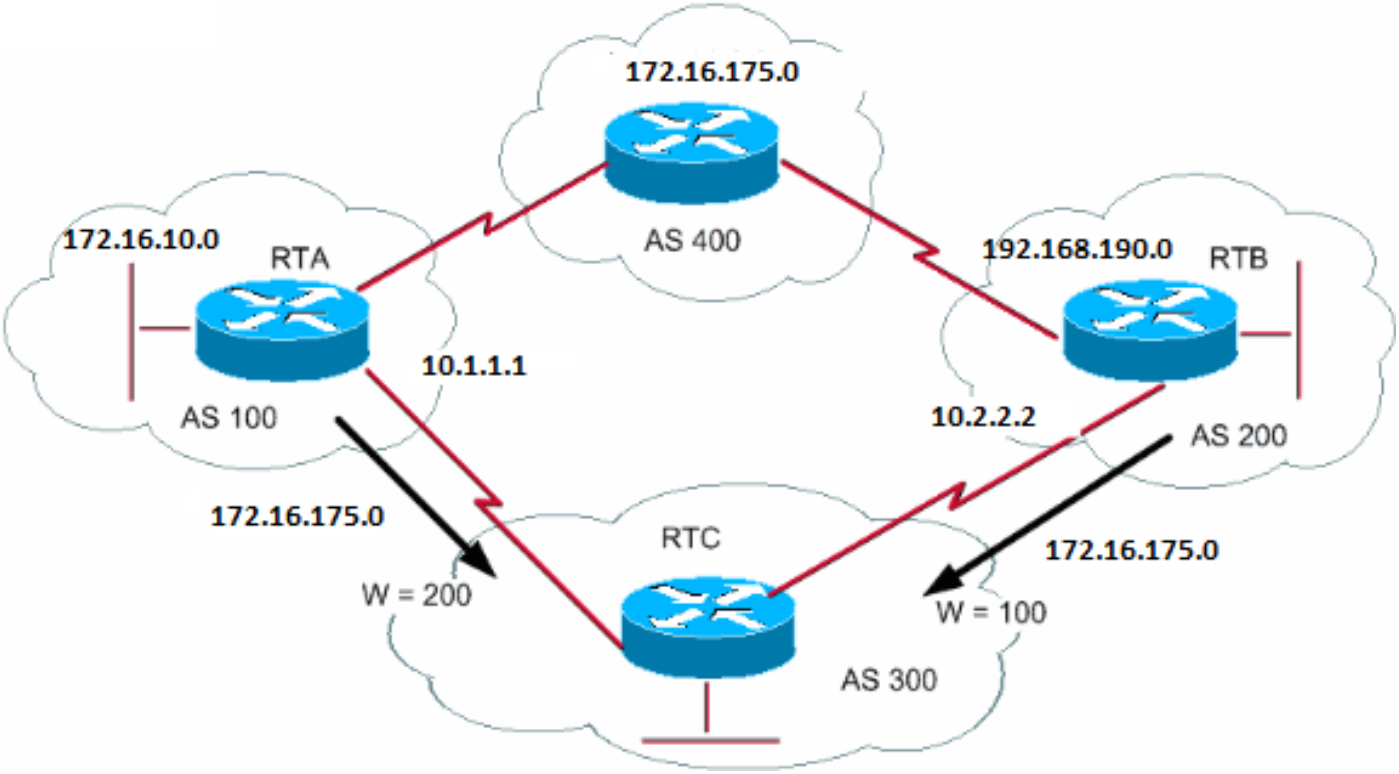


Remarque : Saisissez la commande « clear ip bgp address » pour réinitialiser la session.




```
RTB#
router bgp 100
 network 172.31.202.2
 neighbor 10.1.1.2 remote-as 400
 neighbor 10.3.3.3 remote-as 100
 no synchronization

!--- RTB puts 172.16.10.0 in its IP routing table and advertises the network
!--- to RTD, even if RTB does not have an IGP path to 172.16.10.0.

RTD#
router bgp 400
 neighbor 10.1.1.1 remote-as 100
 network 172.16.0.0

RTA#
router bgp 100
 network 172.31.202.2
 neighbor 10.3.3.4 remote-as 100
```

Attribut weight



L'attribut weight est un attribut défini par Cisco. Cet attribut utilise le poids pour sélectionner le meilleur chemin. Le poids est assigné localement au routeur. La valeur n'a de sens que pour ce routeur spécifique. La valeur n'est pas propagée ou transmise par les autres mises à jour de route. Un poids peut être un nombre entre 0 et 65 535. Les chemins initiés par le routeur ont un poids de 32 768 par défaut et les autres chemins ont un poids de 0.

Les routes avec une valeur de poids supérieure ont la préférence lorsqu'il existe plusieurs routes vers la même destination. Regardez l'exemple de cette section. RTA a appris le réseau 172.16.0.0 d'AS4. RTA propage la mise à jour à RTC. RTB a également appris le réseau 172.16.0.0 d'AS4. RTB propage la mise à jour à RTC. RTC dispose désormais de deux chemins pour atteindre 172.16.0.0 et doit en choisir un. Si vous définissez le poids des mises à jour sur RTC issues de RTA de manière à ce qu'il soit supérieur au poids des mises à jour issues de RTB, vous forcez RTC à utiliser RTA comme prochain saut pour atteindre 172.16.0.0. Plusieurs méthodes permettent de définir ce poids :

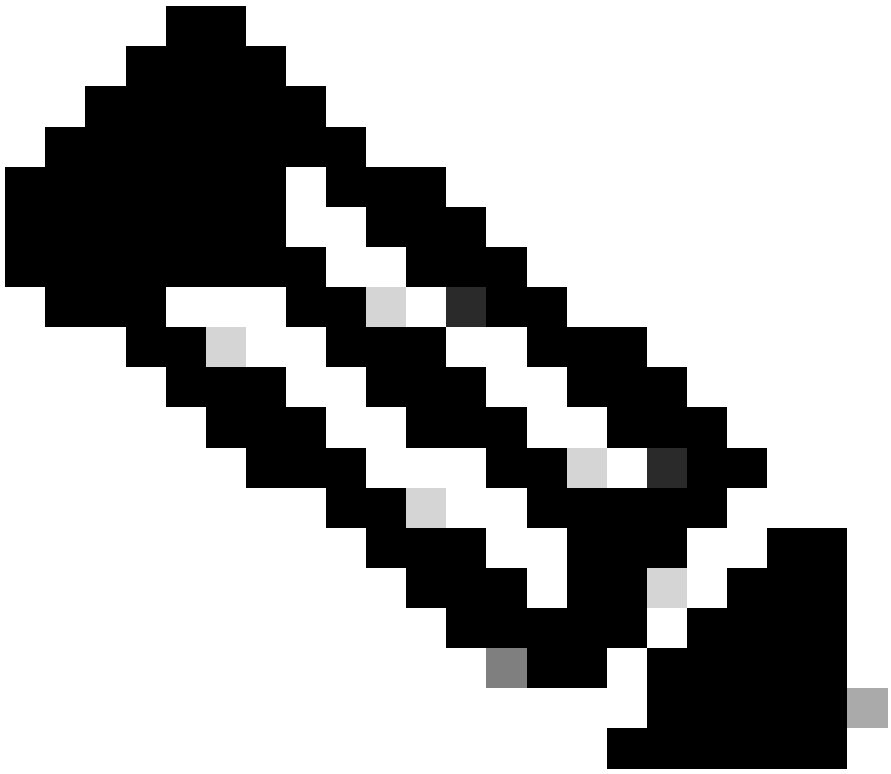
- Utilisez la commande neighbor.

  Pondération du voisin {ip-address|peer-group} <weight>

- Utilisez les listes d'accès AS_PATH.

ip as-path access-list <access-list-number>{permit | deny} <as-regular-expression>

Pondération du voisin <ip-address>filter-list <access-list-number> <weight>



Remarque : Dans certains scénarios, il peut y avoir très peu de commandes qui ne sont pas disponibles dans certaines versions de logiciels.

- Utilisez des mises en correspondance de route.

```
RTC#
router bgp 300
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 weight 200

!--- The route to 172.16.0.0 from RTA has a 200 weight.

 neighbor 10.2.2.2 remote-as 200
 neighbor 10.2.2.2 weight 100

!--- The route to 172.16.0.0 from RTB has a 100 weight.
```

RTA, qui a une valeur de poids supérieure, a la préférence comme prochain saut.

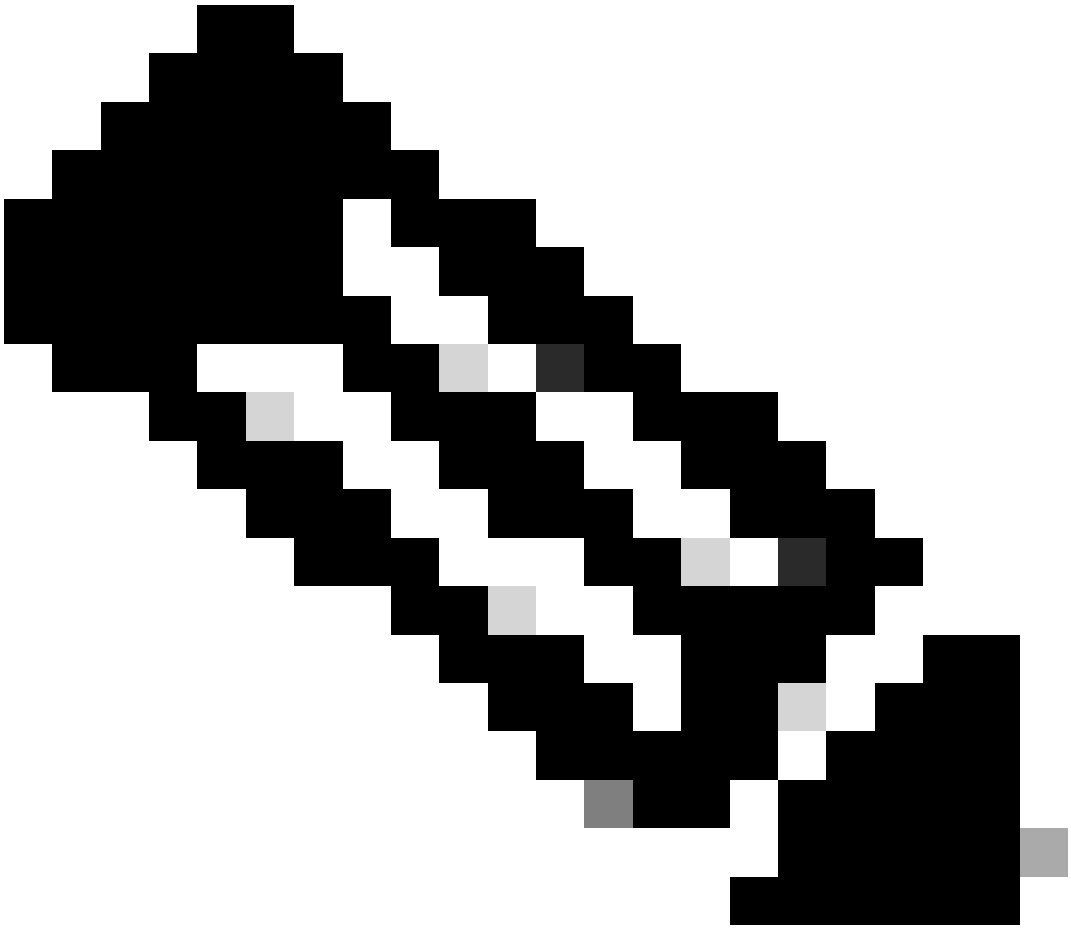
Vous pouvez obtenir les mêmes résultats avec l'IP AS_PATH et les listes de filtres.

```
RTC#
router bgp 300
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 filter-list 5 weight 200
 neighbor 10.2.2.2 remote-as 200
 neighbor 10.2.2.2 filter-list 6 weight 100
...
ip as-path access-list 5 permit ^100$

!--- This only permits path 100.

ip as-path access-list 6 permit ^200$
...
```

Vous pouvez également obtenir les mêmes résultats en utilisant des mises en correspondance de route.

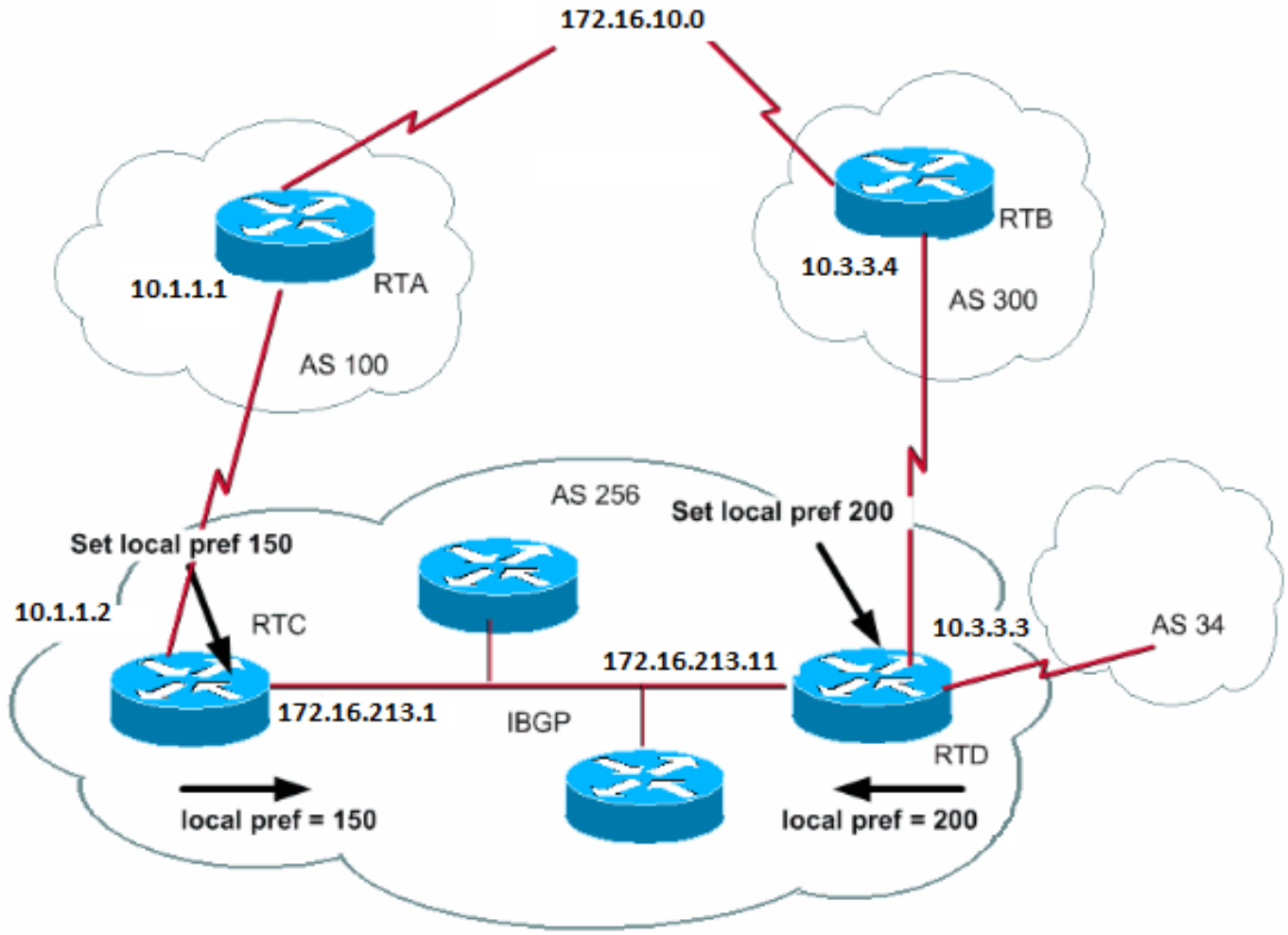
```
RTC#
router bgp 300
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 route-map setweightin in
 neighbor 10.2.2.2 remote-as 200
 neighbor 10.2.2.2 route-map setweightin in
...
ip as-path access-list 5 permit ^100$
```

```
...

route-map setweightin permit 10
 match as-path 5
 set weight 200

!--- Anything that applies to access list 5, such as packets from AS100, has weight 200.

route-map setweightin permit 20
 set weight 100

!--- Anything else has weight 100.
```



Remarque : Vous pouvez modifier la pondération pour choisir le chemin de BGP VPN MPLS avec le chemin IGP comme sauvegarde.

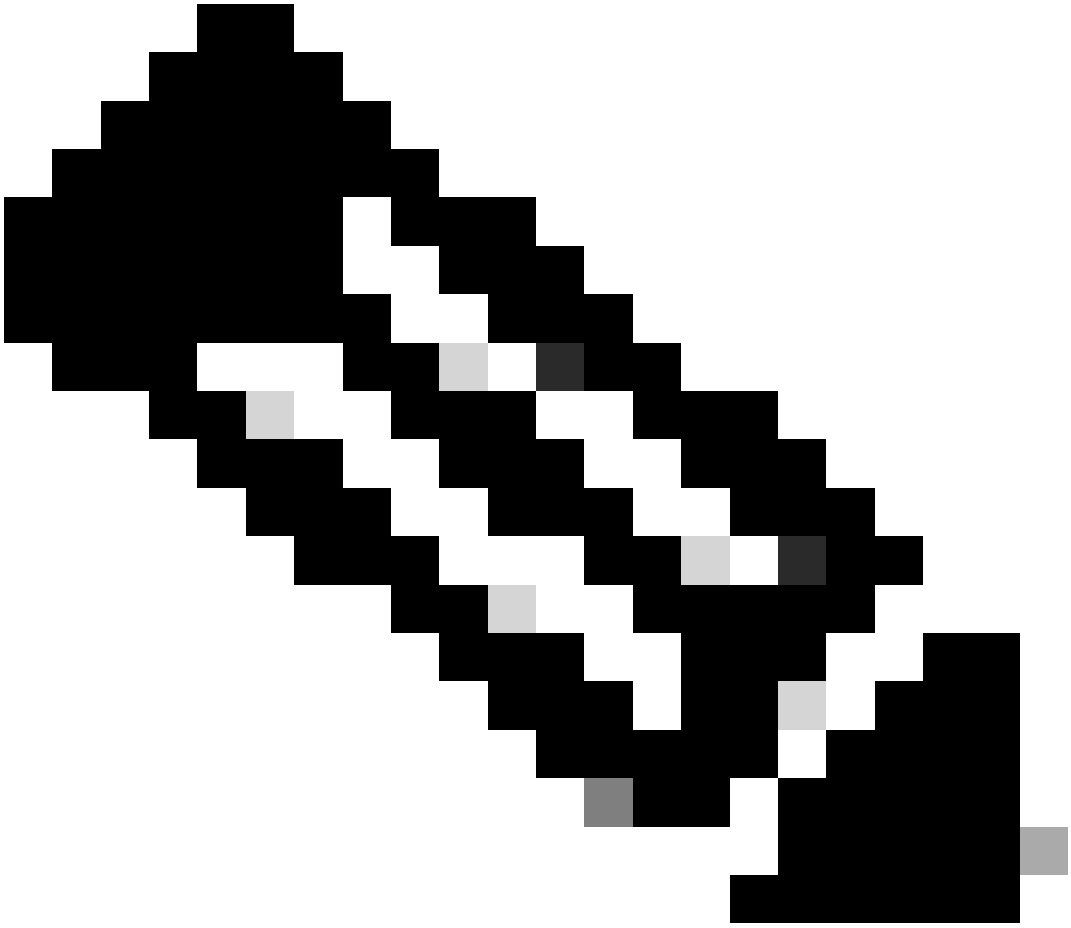
Attribut local preference



La préférence locale est une indication transmise à l'AS concernant le chemin préféré pour quitter l'AS afin d'atteindre un réseau donné. Un chemin avec une préférence locale plus élevée est préféré. La valeur par défaut de l'attribut local preference est 100.

À la différence de l'attribut weight, qui s'applique uniquement au routeur local, local preference est un attribut que les routeurs échangent au sein de l'AS.

Vous définissez la préférence locale avec l'émission de la commande bgp default local-preference value. Vous pouvez également définir la préférence locale à l'aide de mises en correspondance de route, comme le montre l'exemple de cette section :



Remarque : Il est nécessaire d’effectuer une réinitialisation logicielle (c’est-à-dire effacer le processus BGP sur le routeur) pour que les modifications soient prises en compte. Afin d'effacer le processus bgp, utilisez la clear ip bgp [soft][in/out] commande où soft indique une réinitialisation logicielle et ne déchire pas la session et [in/out] spécifie la configuration entrante ou sortante. Si la mention de « in/out » n’est pas précisée, les sessions entrantes et sortantes seront réinitialisées.

La commande bgp default local-preference définit la préférence locale sur les mises à jour hors des routeurs qui accèdent aux homologues du même AS. Dans le diagramme de cette section, AS256 reçoit des mises à jour au sujet de 172.16.10.0 de deux côtés différents de l'organisation. La préférence locale vous aide à déterminer comment quitter AS256 afin d'atteindre ce réseau. Supposons que RTD est le point de sortie préféré. Cette configuration définit la préférence locale pour les mises à jour issues d'AS300 sur 200 et pour les mises à jour issues d'AS100 sur 150 :

```
RTC#
router bgp 256
 neighbor 10.1.1.1 remote-as 100
  neighbor 10.213.11.2 remote-as 256
 bgp default local-preference 150

RTD#
router bgp 256
 neighbor 10.3.3.4 remote-as 300
 neighbor 10.213.11.1 remote-as 256
 bgp default local-preference 200
```

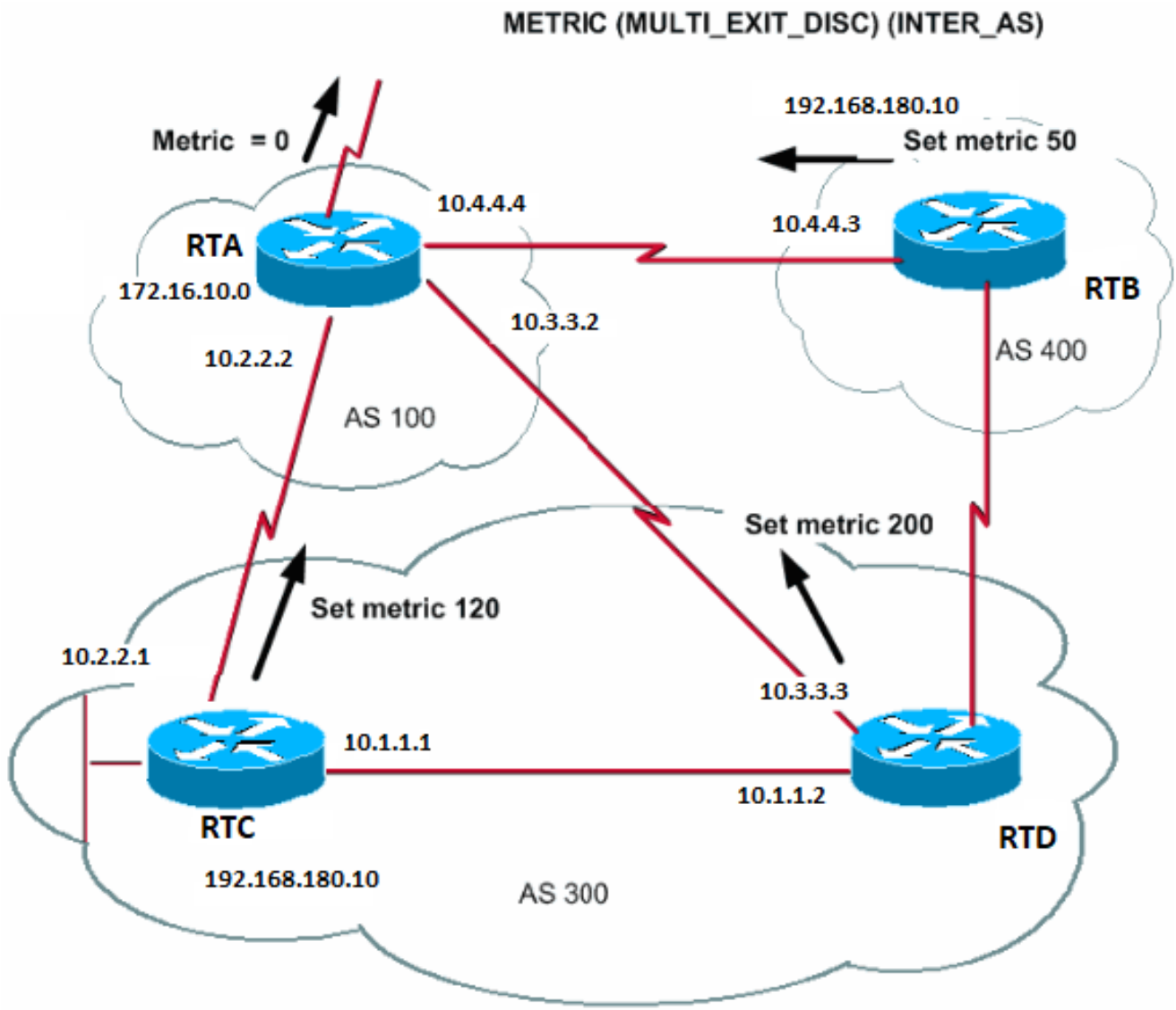
Dans cette configuration, RTC définit la préférence locale de toutes les mises à jour sur 150. Le même RTD définit la préférence locale de toutes les mises à jour sur 200. Il y a un échange de préférence locale dans AS256. Par conséquent, RTC et RTD se rendent compte que le réseau 172.16.10.0 a une préférence locale plus élevée quand les mises à jour viennent d'AS300 plutôt que d'AS100. Tout le trafic dans AS256 qui a ce réseau comme destination transmet avec RTD en tant que point de sortie.

L'utilisation de mises en correspondance de route offre plus de souplesse. Dans l'exemple de cette section, toutes les mises à jour reçues par RTD sont marquées avec la préférence locale 200 quand elles atteignent RTD. Les mises à jour issues d'AS34 sont également marquées avec la préférence locale 200. Cette balise peut être inutile. Pour cette raison, vous pouvez utiliser des mises en correspondance de route pour spécifier les mises à jour spécifiques qui doivent être marquées avec une préférence locale spécifique. Voici un exemple :

```
RTD#
router bgp 256
 neighbor 10.3.3.4 remote-as 300
 neighbor 10.3.3.4 route-map setlocalin in
 neighbor 10.213.11.1 remote-as 256
....
ip as-path access-list 7 permit ^300$
...

route-map setlocalin permit 10
 match as-path 7
 set local-preference 200

route-map setlocalin permit 20
 set local-preference 150
```

Avec cette configuration, toute mise à jour issue d'AS300 a une préférence locale de 200. Toutes les autres mises à jour, telles que les mises à jour issues d'AS34, ont une valeur de 150.

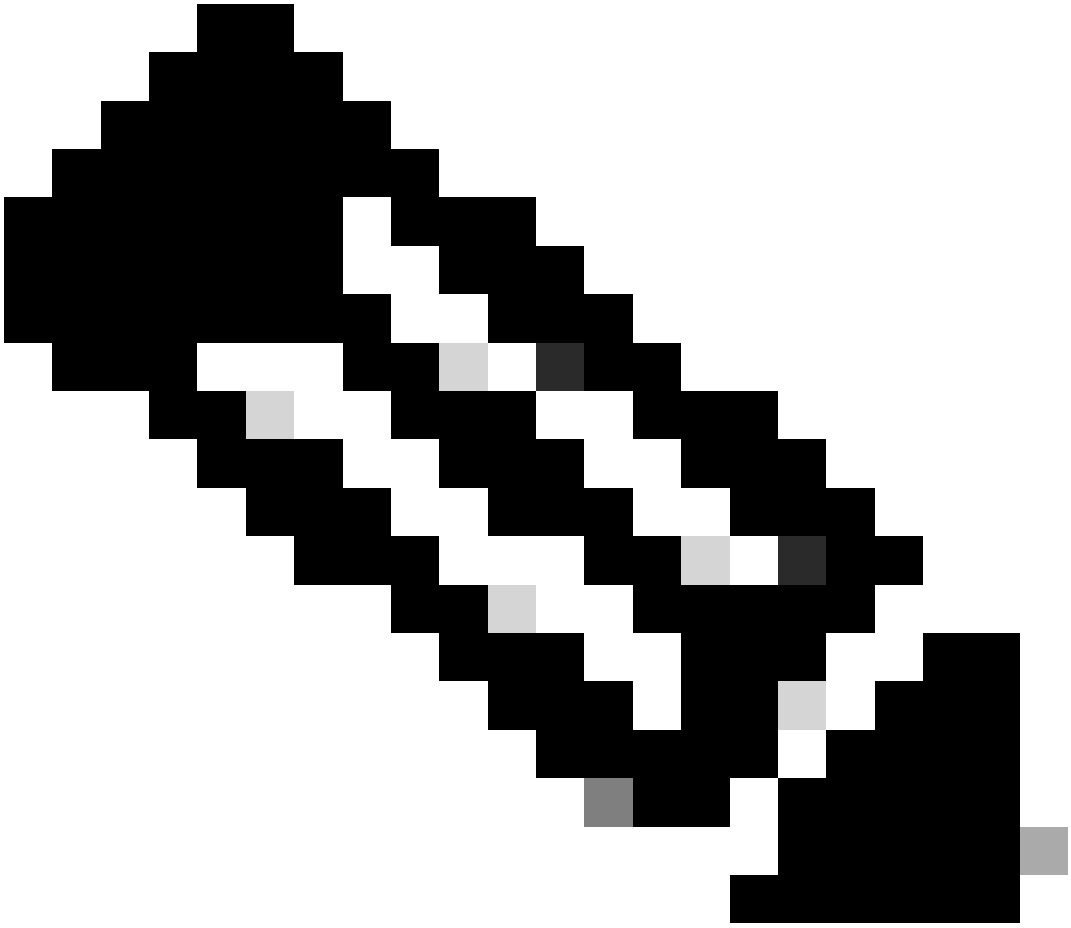
Attribut metric



L'attribut metric porte également le nom MULTI_EXIT_DISCRIMINATOR, MED (BGP4) ou INTER_AS (BGP3). L'attribut est un renseignement fourni aux voisins externes au sujet du chemin préféré dans un AS. Il permet d'influencer dynamiquement l'autre AS concernant l'accès à une route donnée lorsque l'AS comporte plusieurs points d'entrée. Une valeur inférieure est préférée pour l'attribut metric.

À la différence de la préférence locale, la métrique est échangée entre les AS. Une métrique est transmise à un AS mais ne quitte pas l'AS. Lorsqu'une mise à jour entre dans l'AS avec une métrique donnée, cette métrique est utilisée pour prendre des décisions au sein de l'AS. Quand cette même mise à jour est transmise à un troisième AS, cette métrique revient à 0. Le diagramme de cette section montre la configuration de la métrique. La valeur par défaut de l'attribut metric est 0.

À moins de recevoir d'autres instructions, le routeur compare les métriques des chemins des voisins dans le même AS. Pour que le routeur puisse comparer les métriques des voisins issus de différents AS, vous devez émettre la commande de configuration spéciale bgp always-compare-med sur le routeur.



Remarque : Deux commandes de configuration de BGP peuvent influer sur la sélection de chemin basée sur le discriminateur à plusieurs sorties (MED). Ces commandes sont la commande bgp deterministic-med et la commande bgp always-compare-med. L'émission de la commande bgp deterministic-med assure la comparaison de la variable MED pour le choix de la route lorsque plusieurs homologues annoncent dans le même AS. L'émission de la commande bgp always-compare-med assure la comparaison du MED des chemins des voisins situés dans différents AS. La commande bgp always-compare-med est utile quand plusieurs fournisseurs de service ou entreprises s'accordent sur une politique uniforme pour la configuration du MED. Référez-vous à Différence entre la commande bgp deterministic-med et la commande bgp always-compare-med pour comprendre comment ces commandes influencent la sélection des chemins BGP.

Dans le schéma de la présente section, AS100 obtient des renseignements sur le réseau 192.168.180.10 par l'intermédiaire de trois routeurs : RTC, RTD et RTB. RTC et RTD sont dans AS300, et RTB dans AS400.

Dans cet exemple, la comparaison du chemin d'accès de l'AS sur le RTA par la commande **bgp bestpath as-path ignore** est ignorée. La configuration est réalisée de manière à forcer le BGP à passer à l'attribut suivant pour la comparaison de la route (dans ce cas, la mesure ou le

MED). Si la commande est omise, le BGP peut installer la route 192.168.180.10 à partir du routeur RTC, car c'est celui qui a le chemin AS le plus court.

Supposons que vous avez défini la métrique provenant de RTC sur 120, la métrique provenant de RTD sur 200, et la métrique provenant de RTB sur 50. Par défaut, un routeur compare les métriques provenant des voisins situés dans le même AS. Par conséquent, RTA peut seulement comparer la métrique provenant de RTC avec la métrique provenant de RTD. RTA choisit RTC comme meilleur prochain saut parce que 120 est inférieur à 200. Lorsque le RTA reçoit une mise à jour du RTB avec la mesure 50, il ne peut pas comparer la mesure à 120, car le RTC et le RTB se trouvent dans des AS différents. Le RTA doit faire un choix en fonction d'autres attributs.

Afin de forcer RTA pour comparer les métriques, vous devez émettre la commande bgp always-compare-med sur RTA. Les configuration suivantes illustrent ce processus :

```
RTA#
router bgp 100
 neighbor 10.2.2.1 remote-as 300
 neighbor 10.3.3.3 remote-as 300
 neighbor 10.4.4.3 remote-as 400
 bgp bestpath as-path ignore

RTC#
router bgp 300
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 route-map setmetricout out
 neighbor 10.1.1.2 remote-as 300

route-map setmetricout permit 10
   set metric 120

RTD#
router bgp 300
 neighbor 10.3.3.2 remote-as 100
 neighbor 10.3.3.2 route-map setmetricout out
 neighbor 10.1.1.1 remote-as 300

route-map setmetricout permit 10
 set metric 200

RTB#
router bgp 400
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.4.4.4 route-map setmetricout out

route-map setmetricout permit 10
 set metric 50
```

Avec ces configurations, RTA sélectionne RTC comme prochain saut, en tenant compte du fait que tous les autres attributs sont identiques. Afin d'inclure RTB dans la comparaison métrique, vous devez configurer RTA de cette façon :

```
RTA#
router bgp 100
```

```
 neighbor 2.2.21 remote-as 300
 neighbor 10.3.3.3 remote-as 300
 neighbor 10.4.4.3 remote-as 400
 bgp always-compare-med
```

Dans ce cas, RTA sélectionne RTB comme meilleur prochain saut afin d'atteindre le réseau 192.168.180.10.

Vous pouvez également définir une mesure lors de la redistribution des routes dans BGP si vous exécutez la commande **default-metricnumber**.

Supposons que, dans l'exemple de cette section, RTB injecte un réseau par l'intermédiaire d'un chemin statique dans AS100. Voici la configuration :

```
RTB#
router bgp 400
 redistribute static
 default-metric 50

ip route 192.168.180.10 255.255.0.0 null 0

!--- This causes RTB to send out 192.168.180.10 with a metric of 50.
```

Attribut community

L'attribut community est un attribut transitif facultatif situé entre 0 et 4 294 967 200. L’attribut de communauté est un moyen de regrouper les destinations dans une communauté donnée et d’appliquer les décisions de routage qui correspondent aux communautés en question. Les décisions de routage sont accepter, préférer et redistribuer, pour n'en citer que quelques-unes.

Vous pouvez utiliser des mises en correspondance de route pour définir les attributs community. La commande de définition de la mise en correspondance de route a la syntaxe suivante :

<#root>

set community community-number [additive] [well-known-community]

Voici quelques communautés notoires prédéfinies à utiliser dans cette commande :

- **no-export : pas d'annonce aux homologues eBGP.** Gardez cette route dans un AS.

- **no-advertise : pas d'annonce de cette route aux homologues (internes ou externes).**

- **internet : annonce de cette route à la communauté Internet.** N'importe quel routeur appartient à cette communauté.

- **local-as : utilisé dans les scénarios de confédération pour empêcher la transmission de paquets en dehors des AS locaux.**

Voici deux exemples de mises en correspondance de route qui définissent la communauté :

```
route-map communitymap
 match ip address 1
 set community no-advertise
```

ou

```
route-map setcommunity
 match as-path 1
 set community 200 additive
```

Si vous ne définissez pas le mot clé additive, 200 remplace n'importe quelle communauté ancienne déjà existante. Si vous utilisez le mot clé additif, un ajout de 200 à la communauté se produit. Même si vous définissez l'attribut community, il n'est pas transmis aux voisins par défaut. Afin d'envoyer l'attribut à un voisin, vous devez utiliser cette commande :

neighbor {ip-address | peer-group-name} send-community

Voici un exemple :

```
RTA#
router bgp 100
 neighbor 10.3.3.3 remote-as 300
 neighbor 10.3.3.3 send-community
 neighbor 10.3.3.3 route-map setcommunity out
```

Dans le logiciel Cisco IOS, versions 12.0 et ultérieures, vous pouvez configurer des communautés dans trois formats différents : décimal, hexadécimal et AA:NN. Par défaut, le Logiciel Cisco IOS utilise le format décimal plus ancien. Afin de configurer et d'afficher dans AA:NN, émettez la commande **ip bgp-community new-**global configuration format. La première partie du format AA:NN représente le numéro de l'AS, tandis que la deuxième indique un numéro de 2 octets.

Voici un exemple :

Sans la commande ip bgp-community new-format en configuration globale, l'émission de la commande show ip bgp 10.6.0.0 affiche la valeur de l'attribut community au format décimal. Dans cet exemple, la valeur de l'attribut de communauté affiche 6553620.

<#root>

```
Router#
```

show ip bgp 10.6.0.0

```
BGP routing table entry for 10.6.0.0/8, version 7
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  1
    10.10.10.1 from 10.10.10.1 (10.255.255.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

Community: 6553620

Maintenant, émettez la commande ip bgp-community new-format globalement sur ce routeur.

<#root>

```
Router#
```

configure terminal

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

ip bgp-community new-format

```
Router(config)#
```

exit

Avec la commande de configuration globale **ip bgp-community new-format**, la valeur de communauté s'affiche dans le format AA:NN. La valeur s'affiche en format 100:20 dans la sortie de la commande **show ip bgp 10.6.0.0** de l'exemple suivant :

<#root>

```
Router#
```

show ip bgp 10.6.0.0

```
BGP routing table entry for 10.6.0.0/8, version 9
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  1
    10.10.10.1 from 10.10.10.1 (10.255.255.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

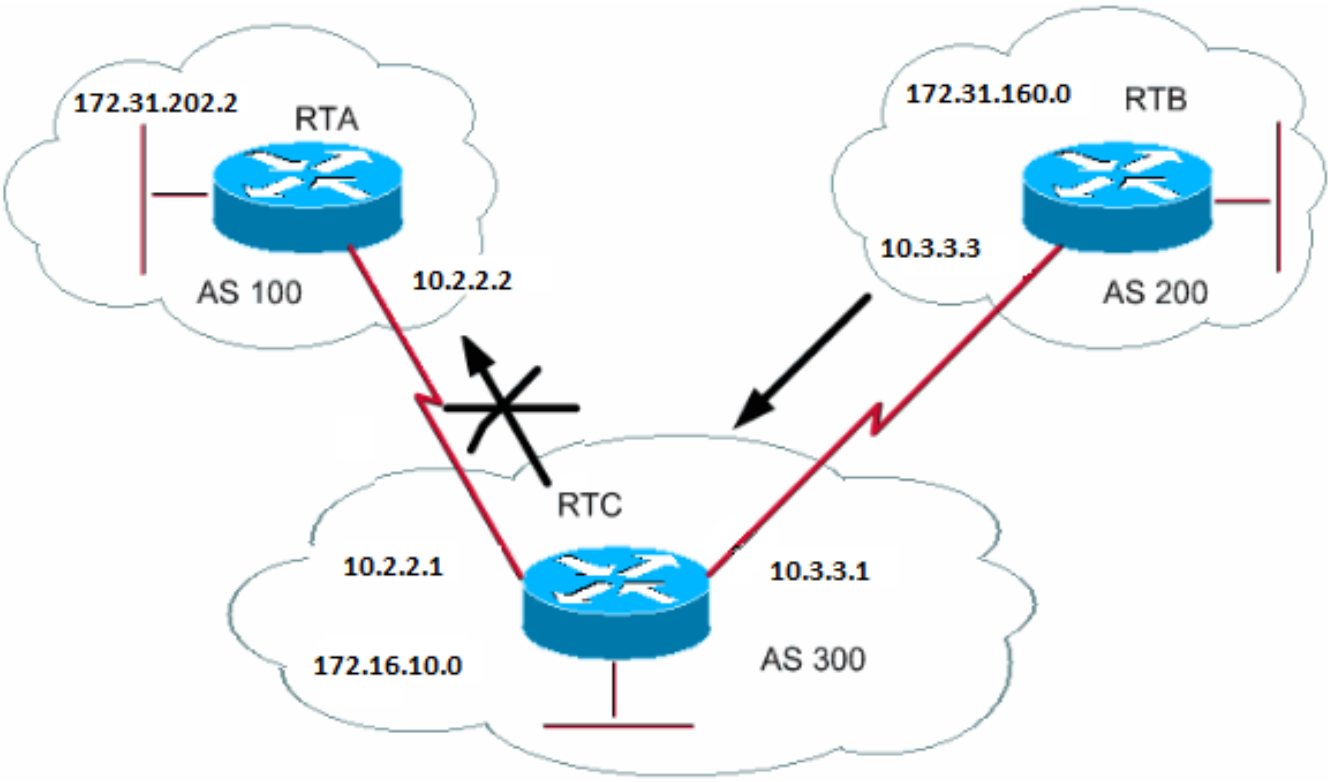
Community: 100:20

Études de cas BGP 3

Filtre BGP

Différentes méthodes de filtre vous permettent de contrôler l'envoi et la réception des mises à jour BGP. Vous pouvez filtrer les mises à jour BGP en utilisant les informations de route, les informations de chemin ou les communautés comme base. Toutes les méthodes permettent d'obtenir les mêmes résultats. Le choix d'une méthode plutôt qu'une autre dépend de la configuration du réseau spécifique.

Filtre de routage



Pour restreindre les informations de routage que le routeur apprend ou annonce, vous pouvez filtrer BGP en utilisant les mises à jour de routage à destination ou en provenance d'un voisin particulier. Vous définissez une liste d'accès et appliquez cette dernière aux mises à jour à destination en ou provenance d'un voisin. Émettez la commande suivante en mode de configuration du routeur :

<#root>

neighbor {ip-address | peer-group-name} distribute-list access-list-number {in | out}

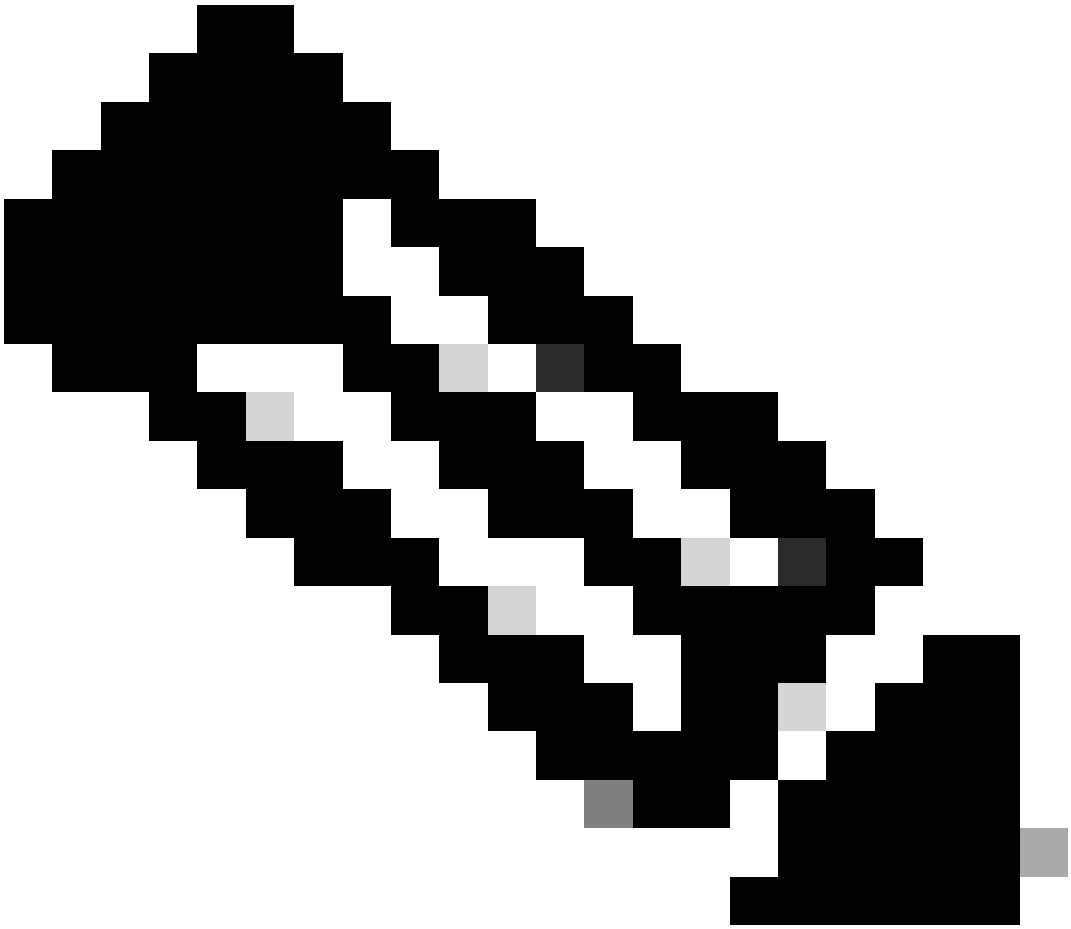
Dans cet exemple, RTB initie le réseau 172.31.160.0 et envoie la mise à jour à RTC. Si RTC veut arrêter la propagation des mises à jour à AS100, vous devez définir une liste d'accès pour filtrer ces mises à jour et appliquer la liste d'accès pendant la communication avec RTA :

```
RTC#
router bgp 300
```

```
 network 172.16.10.0
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 distribute-list 1 out

access-list 1 deny 172.31.160.0 0.0.255.255

access-list 1 permit 0.0.0.0 255.255.255.255

!--- Filter out all routing updates about 160.10.x.x.
```

L'utilisation des listes d'accès est un peu délicate quand vous gérez des super-réseaux qui peuvent entraîner des conflits.

Supposons que, dans l'exemple de cette section, RTB utilise différents sous-réseaux de 160.10.x.x. Votre objectif est de filtrer les mises à jour et d'annoncer uniquement 192.168.160.0/8.



Remarque : La notation /8 indique que vous utilisez 8 bits de filtre d’adresse locale, qui commencent à l’extrémité gauche de l’adresse IP. Cette adresse est équivalente à 192.168.160.0 255.0.0.0.

La commande access-list 1 permit 192.168.160.0 0.255.255.25 5autorise 192.168.160.0/8, 192.168.160.0/9, etc. Pour restreindre la mise à jour uniquement à 192.168.160.0/8, vous devez utiliser une liste d'accès étendue au format suivant :

<#root>

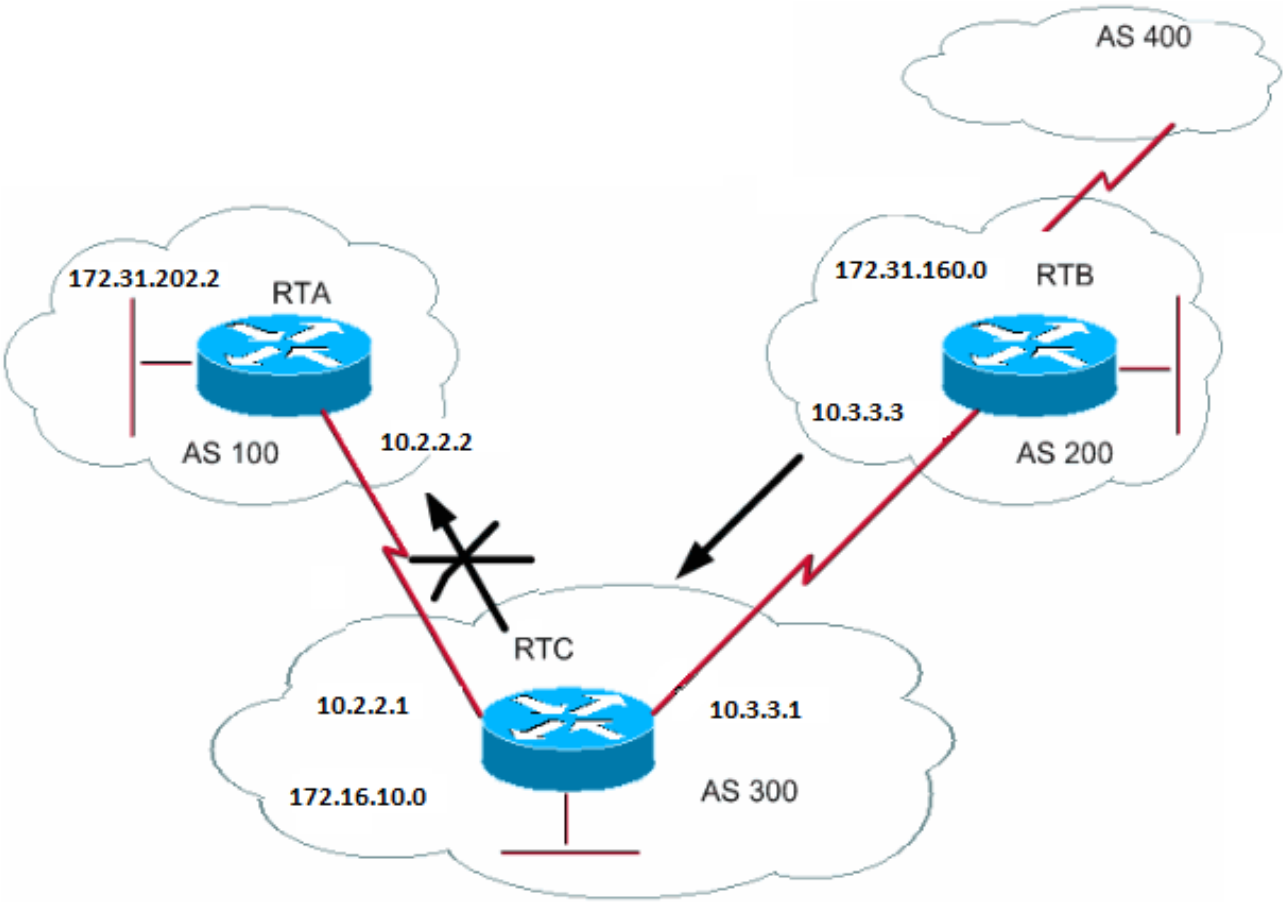
access-list 101 permit ip 192.168.160.0 0.255.255.255 255.0.0.0 0.0.0.0.

Cette liste autorise uniquement 192.168.160.0/8.

Consultez la section Block One or More Networks From a BGP Peer (bloquer un ou plusieurs réseaux contre un homologue BGP) pour voir des exemples de configuration sur la façon de filtrer les réseaux à partir des homologues BGP. La méthode utilise la commande **distribute-list** avec des listes de contrôle d’accès standard et étendues (ACL), en plus de la possibilité de filtrer la liste de préfixes.

Filtre de chemin

Vous pouvez également filtrer les chemins.



Vous pouvez spécifier une liste d'accès sur les mises à jour entrantes et sortantes à l'aide des informations des chemins d'AS BGP. Dans le diagramme de cette section, vous pouvez bloquer les mises à jour sur 172.31.160.0 de sorte qu'elles ne soient pas transmises à AS100. Pour bloquer les mises à jour, définissez une liste d'accès sur RTC qui empêche la transmission à AS100 de toutes les mises à jour en provenance d'AS200. Émettez les commandes suivantes :

<#root>

ip as-path access-list access-list-number {permit | deny} as-regular-expression

<#root>

neighbor {ip-address | peer-group-name} filter-list access-list-number {in | out}

Cet exemple interrompt l'envoi par RTC des mises à jour sur 172.31.160.0 à RTA :

```
RTC#
router bgp 300
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 filter-list 1 out

!--- The 1 is the access list number below.

ip as-path access-list 1 deny ^200$
ip as-path access-list 1 permit .*
```

Dans cet exemple, la access-list 1commande force le refus de toute mise à jour avec des informations de chemin qui commencent par 200 et se terminent par 200. ^200$ dans la commande est une « expression régulière » dans laquelle ^ signifie « commence par » et $ signifie « se termine par ». Comme le RTB envoie des mises à jour vers 172.31.160.0 avec des détails sur le chemin d'accès commençant et se terminant par 200, les mises à jour correspondent à la liste d'accès. La liste d'accès refuse ces mises à jour.

Le `.*` est une autre expression régulière dans laquelle le `.` indique « n'importe quel caractère » et le `*` signifie « la répétition de ce caractère ». Or, `.*` représente tout détail sur le chemin qui est nécessaire pour la transmission des autres mises à jour.

Que se passe-t-il si, au lieu d'utiliser `^200$`, vous utilisiez `^200`? Avec un AS400, comme le montre le diagramme de cette section, les mises à jour initiées par AS400 ont des informations de chemin de la forme (200, 400). Dans ces informations de chemin, 200 est premier et 400 est dernier. Ces mises à jour correspondent à la liste d'accès `^200`, car les détails sur le chemin commencent par 200. La liste d'accès empêche la transmission de ces mises à jour à RTA, ce qui n'est pas la condition requise.

Pour vérifier si vous avez mis en œuvre la bonne expression régulière, exécutez la commande **show ip bgp regexpregular-expression**. Cette commande montre tous les chemins correspondant à la configuration de l'expression régulière.

Expression régulière AS

Cette section explique la création d'une expression régulière.

Une expression régulière est un modèle à mettre en correspondance avec une chaîne d'entrée. Quand vous créez une expression régulière, vous spécifiez une chaîne à laquelle l'entrée doit correspondre. Dans le cas de BGP, vous spécifiez une chaîne qui se compose des informations de

chemin auxquelles une entrée doit correspondre.

Dans l’exemple de la section **Path Filter**, vous avez mentionné la chaîne `^200$`. Vous vouliez, pour décider, que les détails sur le chemin qui entrent dans les mises à jour correspondent à la chaîne.

Une expression régulière comporte les éléments suivants :

- **Plage**

  Une plage est une suite de caractères entre crochets gauche et droit. Par exemple, [abcd].

- **Atome**

  Un atome est un caractère unique. Voici quelques exemples :

`.`

- Le `.` correspond à un caractère unique.

`^`

- Le ^ correspond au début de la chaîne d'entrée.

`$`

- $ correspond à la fin de la chaîne d'entrée.

\

- Le \ correspond au caractère.

_

- Le _ correspond à une virgule (`,`), à une accolade gauche (`{`), à une accolade droite (`}`), au début de la chaîne d’entrée, à la fin de la chaîne d’entrée ou à une espace.

•

Pièce

Une pièce est un des symboles qui suivent un atome :

*

- * correspond à 0 ou à plusieurs séries de l'atome.

+

- + correspond à 1 ou à plusieurs séries de l'atome.

?

- Le ? correspond à la chaîne d’atome ou à la chaîne « null ».

- **Bureau régional**

Une branche est composée de 0 ou plusieurs parties concaténées.

Voici quelques exemples d'expressions régulières :

a*

- Cette expression indique n'importe quelle occurrence de la lettre « a », ce qui inclut l'absence de lettre.

```
a+
```

- Cette expression indique qu'au moins une occurrence de la lettre « a » doit être présente.

```
ab?a
```

- Cette expression correspond à « aa » ou « aba ».

```
_100_
```

- Cette expression signifie par l'intermédiaire d'AS100.

```
_100$
```

- Cette expression indique l'origine AS100.

```
^100 .*
```

- Cette expression indique une transmission depuis AS100.

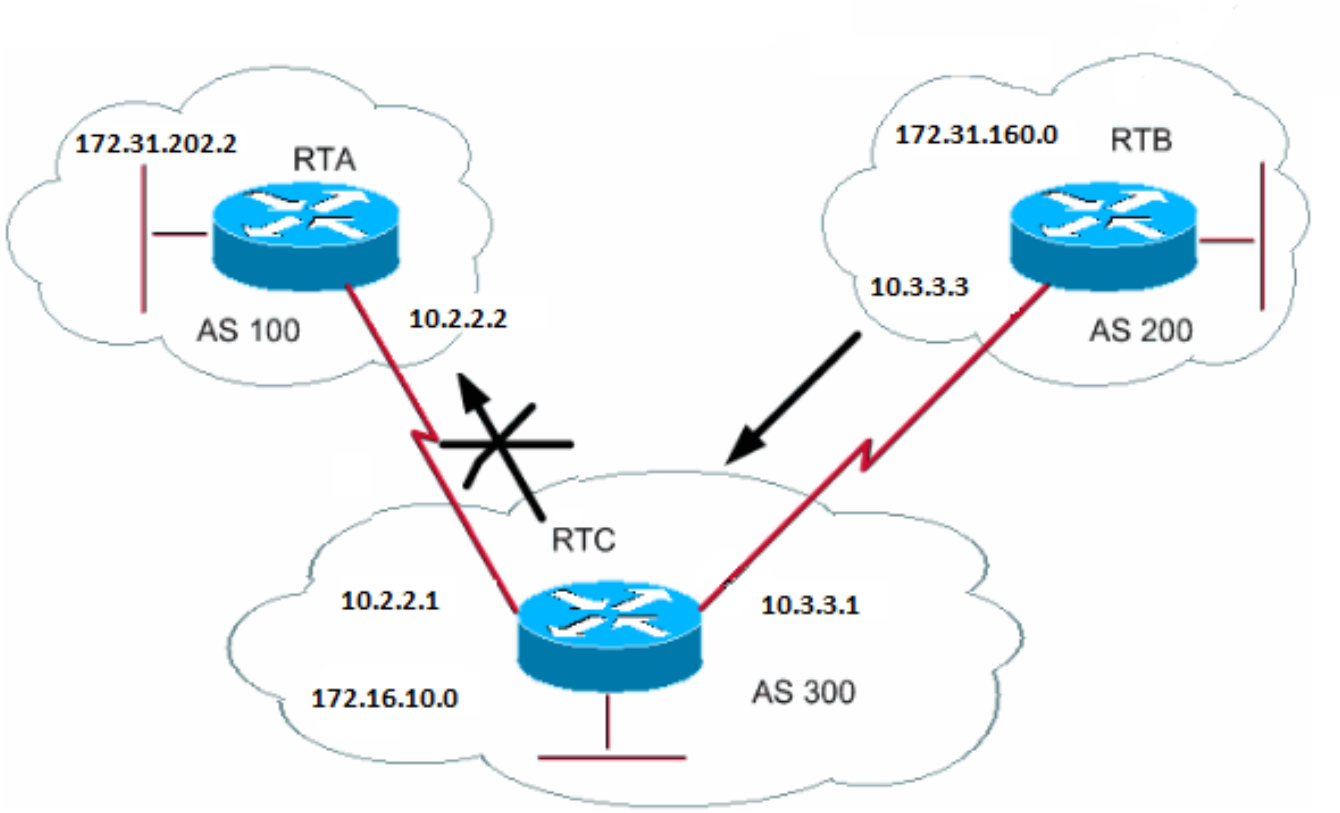
```
^$
```

- Cette expression indique une origine depuis cet AS.

Consultez la section Use Regular Expressions in BGP (utiliser les expressions régulières dans le BGP) pour voir des exemples de configuration concernant le filtrage des expressions régulières.

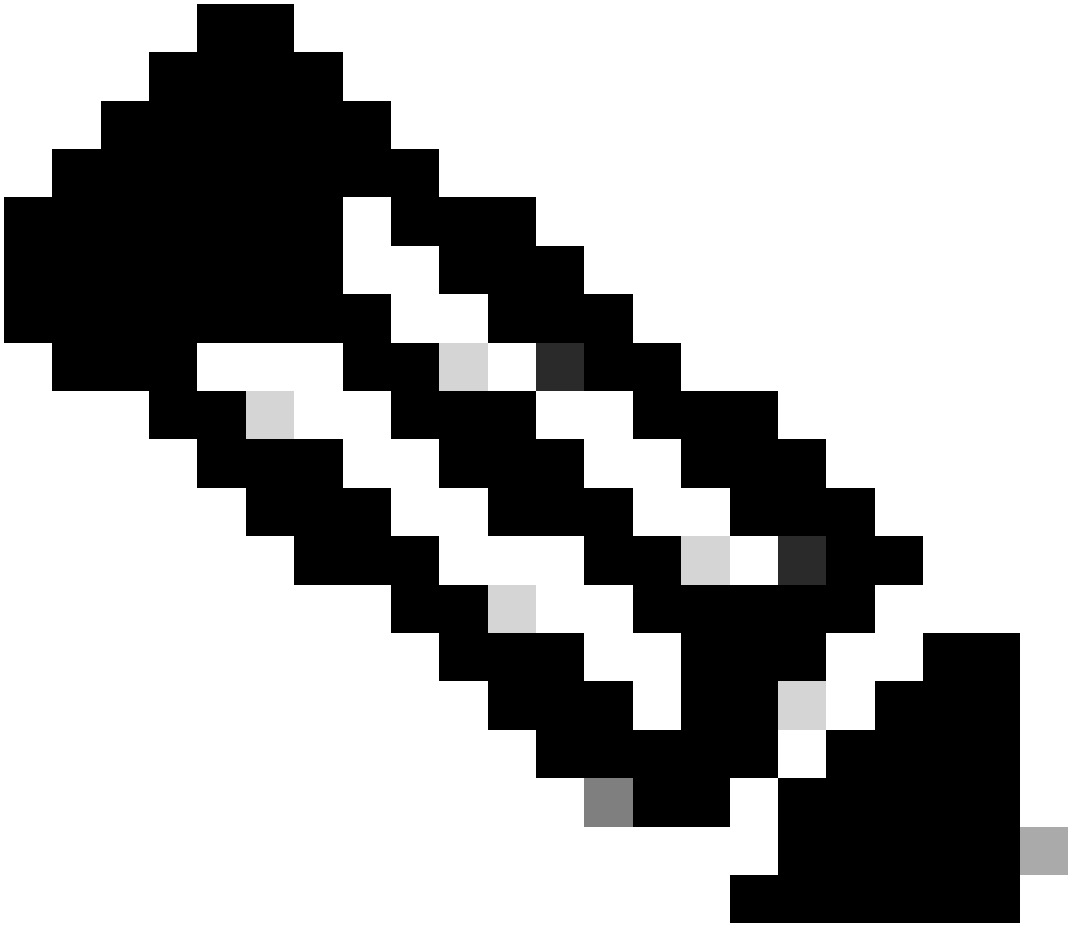
Filtre de communauté BGP

Ce document a couvert le filtrage de route et le filtre de chemin AS. Une autre méthode est le filtrage de la communauté. La section « Community Attribute » (attribut de la communauté) traite de la communauté et explique par des exemples comment utiliser cette communauté.

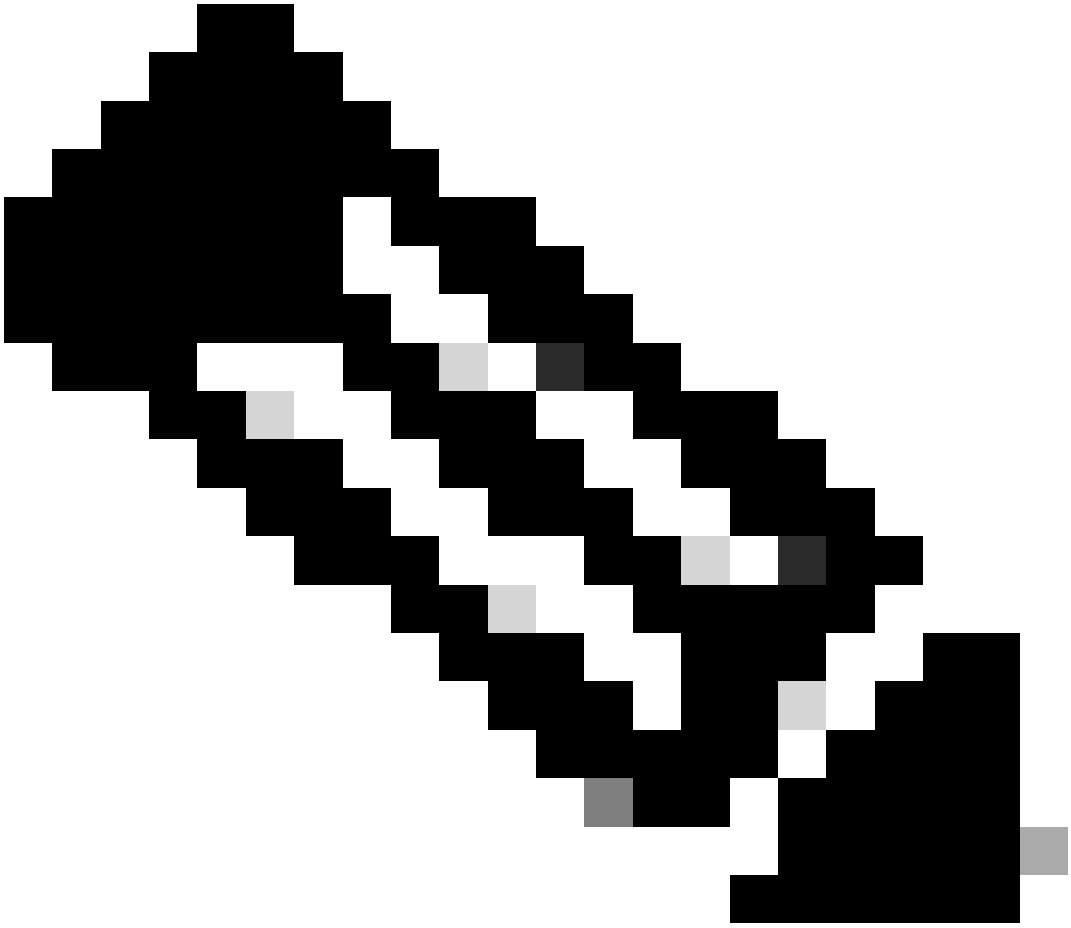


Dans cet exemple, vous voulez que RTB définisse l'attribut community sur les routes BGP que RTB annonce de sorte que RTC ne propage pas ces routes aux homologues externes. Utilisez l'attribut no-exportcommunity.

```
RTB#
router bgp 200
 network 172.31.160.0
 neighbor 10.3.3.1 remote-as 300
 neighbor 10.3.3.1 send-community
 neighbor 10.3.3.1 route-map setcommunity out

route-map setcommunity
 match ip address 1
 set community no-export

access-list 1 permit 0.0.0.0 255.255.255.255
```



Remarque : cet exemple utilise la commanderoute-map setcommunity afin de définir la communauté sur no-export.



Remarque : la **neighbor send-community** commande est nécessaire pour envoyer cet attribut à RTC.

Quand RTC obtient les mises à jour avec l'attribut NO_EXPORT, RTC ne propage pas les mises à jour à l'homologue externe RTA.

Dans cet exemple, RTB a défini l'attribut community sur**100 200 additive** . Cette action ajoute la valeur « 100 200 » à toute valeur de communauté actuelle avant la transmission au RTC.

```
RTB#
router bgp 200
 network 172.31.160.0
 neighbor 10.3.3.1 remote-as 300
```

```
 neighbor 10.3.3.1 send-community
 neighbor 10.3.3.1 route-map setcommunity out

route-map setcommunity
 match ip address 2
 set community 100 200 additive

access-list 2 permit 0.0.0.0 255.255.255.255
```

Une liste de communautés est un groupe de communautés que vous utilisez dans une clause match d'une mise en correspondance de route. La liste des communautés permet de filtrer ou définir les attributs avec différentes listes de numéros de communauté comme base.

<#root>

ip community-list <community-list-number> {permit | deny} <community-number>

Par exemple, vous pouvez définir cette mise en correspondance de route, match-on-community :

```
route-map match-on-community
 match community 10

!--- The community list number is 10.

 set weight 20
 ip community-list 10 permit 200 300

!--- The community number is 200 300.
```

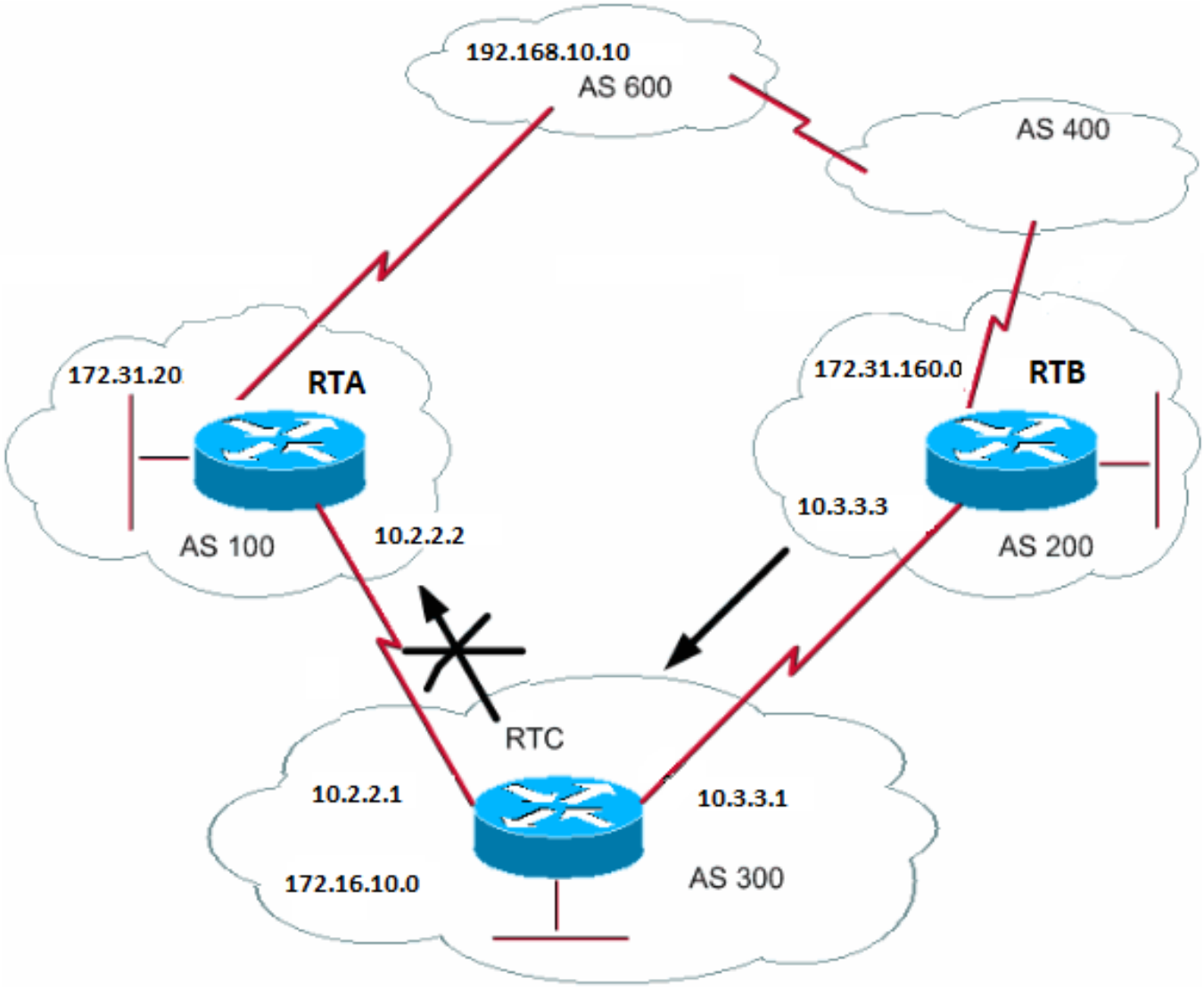
Vous pouvez employer la liste des communautés afin de filtrer ou définir certains paramètres, comme weight et metric, dans certaines mises à jour avec la valeur de la communauté comme base. Dans le second exemple de cette section, RTB a envoyé des mises à jour à RTC avec une communauté de 100 200. Si RTC veut définir le poids avec ces valeurs comme base, vous pouvez faire ceci :

```
RTC#
router bgp 300
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.3.3.3 route-map check-community in

route-map check-community permit 10
 match community 1
 set weight 20

route-map check-community permit 20
 match community 2 exact
 set weight 10

route-map check-community permit 30
 match community 3

ip community-list 1 permit 100
ip community-list 2 permit 200
ip community-list 3 permit internet
```

Dans cet exemple, une route qui a 100 dans l'attribut community correspond à la liste 1. Le poids de cette route est défini sur 20. Toute route qui a seulement 200 comme communauté correspond à la liste 2 et a un poids de 20. Le mot clé exact indique que la communauté se compose de 200 seulement et de rien d'autre. La dernière liste de communautés est ici pour s'assurer que d'autres mises à jour ne sont pas rejetées. Rappelez-vous que tout ce qui ne correspond pas est rejeté par défaut. Le mot clé internet indique toutes les routes parce que toutes les routes sont des membres de la communauté Internet.

Consultez Configure and Control an Upstream Provider Network with BGP Community Values (configurer et contrôler un réseau de fournisseur en amont avec les valeurs de communauté du BGP) pour en savoir plus.

Voisins BGP et mises en correspondance de route



Vous pouvez utiliser la commande neighbor en même temps que les mises en correspondance de route pour filtrer ou définir des paramètres sur des mises à jour entrantes et sortantes.

Les mises en correspondance de route associées à l'instruction neighbor n'exercent aucun effet sur des mises à jour entrantes quand vous utilisez une correspondance basée sur l'adresse IP :

<#root>

neighbor <ip-address> route-map <route-map-name>

Dans le diagramme de cette section, supposez que vous voulez que RTC apprenne d'AS200 les réseaux qui sont locaux à AS200 et rien d'autre. En outre, vous voulez définir le poids à 20 sur les routes acceptées. Utilisez une combinaison de listes d'accès neighbor et as-path :

```
RTC#
router bgp 300
 network 172.16.10.0
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.3.3.3 route-map stamp in

route-map stamp
 match as-path 1
 set weight 20

ip as-path access-list 1 permit ^200$
```

Toutes les mises à jour qui proviennent d'AS200 ont des informations de chemin qui commencent avec 200 et se terminent par 200. Ces mises à jour sont autorisées. Toute autre mise à jour est rejetée.

Supposez que vous voulez :

- une acceptation des mises à jour qui proviennent d'AS200 et ont un poids de 20 ;
- le rejet des mises à jour qui proviennent d'AS400 ;
- un poids de 10 pour d'autres mises à jour.

```
RTC#
router bgp 300
 network 172.16.10.0
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.3.3.3 route-map stamp in

route-map stamp permit 10
 match as-path 1
 set weight 20

route-map stamp permit 20
 match as-path 2
 set weight 10
```

```
ip as-path access-list 1 permit ^200$
ip as-path access-list 2 permit ^200 600 .*
```

Cette instruction définit un poids de 20 pour les mises à jour qui sont locales à AS200. L’instruction définit également une pondération de 10 pour les mises à jour qui se trouvent derrière l’AS400 et abandonne les mises à jour qui proviennent de l’AS400.

Utilisation de la commande set as-path prepend

Dans certaines situations, vous devez manipuler les informations de chemin afin de manipuler le processus de décision BGP. La commande que vous utilisez avec une mise en correspondance de route est :

<#root>

set as-path prepend <as-path#> <as-path#>

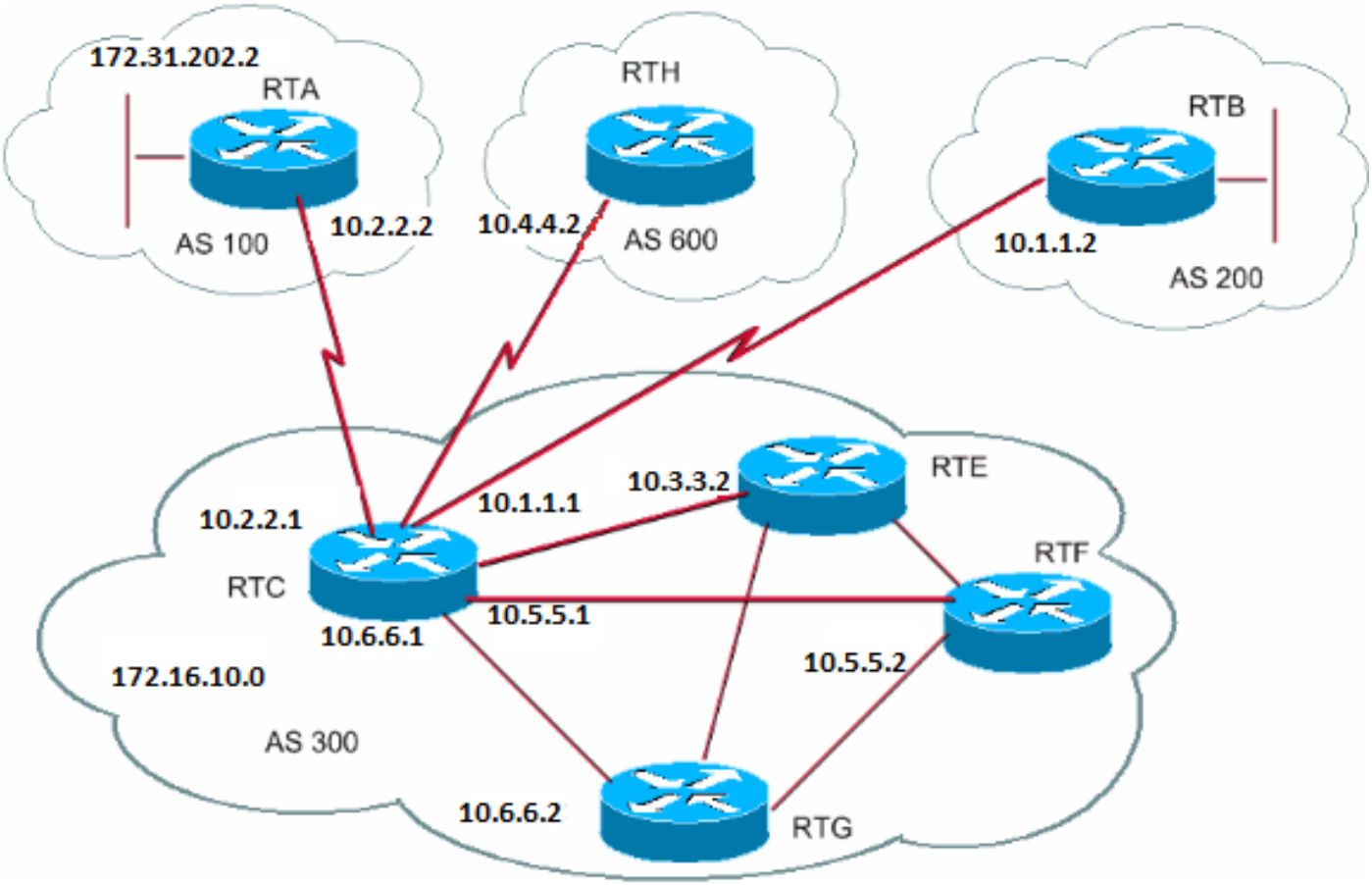
Supposons que, dans le schéma de la section BGP Neighbors and Route Maps (voisins du BGP et cartes de routage), le RTC annonce son réseau 172.16.10.0 à deux AS différents, AS100 et AS200. Quand l'information est propagée à AS600, les routeurs dans AS600 ont les informations d'accessibilité du réseau sur 172.16.10.0 par l'intermédiaire de deux routes différentes. La première route est via AS100 avec le chemin (100, 300), et la seconde est via AS400 avec le chemin (400, 200, 300). Si tous les autres attributs sont identiques, AS600 sélectionne le plus court chemin et choisit la route par l'intermédiaire d'AS100.

AS300 obtient tous les trafics par l'intermédiaire d'AS100. Si vous voulez influencer cette décision du côté d'AS300, vous pouvez faire en sorte que le chemin par AS100 semble plus long que le chemin qui passe par AS400. C’est ce que vous pouvez faire si vous ajoutez des numéros d’AS aux détails sur le chemin actuel qui sont annoncés à l’AS100. Une pratique courante est de répéter vos propres numéros AS de cette façon :

```
RTC#
router bgp 300
 network 172.16.10.0
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 route-map SETPATH out

route-map SETPATH
 set as-path prepend 300 300
```

Compte tenu de cette configuration, l'AS600 reçoit les mises à jour concernant 172.16.10.0 par l'AS100, accompagnées des détails sur le chemin de : (100, 300, 300, 300). Ces informations de chemin sont plus longues que le (400, 200, 300) que AS600 a reçu d'AS400.

Groupes d'homologues BGP



Un groupe d'homologues BGP est un groupe de voisins BGP avec la même stratégie de mise à jour. Les mises en correspondance de route, les listes de distribution et les listes de filtres définissent en général les stratégies de mise à jour. Vous ne définissez pas les mêmes politiques pour chaque voisin distinct, mais indiquez plutôt un nom de groupe d'homologues et affectez les politiques à ce groupe.

Les membres du groupe d'homologues héritent de toutes les options de configuration du groupe d'homologues. Vous pouvez également configurer des membres pour remplacer ces options si les options n'affectent pas des mises à jour sortantes. Vous pouvez seulement remplacer les options qui sont définies sur les données entrantes.

Afin de définir un groupe d'homologues, exécutez cette commande :

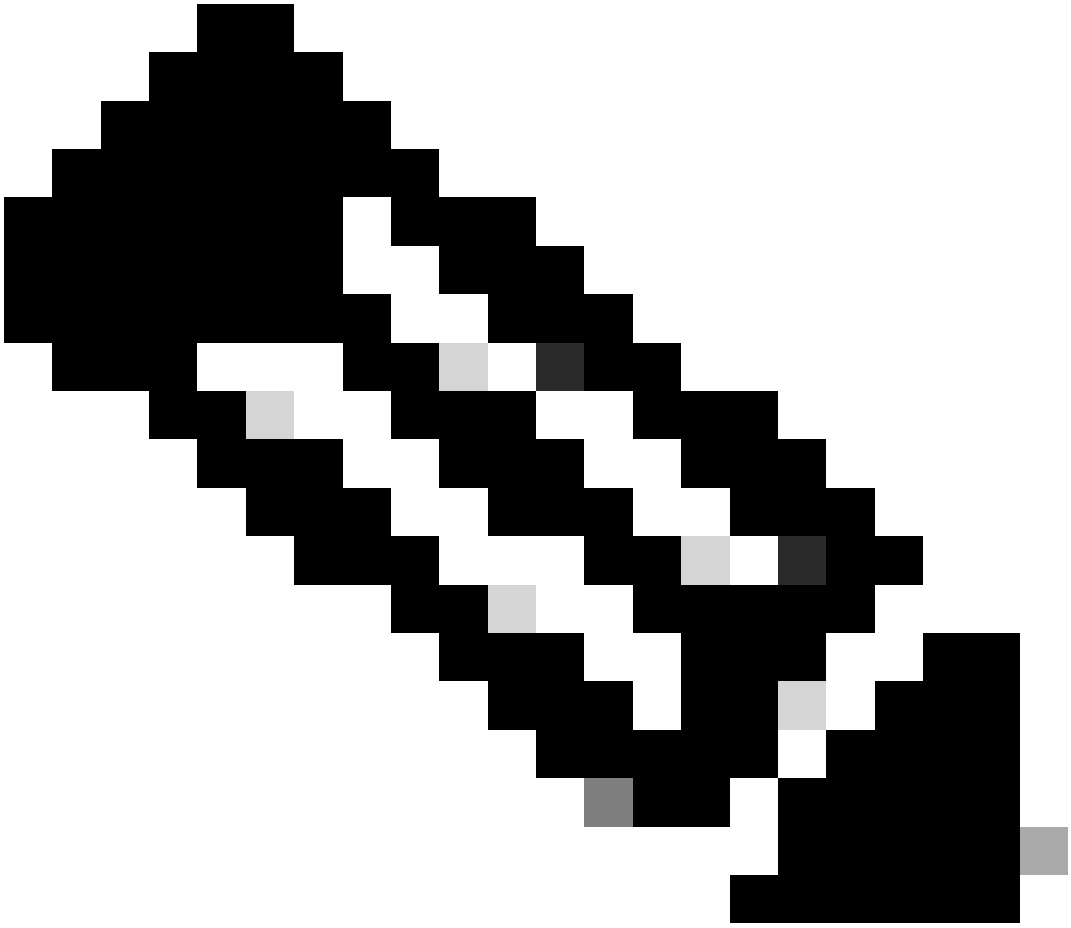
<#root>

neighbor peer-group-name peer-group

Cet exemple applique les groupes d'homologues aux voisins BGP internes et externes :

```
RTC#
router bgp 300
 neighbor internalmap peer-group
 neighbor internalmap remote-as 300
 neighbor internalmap route-map SETMETRIC out
 neighbor internalmap filter-list 1 out
 neighbor internalmap filter-list 2 in
 neighbor 10.5.5.2 peer-group internalmap
 neighbor 10.6.6.2 peer-group internalmap
 neighbor 10.3.3.2 peer-group internalmap
 neighbor 10.3.3.2 filter-list 3 in
```

Cette configuration définit un groupe d'homologues avec le nom internalmap. La configuration définit quelques stratégies pour le groupe, comme une mise en correspondance de route SETMETRIC pour définir la métrique à 5 et deux listes de filtres différentes, 1 et 2. La configuration applique le groupe d'homologues à tous les voisins internes RTE, RTF et RTG. En outre, la configuration définit une liste de filtres 3 distincte pour le voisin RTE. Cette liste de filtres remplace la liste de filtres 2 à l'intérieur du groupe d'homologues.

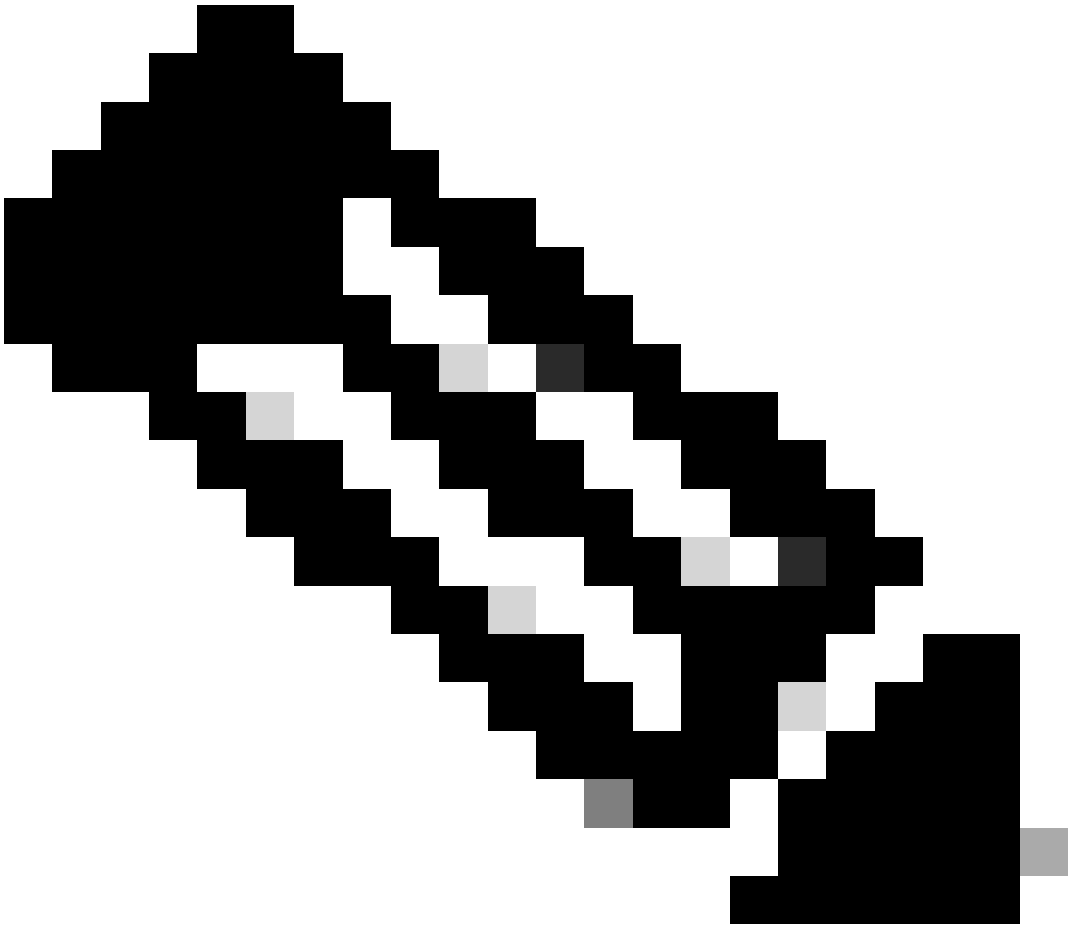


Remarque : Vous pouvez seulement remplacer les options qui concernent les mises à jour entrantes.

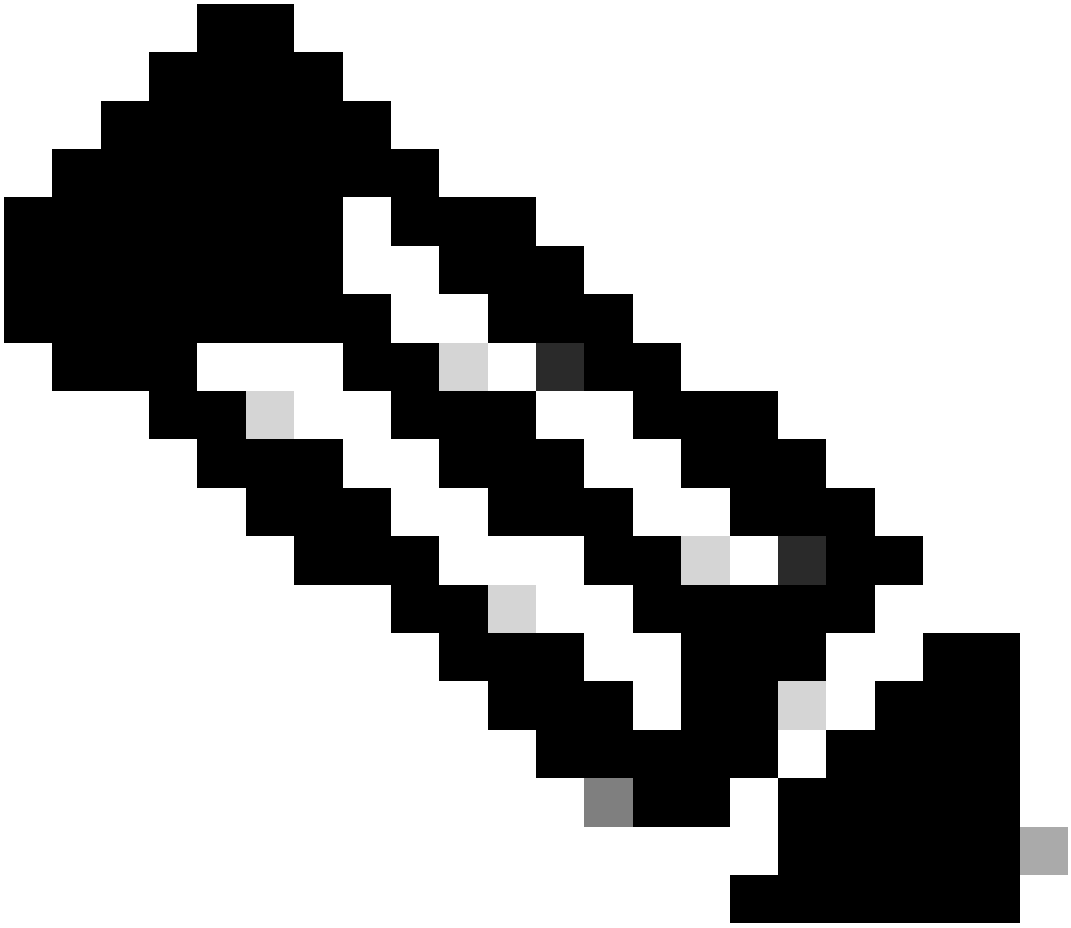
Maintenant, regardez comment vous pouvez utiliser des groupes d'homologues avec des voisins externes. Avec le même diagramme de cette section, vous configurez RTC avec un groupe d'homologues externalmap et appliquez le groupe d'homologues aux voisins externes.

```
RTC#
router bgp 300
 neighbor externalmap peer-group
 neighbor externalmap route-map SETMETRIC
 neighbor externalmap filter-list 1 out
 neighbor externalmap filter-list 2 in
 neighbor 10.2.2.2 remote-as 100
```

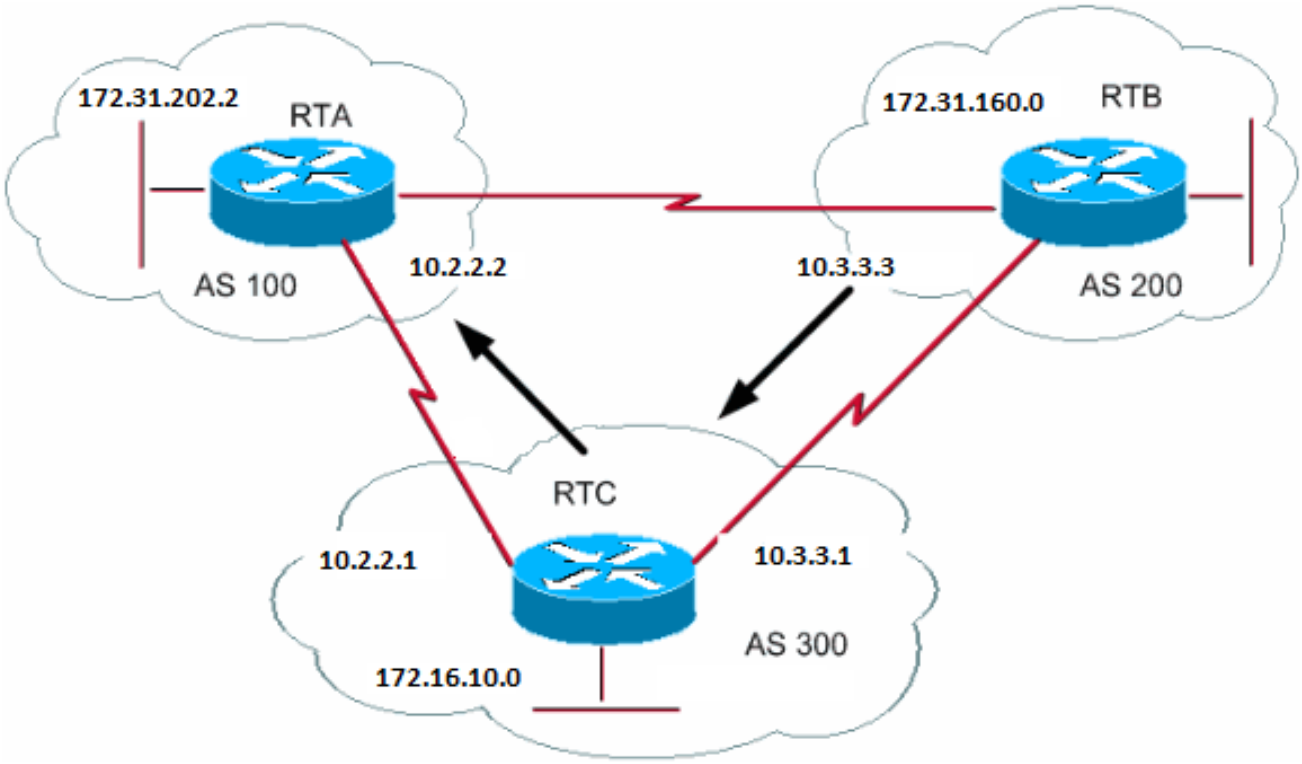
```
neighbor 10.2.2.2 peer-group externalmap
neighbor 10.4.4.2 remote-as 600
neighbor 10.4.4.2 peer-group externalmap
neighbor 10.1.1.2 remote-as 200
neighbor 10.1.1.2 peer-group externalmap
neighbor 10.1.1.2 filter-list 3 in
```



Remarque : Dans ces configurations, vous définissez les instructions « remote-as » à l’extérieur du groupe d’homologues, car vous devez définir différents AS externes. En outre, vous remplacez les mises à jour entrantes du voisin 10.1.1.2 avec l'attribution de la liste de filtres 3. Pour plus d'informations sur les groupes d'homologues, référez-vous à la section Groupes d'homologues BGP.



Remarque : Dans la version 12.0(24)S de Cisco IOS, Cisco a ajouté la fonctionnalité de mise à jour dynamique du BGP des groupes d’homologues. La fonctionnalité est aussi disponible dans les versions ultérieures du logiciel Cisco IOS. La fonctionnalité introduit un nouvel algorithme qui calcule dynamiquement et optimise les groupes de mise à jours de voisins qui partagent les mêmes stratégies sortantes. Ces voisins peuvent partager les mêmes messages de mise à jour. Dans les versions antérieures du logiciel Cisco IOS, le groupe des messages de mise à jour BGP était basé sur les configurations des groupes d'homologues. Cette méthode consistant à grouper les mises à jour a limité les stratégies sortantes et les configurations de sessions spécifiques. La fonctionnalité de groupe d'homologues de mise à jour dynamique BGP sépare la réplication du groupe de mises à jour de la configuration du groupe d'homologues. Cette séparation améliore le temps de convergence et la flexibilité de la configuration du voisin. Référez-vous à la section Groupes d'homologues de mise à jour dynamiques BGP pour plus de détails.



L'une des principales améliorations de BGP4 par rapport à BGP3 est le routage interdomaine sans classe (CIDR). CIDR ou les super-réseaux sont une nouvelle façon de considérer des adresses IP. Avec CIDR, il n'y a aucune notion de classes, comme la classe A, B ou C. Par exemple, le réseau 192.168.213.0 était auparavant un réseau de classe C non autorisé. Maintenant, le réseau est un super-réseau légal, 192.168.213.0/16. Le 16 représente le nombre de bits dans le filtre d'adresse locale, lorsque vous comptez à partir de l'extrémité gauche de l'adresse IP. Cette représentation est semblable à 192.168.213.0 255.255.0.0.

Vous employez des agrégats afin de réduire au minimum la taille du routage des tables. L'agrégation est le processus qui combine les caractéristiques de plusieurs routes différentes de telle manière que l'annonce d'une seule route soit possible. Dans cet exemple RTB génère le réseau 172.31.160.0. Vous configurez RTC pour propager un super-réseau de cette route 192.168.160.0 à RTA :

```
RTB#
router bgp 200
 neighbor 10.3.3.1 remote-as 300
 network 172.31.160.0

#RTC
router bgp 300
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.2.2.2 remote-as 100
 network 172.16.10.0
 aggregate-address 192.168.160.0 255.0.0.0
```

RTC propage l'adresse agrégée 192.168.160.0 à RTA.

Commandes d'agrégat

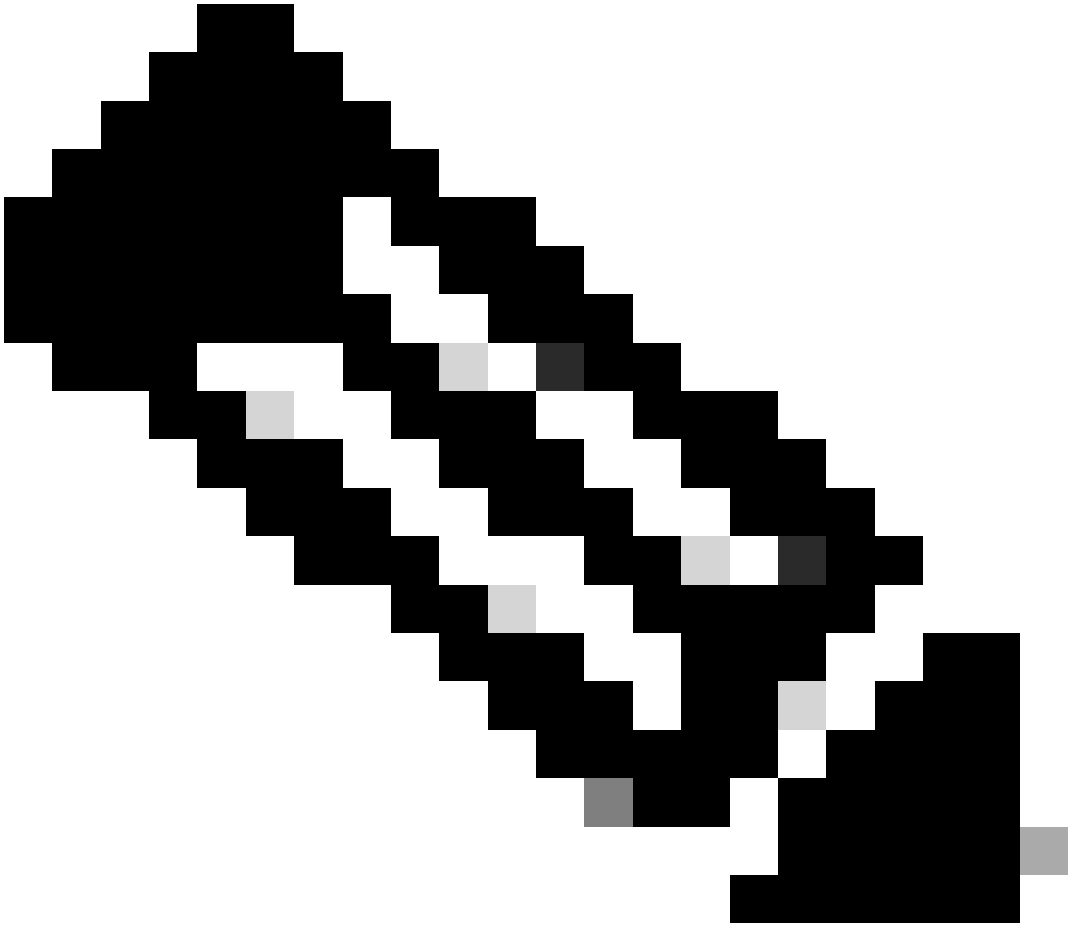
Il y a un large éventail de commandes d'agrégat. Vous devez comprendre comment chacune fonctionne afin d'obtenir le comportement d'agrégation que vous désirez.

La première commande est celle de l’exemple figurant dans la section CIDR and Aggregate Addresses (adresses CIDR et d’agrégation) :

<#root>

aggregate-address address-mask

Cette commande annonce la route du préfixe et toutes les routes plus spécifiques. La commande **aggregate-address 192.168.160.0** propage un réseau supplémentaire 192.168.160.0, sans toutefois empêcher la propagation de 172.31.160.0 au RTA. Les résultats sont la propagation des réseaux 192.168.160.0 et 172.31.160.0 à RTA, qui est l'annonce de la route du prefix et de la route plus spécifique.



Remarque : Vous ne pouvez pas agréger une adresse si vous n’avez pas de route plus précise pour l’adresse en question dans le tableau de routage du BGP.

Par exemple, le RTB ne peut pas générer d’agrégation pour le réseau 192.168.160.0 si le RTB n’a pas d’entrée plus précise pour 192.168.160.0 dans le tableau du BGP. Une injection de la route plus spécifique dans la table BGP est possible. L'injection de la route peut se faire par l'intermédiaire de :

- mises à jour entrantes depuis un autre AS ;

- redistribution d'un IGP ou de statiques dans BGP ;

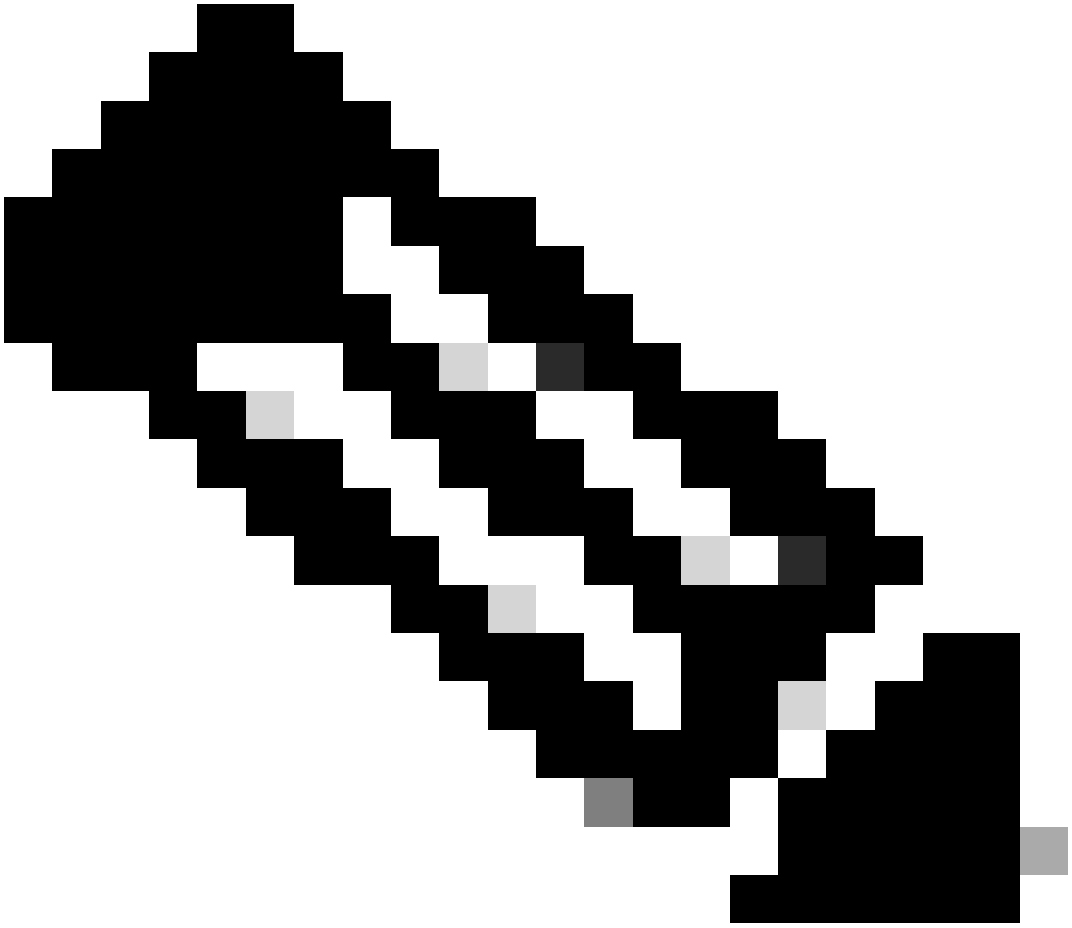
- la commande network , par exemple, network 172.31.160.0.

Si vous souhaitez que le RTC propage seulement le réseau 192.168.160.0 et **non** la route précise, exécutez la commande suivante :

```
<#root>

aggregate-address <address> <mask> summary-only
```

Cette commande annonce seulement le préfixe. La commande supprime toutes les routes plus spécifiques.

La commande **aggregate 192.168.160.0 255.0.0.0 summary-only** propage le réseau 192.168.160.0 et supprime la route plus précise de 172.31.160.0.



Remarque : Si vous agrégez un réseau qui a effectué l’injection dans votre BGP par l’intermédiaire de l’énoncé du réseau, l’entrée du réseau injecte toujours les mises à jour du BGP. Cette injection se produit même si vous utilisez la commande aggregate summary-only . L'exemple dans la section Exemple CIDR 1 traite de cette situation.

<#root>

aggregate-address <address> <mask> as-set

Cette commande annonce le préfixe et les routes plus spécifiques. Mais la commande inclut les informations as-set dans les informations du chemin des mises à jour du routage.

<#root>

aggregate 192.168.0.0 255.0.0.0 as-set

La section « CIDR Example 2 (as-set) » (exemple 2 du CIDR [as-set]) traite de cette commande.

Si vous voulez supprimer les routes plus spécifiques quand vous faites l'agrégation, définissez une mise en correspondance de route et appliquez-la aux agrégats. L'action permet d'être sélectif au sujet de quelles routes plus spécifiques sont à supprimer.

<#root>

aggregate-address <address> <mask> suppress-map <map-name>

Cette commande annonce le préfixe et les routes plus spécifiques. Mais la commande supprime l'annonce basée sur une mise en correspondance de route. Supposez que, avec le diagramme dans la section CIDR et adresses agrégées, vous voulez agréger 192.168.160.0, supprimer la route 192.168.160.20 plus spécifique et permettre la propagation de 172.31.160.0. Utilisez cette mise en correspondance de route :

```
route-map CHECK permit 10
 match ip address 1

access-list 1 permit 192.168.160.20 0.0.255.255
access-list 1 deny 0.0.0.0 255.255.255.255
```

Par définition de suppress-map, il y a une suppression à partir des mises à jour de tous les paquets que la liste d'accès autorise.

Appliquez ensuite la feuille de route à l'instruction aggregate.

```
RTC#
router bgp 300
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 remote-as 100
 network 172.16.10.0
 aggregate-address 192.168.160.0 255.0.0.0 suppress-map CHECK
```

Voici une autre variante :

<#root>

aggregate-address <address> <mask> attribute-map <map-name>

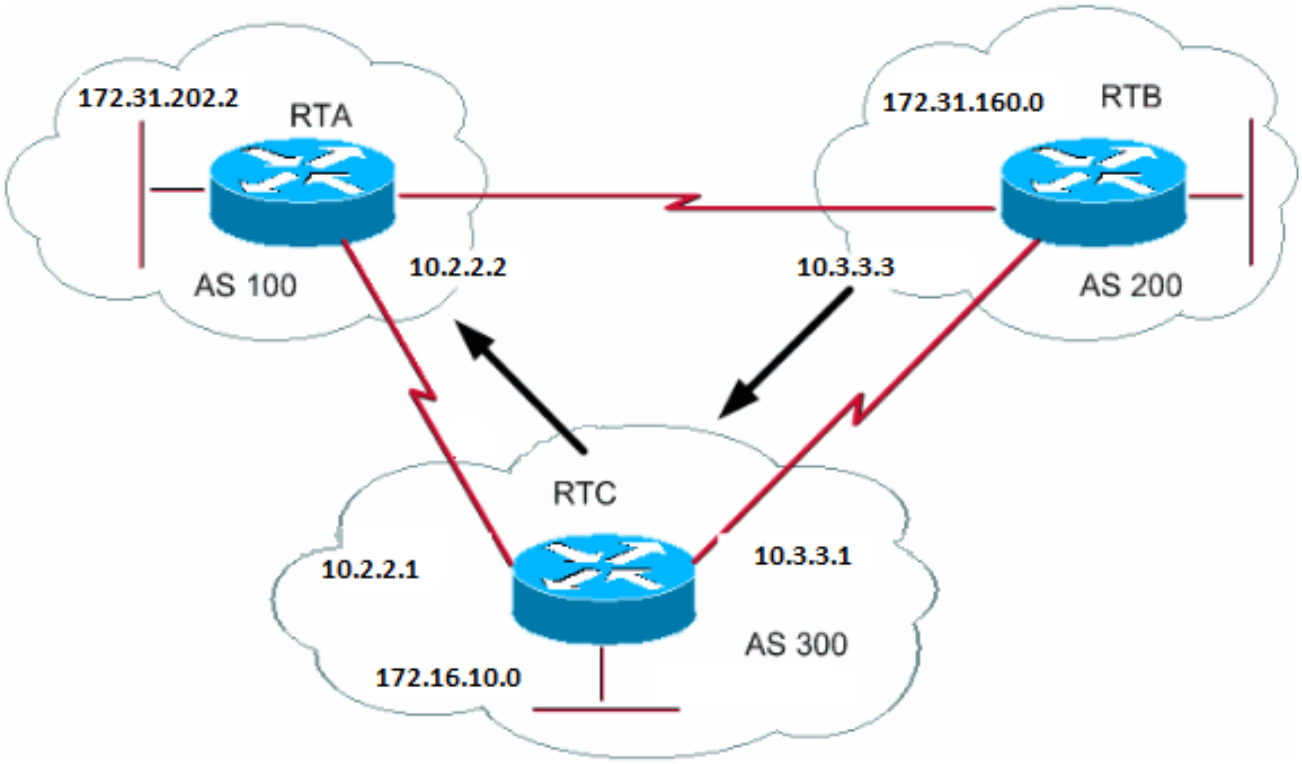
Cette commande vous permet de définir les attributs tels que metric, au moment de l'envoi des agrégats. Pour définir l'origine des agrégats sur IGP, appliquez cette mise en correspondance de route à la commande aggregate attribute-map :

```
route-map SETMETRIC
 set origin igp
```

```
aggregate-address 192.168.160.0 255.0.0.0 attribute-map SETORIGIN
```

Pour en savoir plus, consultez Understand Route Aggregation in BGP (comprendre l'agrégation des routes dans le BGP).

Exemple CIDR 1



Demande : Autorisez le RTB à annoncer le préfixe 192.168.160.0 et supprimez toutes les routes plus précises. Le problème avec cette demande, c'est que le réseau 172.31.160.0 est local à l'AS200, et cela signifie que ce dernier est à l'origine de 172.31.160.0. Vous ne pouvez pas obtenir que RTB génère un préfixe pour 192.168.160.0 sans génération d'une entrée pour 172.31.160.0, même si vous utilisez la commande aggregate summary-only . RTB produit les deux réseaux parce que RTB est le créateur de 172.31.160.0. Il y a deux solutions à ce problème.

La première solution est d'utiliser une route statique et de redistribuer dans BGP. Les résultats sont que RTB annonce l'agrégat avec une origine inachevée (?).

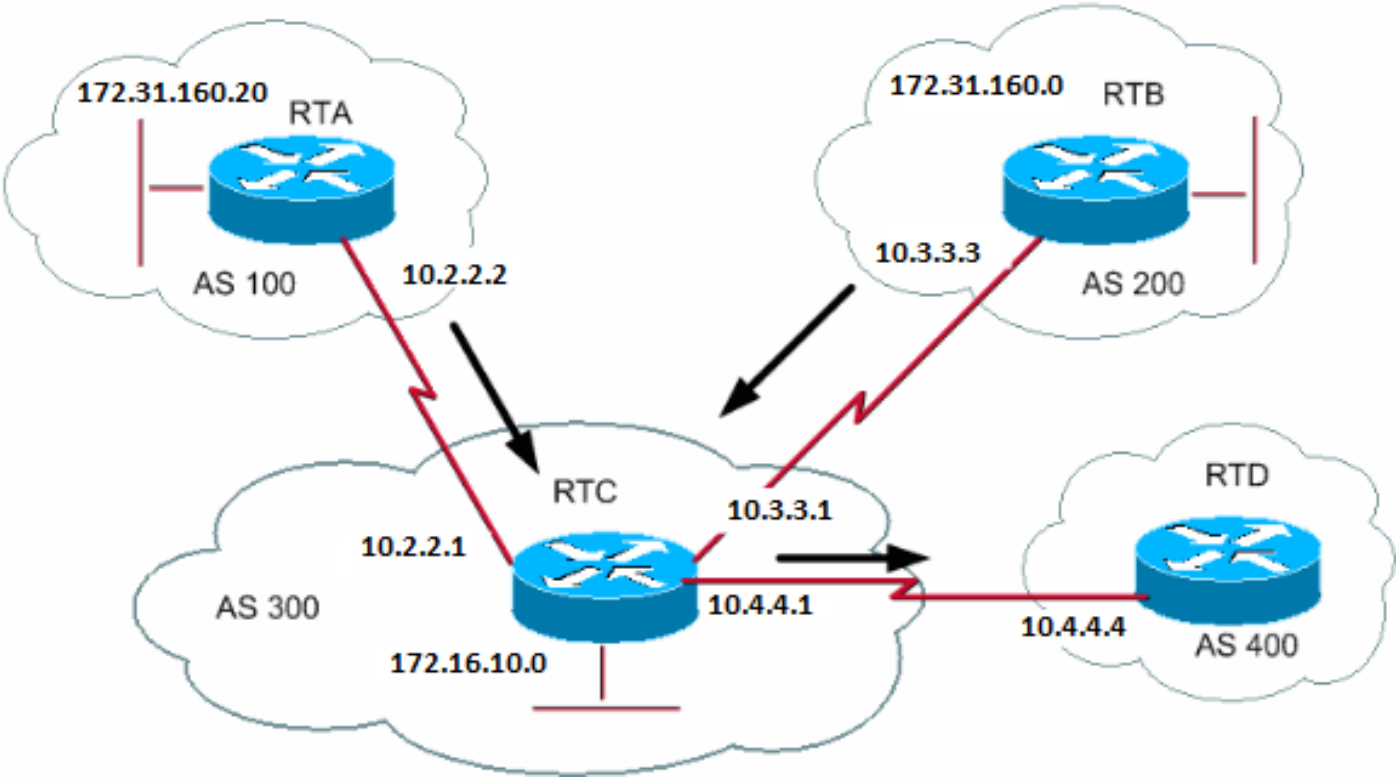
```
RTB#
router bgp 200
 neighbor 10.3.3.1 remote-as 300
 redistribute static

!--- This generates an update for 192.168.160.0 !--- with the origin path as "incomplete".

ip route 192.168.160.0 255.0.0.0 null0
```

La seconde solution consiste à ajouter, en plus de la route statique, une entrée pour la commande network . Cette entrée a le même effet, sauf que l'entrée définit l'origine de la mise à jour à IGP.

```
RTB#
router bgp 200
 network 192.168.160.0 mask 255.0.0.0

!--- This entry marks the update with origin IGP.

 neighbor 10.3.3.1 remote-as 300
 redistribute static

ip route 192.168.160.0 255.0.0.0 null0
```

Exemple CIDR 2 (as-set)

Vous employez l'instruction as-set dans l'agrégation pour réduire la taille des informations du chemin. Avec as-set, le numéro AS est listé une seule fois, indépendamment du nombre de fois qu'il apparaît dans les chemins qui ont été agrégés. Vous utilisez la commande aggregate as-set dans les situations dans lesquelles l'agrégation d'informations entraîne la perte d'informations en ce qui concerne l'attribut du chemin. Dans cet exemple, RTC obtient des mises à jour sur 192.168.160.20 de RTA et des mises à jour sur 172.31.160.0 de RTB. Supposez que RTC veuille agréger le réseau 192.168.160.0/8 et envoie le réseau à RTD. RTD ne connaît pas l'origine de cette route. Si vous ajoutez l'instruction aggregate as-set, vous forcez RTC à générer les informations de chemin sous la forme d'un ensemble {}. Cet ensemble inclut toutes les informations de chemin, indépendamment du chemin qui est arrivé en premier.




```
RTB#
```

```
router bgp 200
 network 172.31.160.0
 neighbor 10.3.3.1 remote-as 300

RTA#
router bgp 100
 network 192.168.160.20
 neighbor 10.2.2.1 remote-as 300
```

Cas 1 :

RTC n'a pas une instruction as-set. RTC envoie une mise à jour 192.168.160.0/8 à RTD avec les informations du chemin (300), comme si la route provenait d'AS300.

```
RTC#
router bgp 300
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.4.4.4 remote-as 400
 aggregate 192.168.160.0 255.0.0.0 summary-only

!--- This command causes RTC to send RTD updates about 192.168.160.0/8
!--- with no indication that 192.168.160.0 actually comes from two different ASs.
!--- This may create loops if RTD has an entry back into AS100 or AS200.
```

Cas 2 :

```
RTC#
router bgp 300
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.4.4.4 remote-as 400
 aggregate 192.168.160.0 255.0.0.0 summary-only
 aggregate 192.168.160.0 255.0.0.0 as-set

!--- This command causes RTC to send RTD updates about 192.168.160.0/8
!--- with an indication that 192.168.160.0 belongs to a set {100 200}.
```

Les deux sujets suivants, la confédération BGP et les réflecteurs de route, s’adressent aux fournisseurs Internet qui veulent mieux contrôler l’expansion de l’homologation iBGP dans leurs AS.

Confédération BGP

La mise en place de la confédération BGP réduit le maillage iBGP à l'intérieur d'un AS. L'astuce consiste à diviser un AS en plusieurs AS et à assigner tout le groupe à une seule confédération. Chaque AS individuel maille entièrement iBGP et a des connexions aux autres AS à l'intérieur de la confédération. Même si ces AS ont des homologues eBGP à l'AS dans la confédération, les AS échangent le routage comme s'ils utilisaient iBGP. De cette façon, la confédération préserve les prochaines informations de saut, de métrique et de préférences locales. Pour le monde extérieur, la confédération apparaît comme un AS unique.

Afin de configurer une confédération BGP, exécutez cette commande :

```
<#root>
```

bgp confederation identifier <autonomous-system>

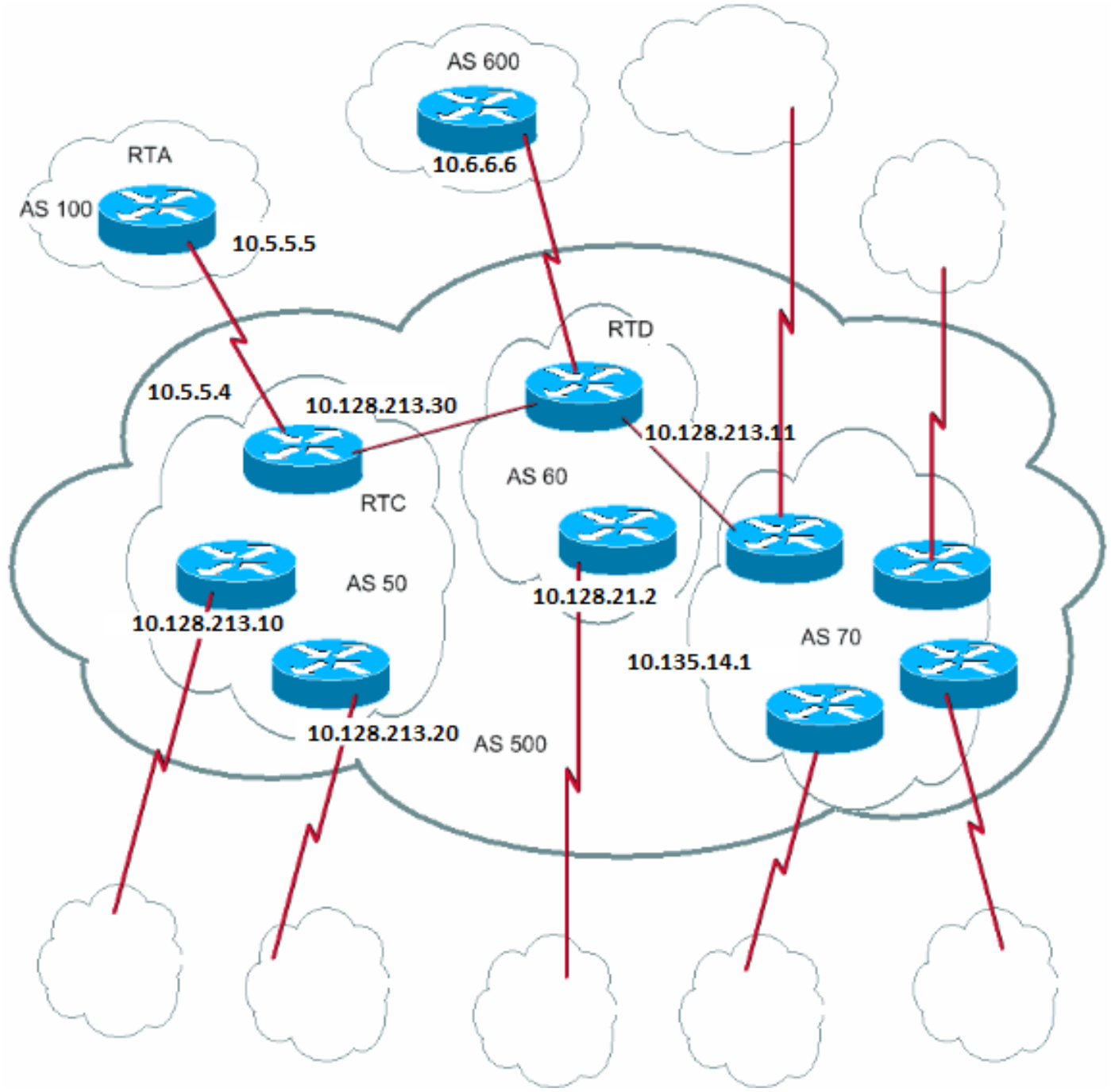
L'identificateur de la confédération est le numéro AS du groupe de la confédération.

L'exécution de cette commande exécute un appairage entre plusieurs AS dans la confédération :

```
<#root>
```

bgp confederation peers <autonomous-system> <autonomous-system>

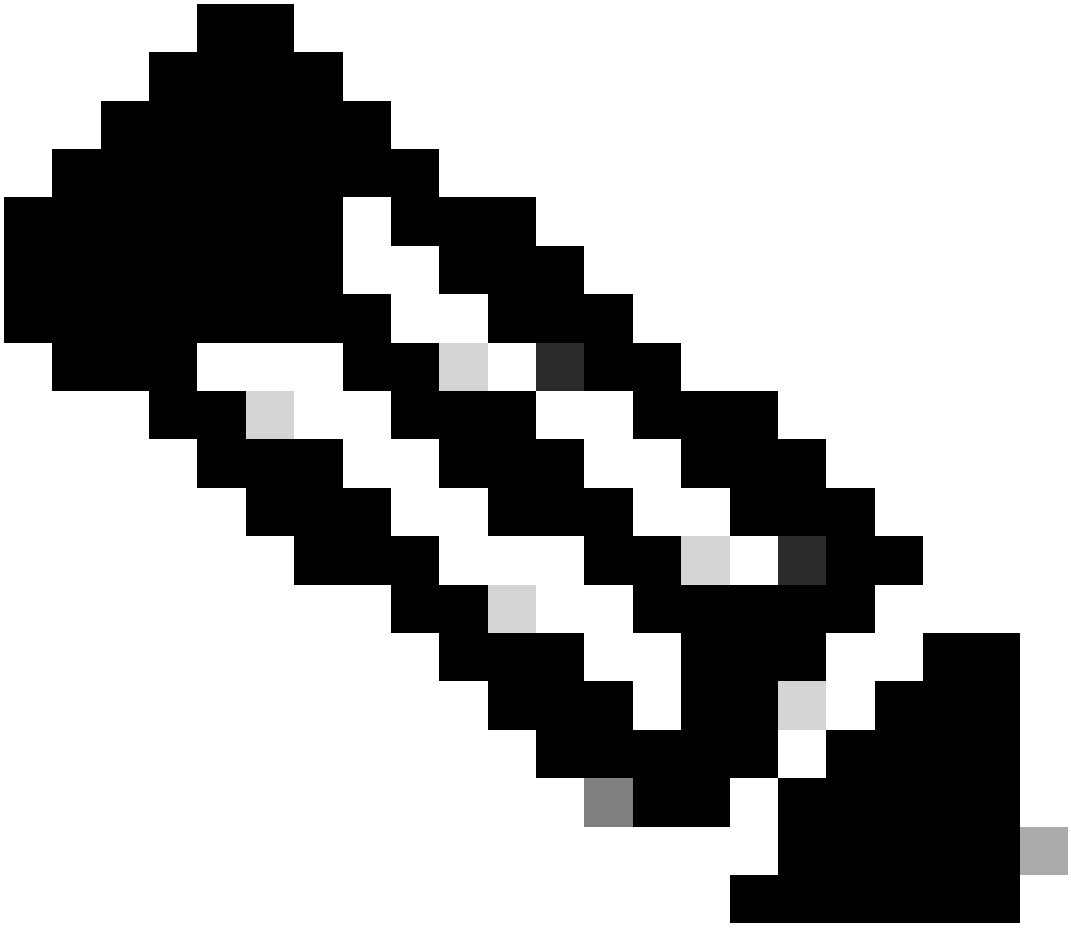
Voici un exemple de confédération :



Supposez que vous avez un AS500 qui se compose de neuf speakers BGP. D'autres speakers non-BGP existent également, mais vous êtes seulement intéressé par les speakers BGP qui ont des connexions eBGP aux autres AS. Si vous voulez exécuter un maillage iBGP complet à l'intérieur d'AS500, vous avez besoin de neuf connexions homologues pour chaque routeur. Vous avez besoin de huit homologues iBGP et d'un homologue eBGP aux AS externes.

Si vous utilisez la confédération, vous pouvez diviser AS500 en plusieurs AS : AS50, AS60 et AS70. Vous donnez l'AS comme identificateur de confédération de 500. Le monde extérieur voit seulement un AS, AS500. Pour chaque AS50, AS60 et AS70, vous définissez un maillage complet des homologues iBGP et vous définissez les listes des homologues de la confédération avec la commande bgp confederation peers.

Voici un exemple de configuration des routeurs RTC, RTD et RTA :



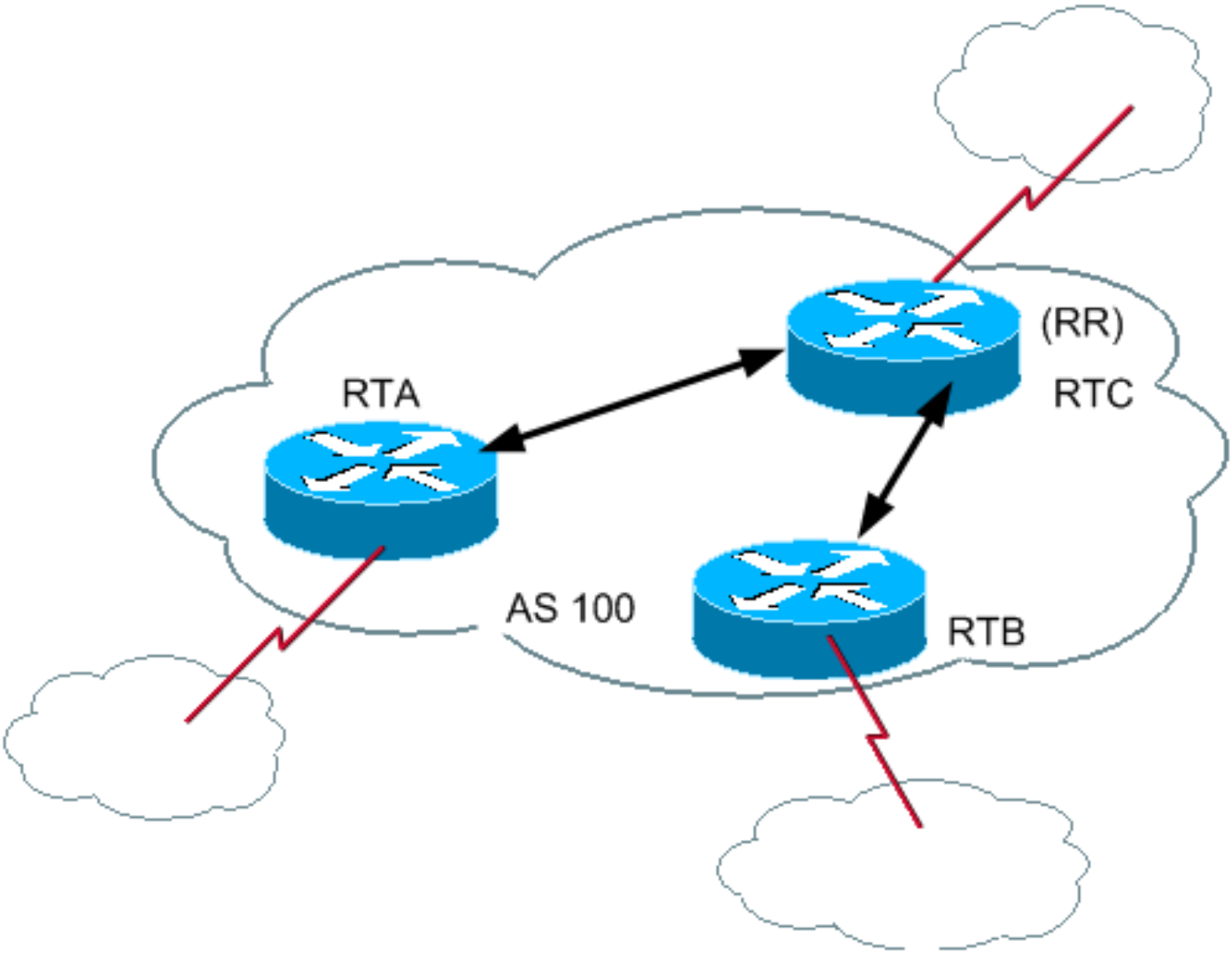
Remarque : Le RTA n’est pas au courant de l’existence des systèmes AS50, AS60 ou AS70. RTA connaît seulement AS500.

```
RTC#
router bgp 50
 bgp confederation identifier 500
 bgp confederation peers 60 70
 neighbor 10.128.213.10 remote-as 50 (IBGP connection within AS50)
 neighbor 10.128.213.20 remote-as 50 (IBGP connection within AS50)
 neighbor 10.128.213.11 remote-as 60 (BGP connection with confederation peer 60)
 neighbor 10.128.213.14 remote-as 70 (BGP connection with confederation peer 70)
 neighbor 10.5.5.5 remote-as 100 (EBGP connection to external AS100)

RTD#
router bgp 60
```

```
 bgp confederation identifier 500
 bgp confederation peers 50 70
 neighbor 10.128.210.2 remote-as 60 (IBGP connection within AS60)
 neighbor 10.128.213.30 remote-as 50(BGP connection with confederation peer 50)
 neighbor 10.128.213.14 remote-as 70 (BGP connection with confederation peer 70)
 neighbor 10.6.6.16 remote-as 600 (EBGP connection to external AS600)

RTA#
router bgp 100
 neighbor 10.5.5.4 remote-as 500 (EBGP connection to confederation 500)
```

Réflecteurs de route

Une autre solution pour l'explosion de l'appairage iBGP dans un AS est d'utiliser des réflecteurs de route (RR). Comme le montre la section iBGP, un haut-parleur BGP n’annonce pas une route que le haut-parleur BGP a détectée par l’intermédiaire d’un autre haut-parleur iBGP vers un troisième haut-parleur iBGP. Vous pouvez assouplir un peu cette restriction et fournir un contrôle supplémentaire, qui permet à un routeur d'annoncer, ou de refléter, des routes acquises par iBGP à d'autres speakers iBGP. Cette réflexion de route réduit le nombre d'homologues iBGP dans un AS.



Dans des cas normaux, maintenez un maillage iBGP complet entre RTA, RTB et RTC dans AS100. Si vous utilisez le concept RR, RTC peut être choisi en tant que RR. De cette façon, RTC a un iBGP partiel pour l'appairage avec RTA et RTB. L'appairage entre RTA et RTB n'est pas

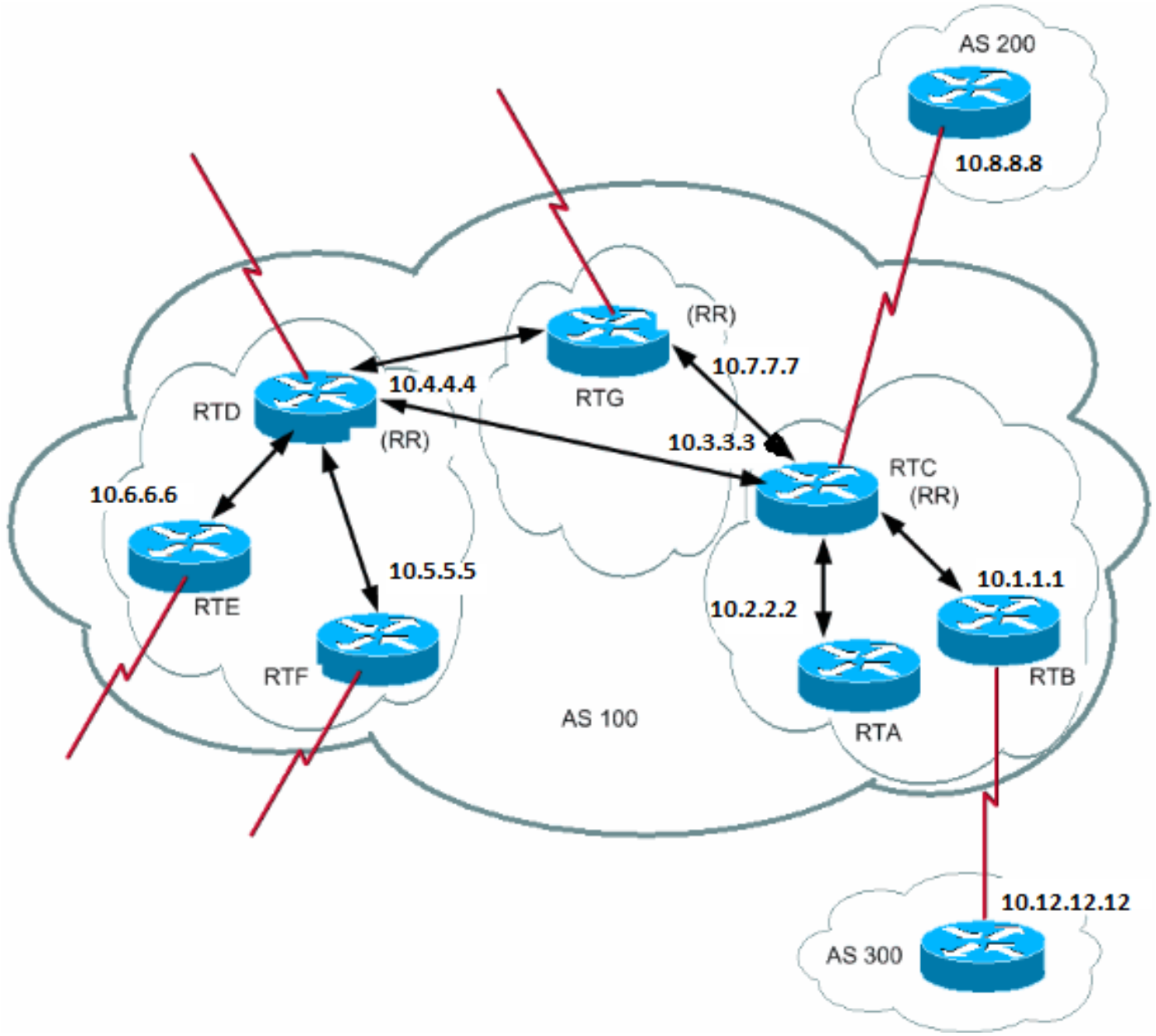
nécessaire parce que RTC est un RR pour les mises à jour qui viennent de RTA et de RTB.

<#root>

neighbor <ip address> route-reflector-client

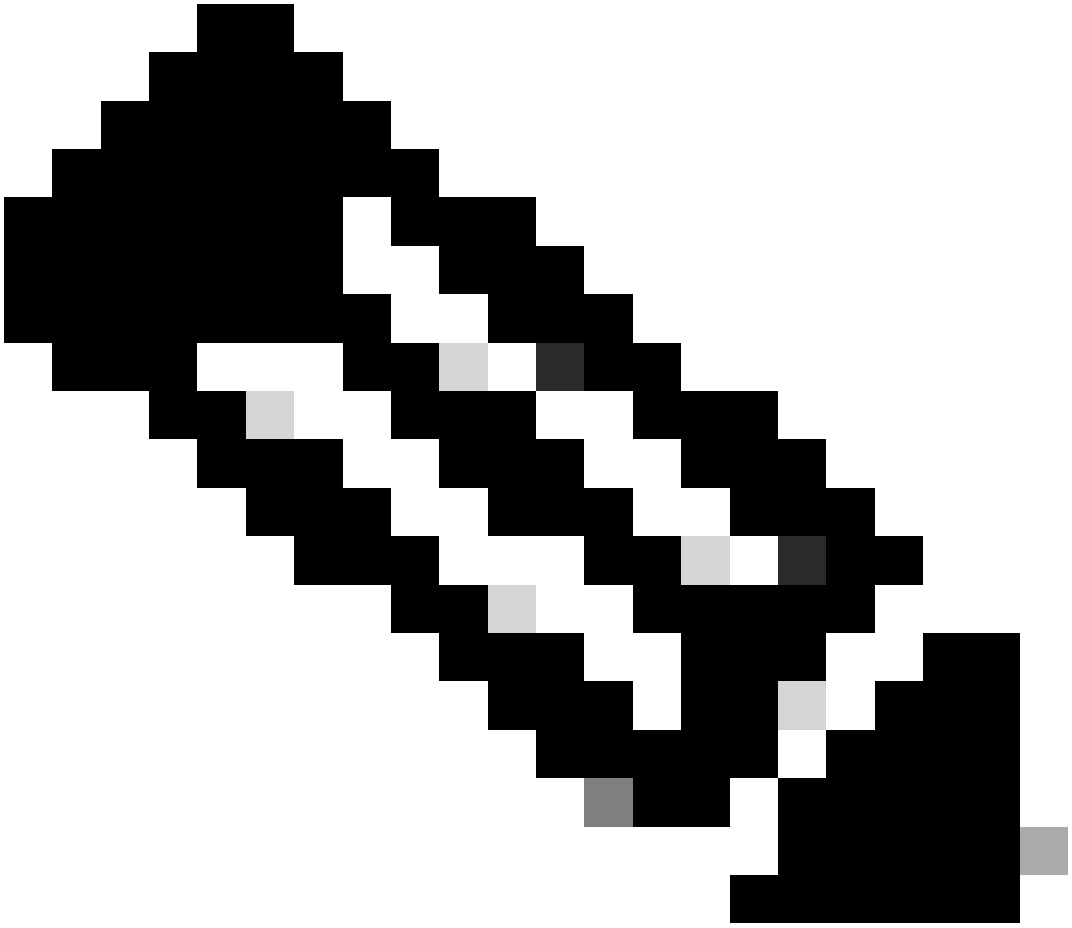
Le routeur avec cette commande est le RR, et les voisins auxquels la commande pointe sont les clients de cet RR. Dans l'exemple, la configuration RTC a la commande neighbor route-reflector-client qui pointe sur les adresses IP de RTA et de RTB. La combinaison du RR et des clients est un « cluster ». Dans cet exemple, RTA, RTB et RTC forment un cluster avec un RR unique dans AS100.

Les autres homologues iBGP du RR qui ne sont pas des clients sont des non-clients.



Un AS peut avoir plus d'un RR. Dans cette situation, un RR traite les autres RR comme n'importe quel autre speaker iBGP. Les autres RR peuvent appartenir au même cluster (groupe client) ou à d'autres clusters. Dans une configuration simple, vous pouvez diviser l'AS en plusieurs clusters. Vous configurez chaque RR avec d'autres RR comme homologues nonclients dans une topologie entièrement maillée. Les clients ne doivent pas se jumeler avec des haut-parleurs iBGP à l’extérieur de la grappe de clients.

Dans le schéma précédent, les RTA, RTB et RTC forment une seule grappe. RTC est le RR. Pour le RTC, RTA et RTB sont des clients et tout le reste est un nonclient. Rappelez-vous que la commande neighbor route-reflector-client pointe aux clients d'un RR. Le même RTD est le RR pour les RTE et RTF clients. RTG est un RR dans un cluster tiers.



Remarque : Les routeurs RTD, RTC et RTG sont entièrement maillés, mais pas les routeurs à l’intérieur d’une grappe.

Quand un RR reçoit une route, le RR route comme le montre la liste. Cependant, cette activité dépend du type d'homologue :

- Routage d'un homologue nonclient - Se reflète sur tous les clients dans le cluster.
- Routage d'un homologue client - Se reflète sur tous les homologues nonclients et également sur les homologues clients.

- Routage d'un homologue eBGP - Envoie une mise à jour à tous les homologues clients et nonclients.

Voici la configuration BGP des routeurs RTC, RTD et RTB :

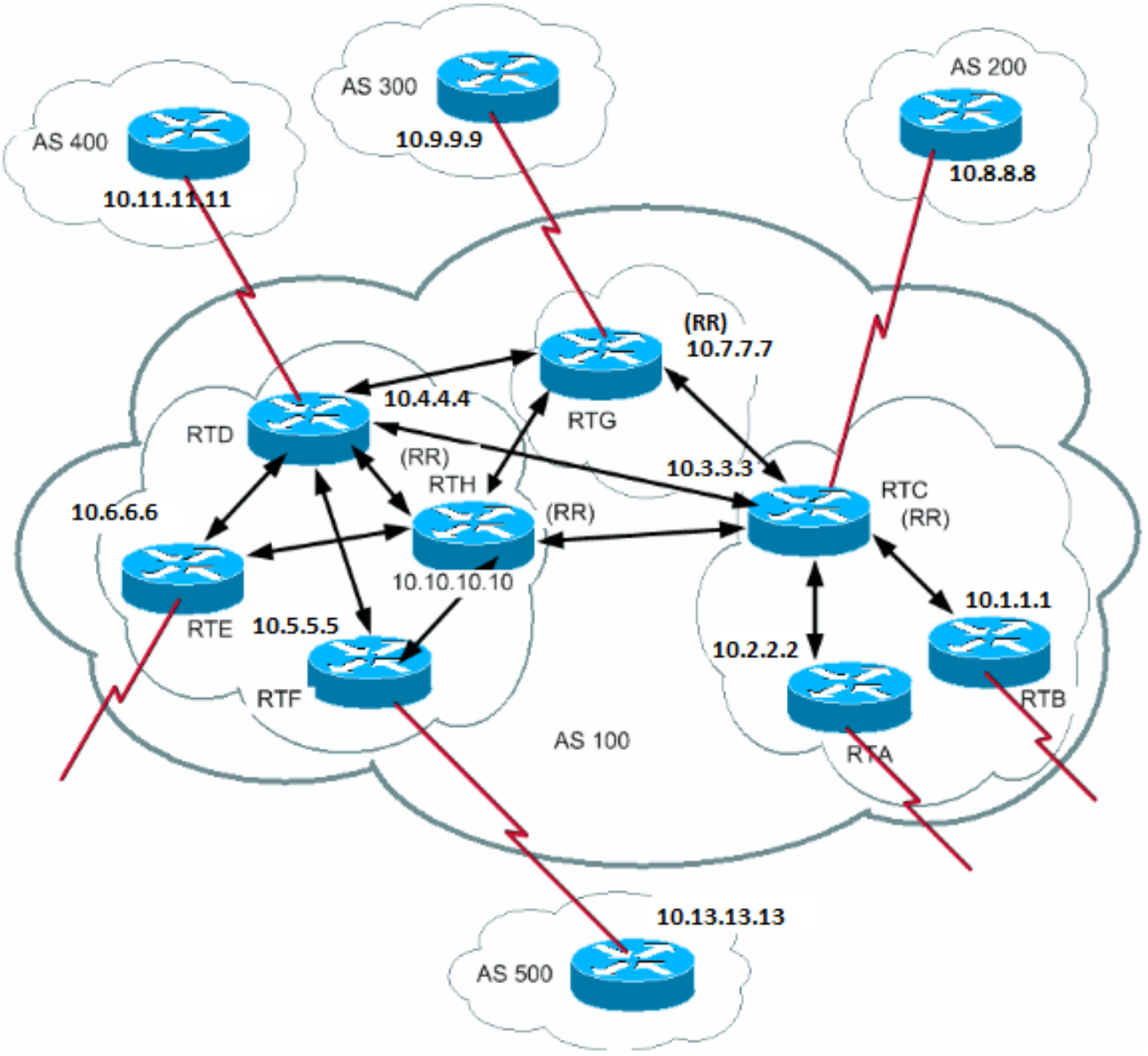
```
RTC#
router bgp 100
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 route-reflector-client
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.7.7.7 remote-as 100
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.8.8.8 remote-as 200

RTB#
router bgp 100
 neighbor 10.3.3.3 remote-as 100
 neighbor 10.12.12.12 remote-as 300

RTD#
router bgp 100
 neighbor 10.6.6.16 remote-as 100
 neighbor 10.6.6.16 route-reflector-client
 neighbor 10.5.5.5 remote-as 100
 neighbor 10.5.5.5 route-reflector-client
 neighbor 10.7.7.7 remote-as 100
 neighbor 10.3.3.3 remote-as 100
```

Puisqu'il y a une réflexion des routes acquises iBGP, il peut y avoir une boucle d'informations de routage. Le schéma RR inclut quelques méthodes pour éviter cette boucle :

- **originator-id - C'est un attribut BGP facultatif et non transitif qui est long de 4 octets.** Un RR crée cet attribut. L'attribut porte l'ID du router (RID) du créateur de la route dans l'AS local. Si, en raison d'une mauvaise configuration, les informations de routage reviennent au créateur, l'information est ignorée.

- **cluster-list** – La section « Multiple RR within a Cluster » (les divers RR d’une grappe) couvre la liste des grappes.

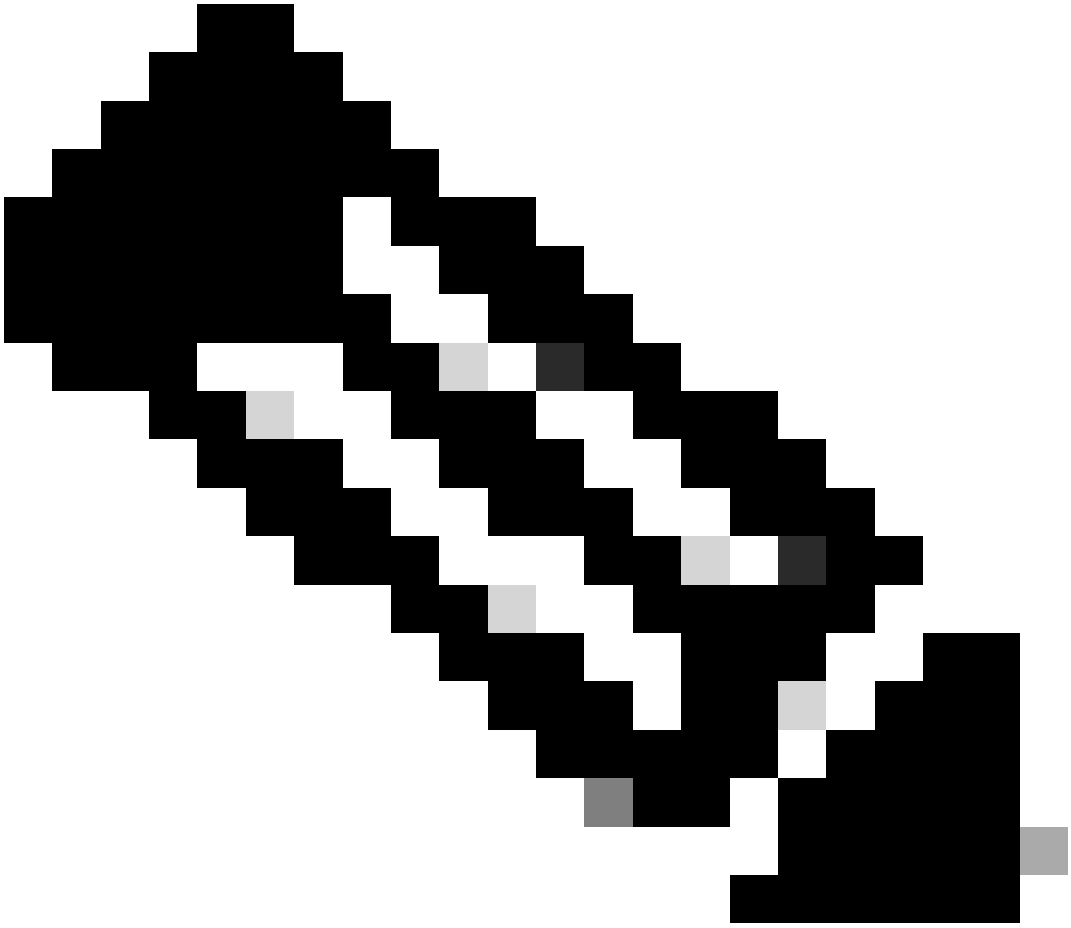
Plusieurs RR dans un cluster



Habituellement, un cluster de clients a un RR unique. Dans ce cas, l'ID du router du RR identifie le cluster. Afin d'augmenter la redondance et d'éviter des points de panne uniques, un cluster peut avoir plus d'un RR. Vous devez configurer tous les RR dans le même cluster avec un ID de cluster de 4 octets de sorte qu'un RR puisse identifier les mises à jour de RR dans le même cluster.

Une liste de clusters est une séquence d'ID de clusters que la route a passés. Lorsqu'un RR reflète une route depuis des clients RR vers des nonclients hors du cluster, RR ajoute l'ID du cluster local à la liste des clusters. Si cette mise à jour a une liste de clusters vide, le RR en crée une. Avec cet attribut, un RR peut déterminer si les informations de routage sont revenues au même cluster en raison d'une mauvaise configuration. Si l'ID du cluster local est trouvé dans la liste des clusters, l'annonce est ignorée.

Dans le diagramme de cette section, RTD, RTE, RTF et RTH appartiennent à un seul cluster. RTD et RTH sont les RR pour le même cluster.

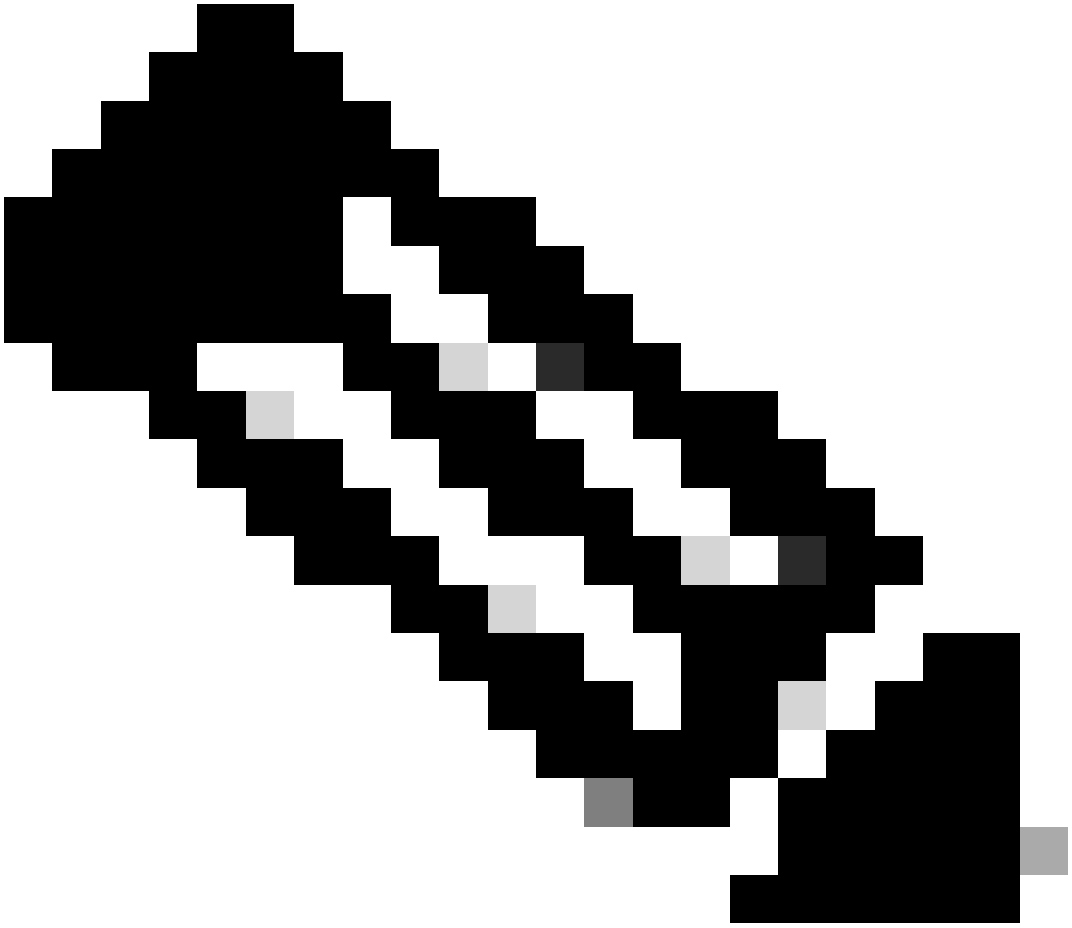


Remarque : Il y a redondance, car le RTH a un jumelage entièrement maillé avec tous les RR. Si RTD s'arrête, RTH remplace RTD.

Voici la configuration de RTH, RTD, RTF et RTC :

```
RTH#
router bgp 100
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.5.5.5 remote-as 100
 neighbor 10.5.5.5 route-reflector-client
 neighbor 10.6.6.16 remote-as 100
 neighbor 10.6.6.16 route-reflector-client
 neighbor 10.7.7.7 remote-as 100
 neighbor 10.3.3.3 remote-as 100
```

```
 neighbor 10.9.9.9 remote-as 300
 bgp cluster-id 10

RTD#
router bgp 100
 neighbor 10.10.10.10 remote-as 100
 neighbor 10.5.5.5 remote-as 100
 neighbor 10.5.5.5 route-reflector-client
 neighbor 10.6.6.16 remote-as 100
 neighbor 10.6.6.16 route-reflector-client
 neighbor 10.7.7.7 remote-as 100
 neighbor 10.3.3.3 remote-as 100
 neighbor 10.11.11.11 remote-as 400
 bgp cluster-id 10

RTF#
router bgp 100
 neighbor 10.10.10.10 remote-as 100
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.13.13.13 remote-as 500

RTC#
router bgp 100
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 route-reflector-client
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.7.7.7 remote-as 100
 neighbor 10.10.10.10 remote-as 100
 neighbor 10.8.8.8 remote-as 200
```



Remarque : Vous n’avez pas besoin de la commande « bgp cluster-id » pour le RTC, car il n’existe qu’un seul RR dans cette grappe.

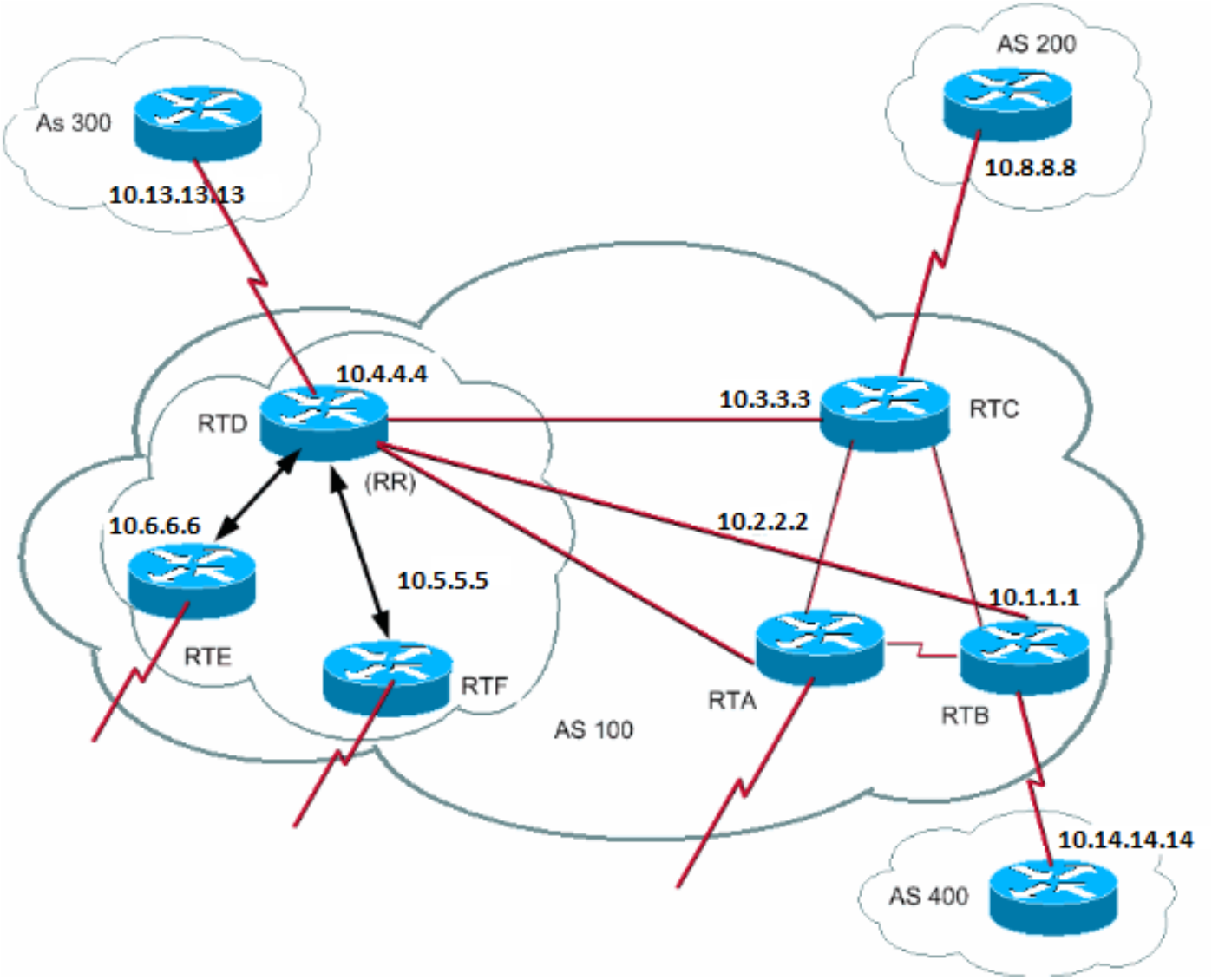
Attention : Cette configuration n’utilise pas de groupes d’homologues. N'utilisez pas les groupes d'homologues si les clients à l'intérieur d'un cluster n'ont pas des homologues iBPG directs réciproques et s'ils échangent des mises à jour via RR. Si vous configurez des groupes d'homologues, un retrait potentiel à la source d'une route sur le RR se transmet à tous les clients dans le cluster. Cette transmission peut poser des problèmes.

La sous-commande du routeur bgp client-to-client reflection **est activée par défaut sur le RR.** Si vous désactivez la réflection client-à-client BGP sur le RR et vous rendez l'appairage BGP redondant entre les clients, vous pouvez sans risque utiliser des groupes d'homologues. Consultez la section Limitations of Peer Groups (limites des groupes d’homologues) pour en savoir plus.

RR et speakers BGP conventionnels

Un AS peut avoir des speakers BGP qui ne comprennent pas le principe des RR. Ce document appelle ces routeurs des speakers BGP

conventionnels. Le schéma RR permet à de tels speakers BGP conventionnels de coexister. Ces routeurs peuvent être des membres d'un groupe de clients ou d'un groupe de nonclients. L'existence de ces routeurs permet la migration facile et progressive du modèle iBGP actuel au modèle RR. Vous pouvez commencer à créer des clusters si vous configurez un seul routeur en tant que RR et rendez les autres RR et clients RR des homologues iBGP normaux. Ensuite, vous pouvez graduellement créer plus de clusters.



Dans ce diagramme, RTD, RTE, et RTF appliquent le concept de réflexion de route. RTC, RTA et RTB sont des routeurs traditionnels. Vous ne pouvez pas configurer ces routeurs comme RR. Vous pouvez exécuter un maillage iBGP normal entre ces routeurs et RTD. Plus tard, quand vous serez prêt à la mise à niveau, vous pourrez transformer le RTC en RR avec des RTA et RTB clients. Les clients n'ont pas besoin de comprendre le schéma de réflexion de la route; seuls les RR nécessitent la mise à niveau.

Voici la configuration de RTD et RTC :

```
RTD#
router bgp 100
 neighbor 10.6.6.16 remote-as 100
 neighbor 10.6.6.16 route-reflector-client
 neighbor 10.5.5.5 remote-as 100
 neighbor 10.5.5.5 route-reflector-client
 neighbor 10.3.3.3 remote-as 100
 neighbor 10.2.2.2 remote-as 100
```

```
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.13.13.13 remote-as 300

RTC#
router bgp 100
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.14.14.14 remote-as 400
```

Quand vous êtes prêt à la mise à niveau d'RTC et à la transformation de RTC en RR, supprimez le maillage complet iBGP et faites de RTA et RTB des clients de RTC.

Éviter la boucle des informations de routage

Jusqu'à présent, ce document a mentionné deux attributs que vous pouvez utiliser pour éviter les boucles d'information potentielles : **originator-id** et **cluster-list**.

Un autre moyen de contrôler les boucles est d'utiliser plus de restrictions au niveau de la clause set **des mises en correspondance de route sortantes.** La clause set pour les mises en correspondance de route sortantes n'affecte pas les routes qui se reflètent aux homologues iBGP.

Vous pouvez également appliquer davantage de restrictions sur la commande **next-hop-self**, qui est une option de configuration par voisin. Lorsque vous utilisez **next-hop-self** sur les RR, la clause concerne seulement le prochain saut des routes détectées par l'eBGP, car le saut des routes détectées ne doit pas être modifié.

Atténuation de la déflexion de route

Le logiciel Cisco IOS version 11.0 a introduit le route dampening. Le route dampening est un mécanisme qui permet de réduire au minimum l'instabilité que l'oscillation de la route provoque. Le route dampening réduit également l'oscillation sur le réseau. Vous définissez des critères pour identifier les routes dont le comportement est défaillant. Une route qui oscille a une pénalité de 1000 pour chaque oscillation. Dès que la pénalité cumulative atteint une limite de suppression prédéfinie, l'annonce de la route est supprimée. La pénalité diminue de façon exponentielle en fonction d'un temps de demi-vie configuré au préalable. Lorsque la pénalité est inférieure à une limite de réutilisation prédéfinie, l'annonce de la route n'est plus supprimée.

Le route dampening ne s'applique pas aux routes qui sont externes à un AS et ont appris par l'intermédiaire d'iBGP. De cette façon, le route dampening évite une pénalité plus élevée pour les homologues iBGP des routes externes au AS.

La pénalité décline à une granularité de 5 secondes. Les routes ne sont pas supprimées à une granularité de 10 secondes. Le routeur conserve l'information d'atténuation jusqu'à ce que la pénalité soit inférieure à la moitié de la limite de réutilisation. À ce stade, le routeur purge l'information.

Au début, l'atténuation est désactivée par défaut. S'il y a lieu, cette fonctionnalité pourra être activée par défaut ultérieurement. Ces commandes contrôlent le route dampening :

- **bgp dampening - Active l'atténuation.**

- **no bgp dampening — Désactive l'atténuation.**

- **bgp dampeninghalf-life-time** – Modifie le temps de demi-vie.

Une commande qui définit tous les paramètres en même temps est :

- **bgp dampeninghalf-life-timereusesuppressmaximum-suppress-time**

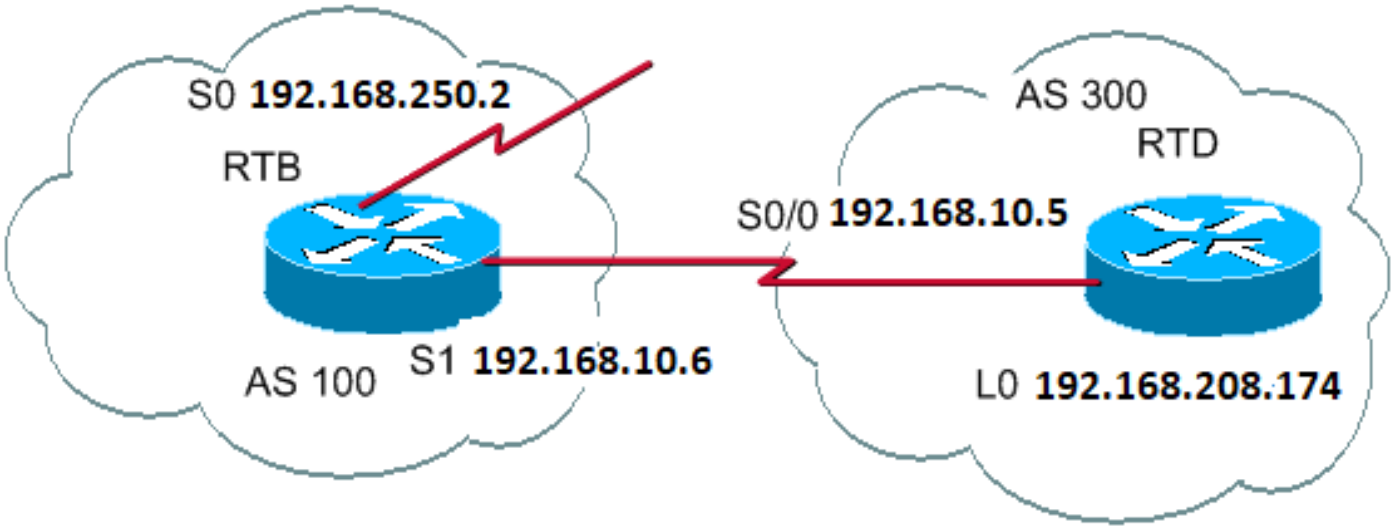
Cette liste détaille la syntaxe :

- **half-life-time** – La plage est de 1 à 45 minutes, et la valeur par défaut actuelle est de 15 minutes.

- **reuse-value** – La plage est de 1 à 20 000, et la valeur par défaut est de 750.

- **delete-value** – La plage est de 1 à 20 000 et la valeur par défaut est de 2 000.

- **max-suppress-time** – Il s’agit de la durée maximale pour la suppression d’une route. La plage va de 1 à 255 minutes, et la valeur par défaut est quatre fois la durée de demi-vie.




```
RTB#
hostname RTB

interface Serial0
 ip address 192.168.250.2 255.255.255.252

interface Serial1
 ip address 192.168.10.6 255.255.255.252

router bgp 100
 bgp dampening
 network 192.168.250.15
 neighbor 192.168.10.5 remote-as 300
```

```
RTD#
hostname RTD

interface Loopback0
 ip address 192.168.208.174 255.255.255.192

interface Serial0/0
 ip address 192.168.10.5 255.255.255.252

router bgp 300
 network 192.168.10.0
 neighbor 192.168.10.6 remote-as 100
```

La configuration de RTB sert au route dampening avec les paramètres par défaut. Si vous supposez que la liaison eBGP à RTD est stable, la table BGP RTB ressemble à ceci :

<#root>

```
RTB#
```

show ip bgp

```
BGP table version is 24, local router ID is 192.168.250.2 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin
codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 192.168.10.0     192.168.10.5           0             0 300 i
*> 192.168.250.15     0.0.0.0                0         32768 i
```

Pour simuler un affollement de la route, exécutez la commande clear ip bgp 192.168.10.6 sur RTD. La table BGP RTB ressemble à ceci :

<#root>

```
RTB#
```

show ip bgp

```
BGP table version is 24, local router ID is 192.168.250.2 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin
codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
 h 192.168.10.0     192.168.10.5           0             0 300 i
*> 192.168.250.15     0.0.0.0                0         32768 i
```

L’entrée BGP pour 192.168.10.0 est dans un état `historique`. Cet emplacement signifie que vous n'avez pas de meilleur chemin pour la route, mais des informations sur l'oscillation de la route existent.

<#root>

```
RTB#
```

show ip bgp 192.168.10.0

```
BGP routing table entry for 192.168.10.0 255.255.255.0, version 25
Paths: (1 available, no best path)
300 (history entry)
    192.168.10.5 from 192.168.10.5 (192.168.208.174)
Origin IGP, metric 0, external
Dampinfo: penalty 910, flapped 1 times in 0:02:03
```

La route a reçu une pénalité pour l’intermittence, mais la pénalité est toujours sous la limite de suppression. 2000 est établi par défaut. La suppression de la route ne s'est pas encore produite. Si la route oscille encore, vous verrez que :

```
    <#root>

RTB#
```

show ip bgp

```
BGP table version is 32, local router ID is 192.168.250.2 Status codes:
s suppressed, d damped, h history, * valid, > best, i - internal Origin codes:
i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*d 192.168.10.0     192.168.10.5           0             0  300 i
*> 192.168.250.15     0.0.0.0               0       32768  i

RTB#
```

show ip bgp 192.168.10.0

```
BGP routing table entry for 192.168.10.0 255.255.255.0, version 32
```

```
Paths: (1 available, no best path)
300, (suppressed due to dampening)
192.168.10.5 from 192.168.10.5 (192.168.208.174)
      Origin IGP, metric 0, valid, external
Dampinfo: penalty 2615, flapped 3 times in 0:05:18 , reuse in 0:27:00
```

Le routage a été atténué ou supprimé. La route est réutilisée quand la pénalité atteint la « valeur de réutilisation ». Dans ce cas, la valeur de réutilisation est par défaut de 750. Les informations de dampening sont purgées quand la pénalité devient inférieure à la moitié de la limite de réutilisation. Dans ce cas, la purge se produit quand la pénalité atteint 375 (750/2=375). Ces commandes affichent et effacent les données statistiques de l'oscillation :

- **show ip bgp flap-statistics - Affiche les statistiques de l'affollement pour tous les chemins.**

- **show ip bgp flap-statistics regexpregular-expression** – Affiche les statistiques d’intermittence pour tous les chemins qui correspondent à l’expression régulière.

- **show ip bgp flap-statistics filter-listlist** – Affiche les statistiques d’intermittence pour tous les chemins qui correspondent au filtre.

- **show ip bgp flap-statistics A.B.C.D m.m.m.m** – Affiche les statistiques d’intermittence pour une entrée unique.

- **show ip bgp flap-statistics A.B.C.D m.m.m.mlonger-prefix** – Affiche les statistiques d’intermittence pour des entrées plus précises.

- **show ip bgp neighbor [dampened-routes] | [flap-statistics]**– Affiche les statistiques d’intermittence pour tous les chemins d’un voisin.

- **clear ip bgp flap-statistics — Efface les statistiques de l'affollement pour toutes les routes.**

- **clear ip bgp flap-statistics regexpregular-expression**– Efface les statistiques d’intermittence pour tous les chemins qui correspondent à l’expression régulière.

- **clear ip bgp flap-statistics filter-listlist**– Efface les statistiques d’intermittence pour tous les chemins qui correspondent au filtre.

- **clear ip bgp flap-statisticsA.B.C.D m.m.m.m**– Efface les statistiques d’intermittence pour une entrée unique.

- **clear ip bgpA.B.C.Dflap-statistics** – Efface les statistiques d’intermittence pour tous les chemins d’un voisin.

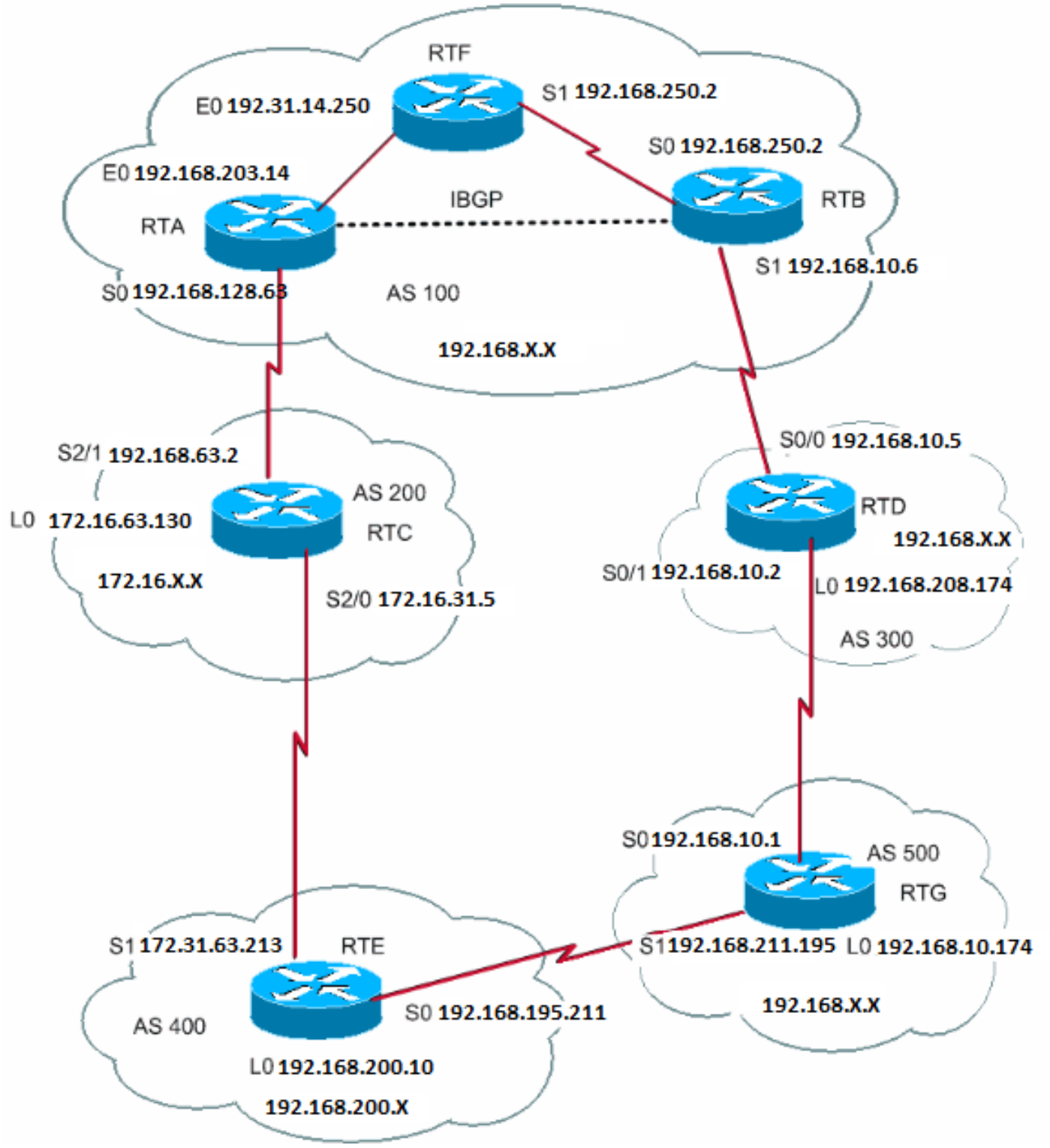
Comment BGP sélectionne un chemin

Maintenant que vous connaissez les attributs et la terminologie BGP, référez-vous à la section Algorithme de sélection du meilleur chemin BGP.

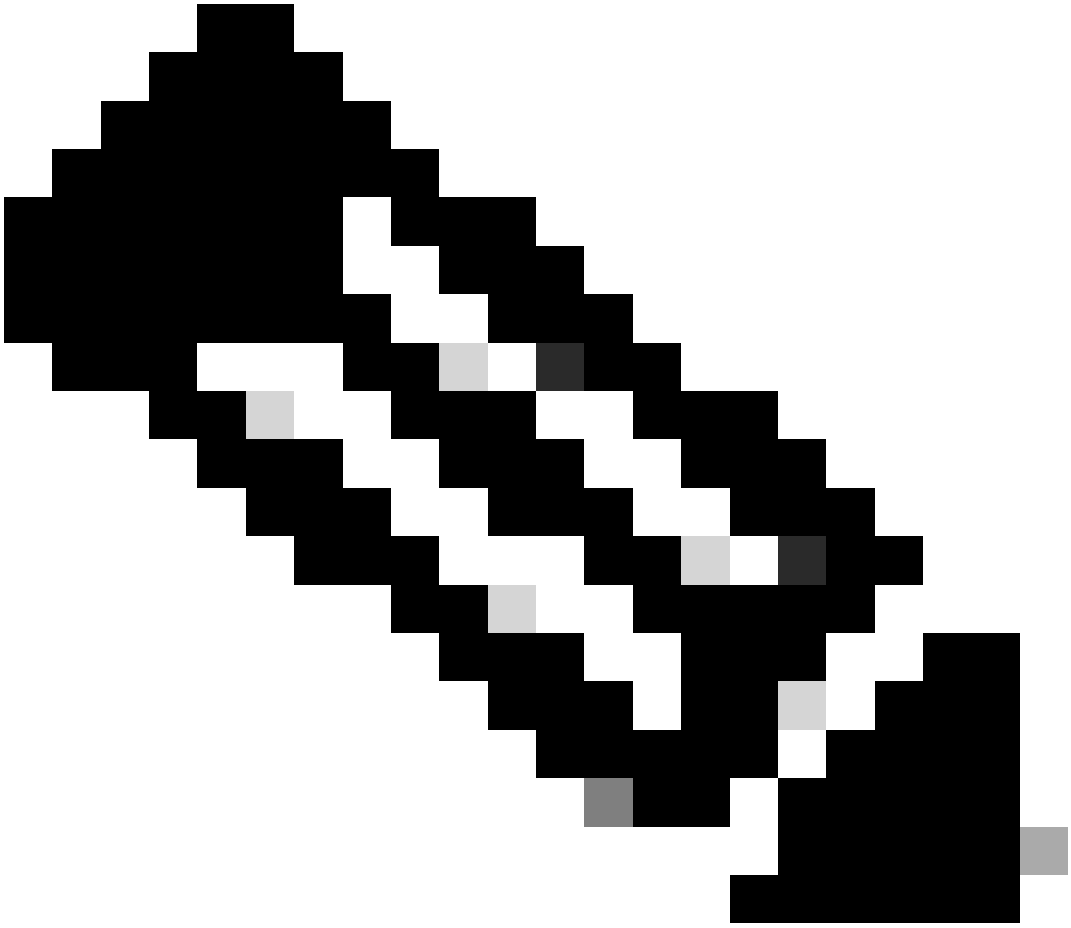
Études de cas BGP 5

Exemple de projet pratique

Cette section inclut un exemple de projet qui montre les tables de configuration et de routage telles qu'elles apparaissent réellement sur des routeurs Cisco.



Cette section montre comment construire cette configuration pas à pas et les problèmes potentiellement rencontrés. Toutes les fois que vous avez un AS qui se connecte à deux ISP par l'intermédiaire d'eBGP, exécutez toujours iBGP dans votre AS afin de mieux contrôler vos routes. Dans cet exemple, iBGP s'exécute dans AS100 entre RTA et RTB, et OSPF s'exécute comme un IGP. Supposez que vous vous connectez à deux ISP, AS200 et AS300. Voici la première exécution des configurations pour tous les routeurs :



Remarque : Ces configurations ne sont pas les configurations finales.

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 192.168.203.250 255.255.255.0

interface Ethernet0
ip address 192.168.203.14 255.255.255.0

interface Serial0
```

```
 ip address 192.168.128.63 255.255.255.252

router ospf 10
 network 192.168.203.25 0.0.255.255 area 0

router bgp 100
 network 192.168.203.13
 network 192.168.250.14
 neighbor 172.31.63.250 remote-as 200
 neighbor 192.168.250.2 remote-as 100
 neighbor 192.168.250.2 update-source Loopback0

RTF#
hostname RTF

ip subnet-zero

interface Ethernet0
 ip address 172.31.14.250 255.255.255.0

interface Serial1
 ip address 172.16.15.250 255.255.255.252

router ospf 10
 network 192.168.203.25 0.0.255.255 area 0

RTB#
hostname RTB

ip subnet-zero

interface Serial0
 ip address 192.168.250.2 255.255.255.252

interface Serial1
 ip address 192.168.10.6 255.255.255.252

router ospf 10
 network 192.168.203.25 0.0.255.255 area 0

router bgp 100
network 192.168.250.15
 neighbor 192.168.10.5 remote-as 300
 neighbor 192.168.203.250 remote-as 100

RTC#
hostname RTC

ip subnet-zero

interface Loopback0
 ip address 192.168.128.6330 255.255.255.192

interface Serial2/0
 ip address 172.16.31.5 255.255.255.252
!
interface Serial2/1
 ip address 172.31.63.250 255.255.255.252

router bgp 200
 network 172.31.10.0
 neighbor 192.168.128.63 remote-as 100
```

```
 neighbor 172.31.63.213 remote-as 400

RTD#
hostname RTD

ip subnet-zero

interface Loopback0
ip address 192.168.208.174 255.255.255.192

interface Serial0/0
 ip address 192.168.10.5 255.255.255.252
!
interface Serial0/1
 ip address 192.168.10.2 255.255.255.252

router bgp 300
 network 192.168.10.0
 neighbor 192.168.10.1 remote-as 500
 neighbor 192.168.10.6 remote-as 100

RTE#
hostname RTE

ip subnet-zero

interface Loopback0
ip address 192.168.200.10 255.255.255.0

interface Serial0
 ip address 192.168.195.211 255.255.255.252

interface Serial1
 ip address 172.31.63.213 255.255.255.252
 clockrate 1000000

router bgp 400
 network 192.168.10.10
 neighbor 172.16.31.5 remote-as 200
 neighbor 192.168.211.195 remote-as 500

RTG#
hostname RTG

ip subnet-zero

interface Loopback0
 ip address 192.168.211.19574 255.255.255.192

interface Serial0
 ip address 192.168.10.1 255.255.255.252

interface Serial1
 ip address 192.168.211.195 255.255.255.252

router bgp 500
 network 192.168.211.10
 neighbor 192.168.10.2 remote-as 300
 neighbor 192.168.195.211 remote-as 400
```

Utilisez toujours la network commande ou redistribuez les entrées statiques dans BGP pour annoncer les réseaux. Cette méthode est préférable à la redistribution d'IGP dans BGP. Cet exemple utilise la network commande pour injecter des réseaux dans BGP.

Ici, vous commencez avec l'interface s1 à l'arrêt de RTB, comme si le lien entre RTB et RTD n'existait pas. Voici la table BGP RTB :

<#root>

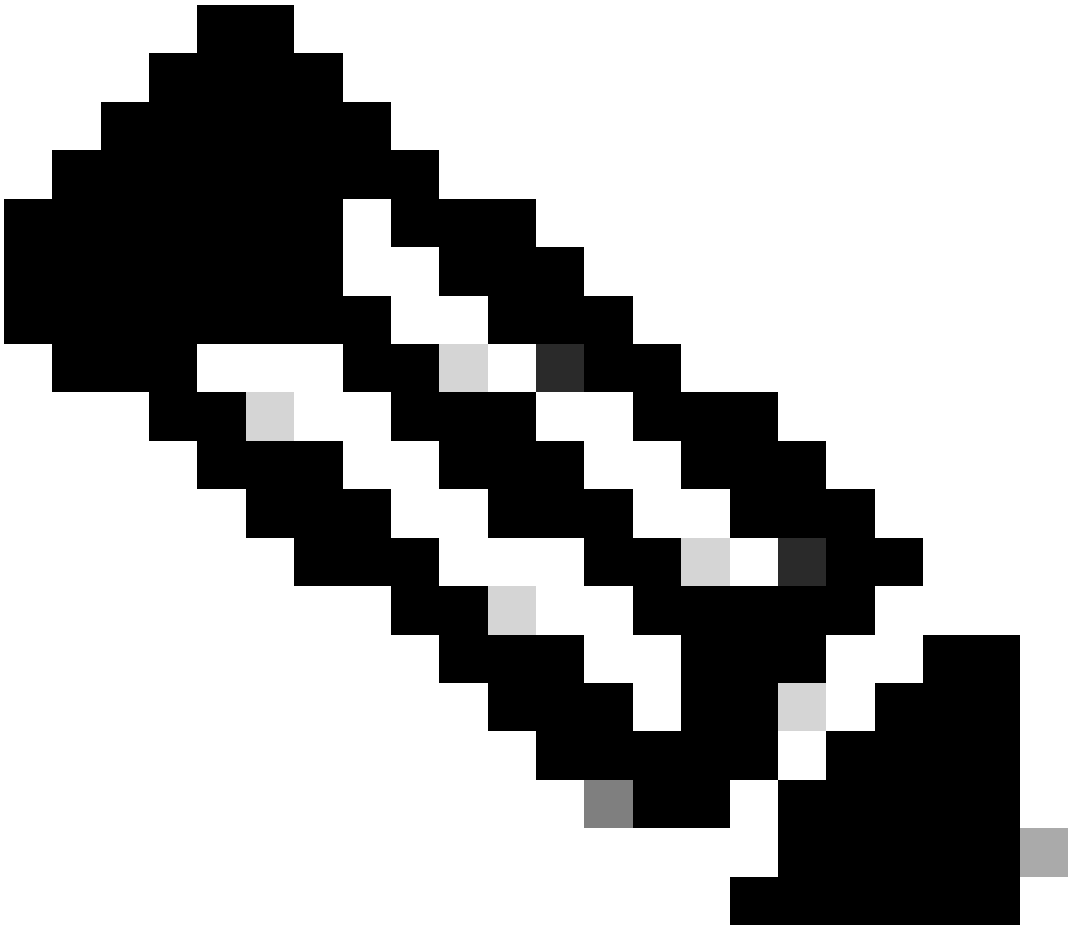
RTB#

show ip bgp BGP

```
table version is 4, local router ID is 192.168.250.2 Status
codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
     Network          Next Hop          Metric LocPrf Weight Path
*i172.31.10.0      172.31.63.250           0    100      0 200 i
*i192.168.10.0     172.31.63.250                100      0 200 400 500
300 i
*i192.168.211.10     172.31.63.250                100      0 200 400 500 i
*i192.168.10.10     172.31.63.250                100      0 200 400 i
*>i192.168.203.13    192.168.203.250          0    100      0 i
*>i192.168.250.14    192.168.203.250          0    100      0 i
*>192.168.250.15     0.0.0.0                  0         32768 i
```

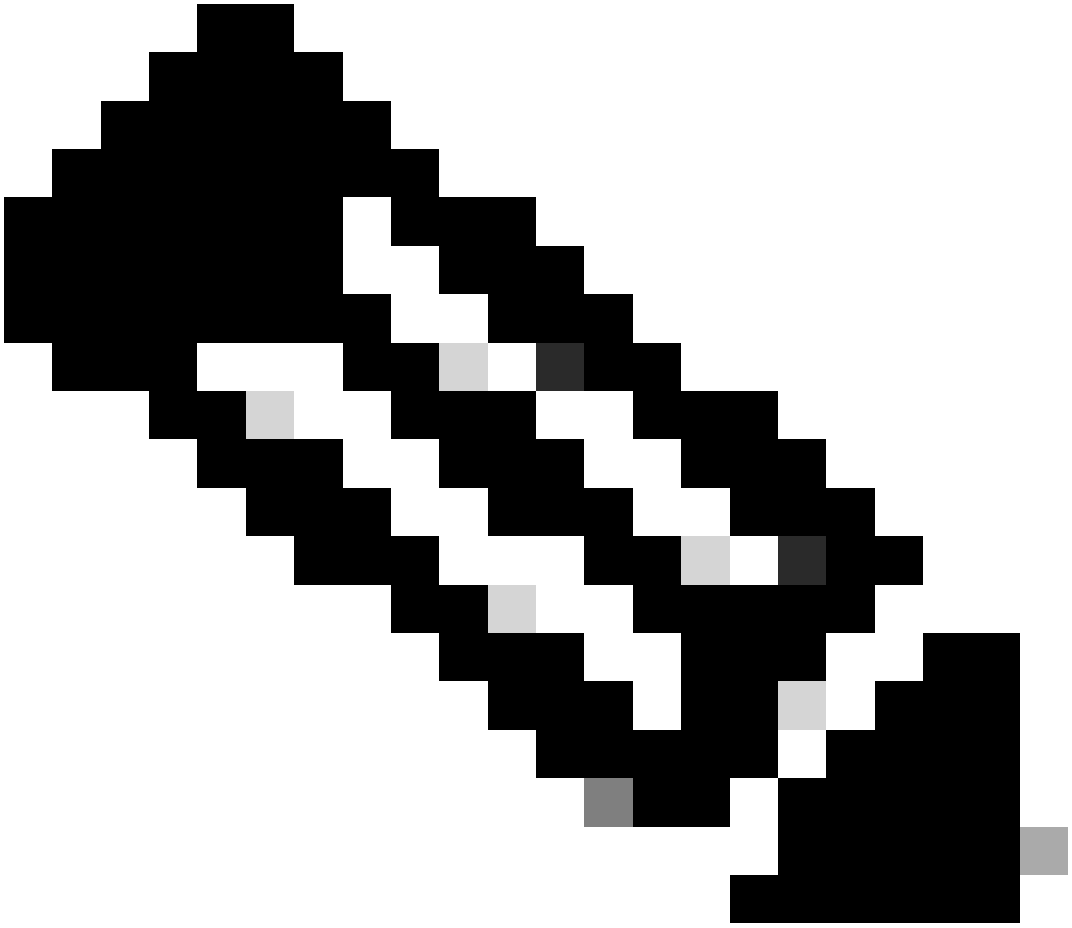
Dans cette table, les notations suivantes apparaissent :

- Un `i` au début : indique que l’entrée a été détectée par un homologue iBGP.
- Un `i` à la fin : indique que l’origine des détails sur le chemin est IGP.
- `Path information` : les détails sur le chemin sont intuitifs. Par exemple, le réseau 172.31.10.0 est appris par l'intermédiaire du chemin 200 avec un prochain saut de 172.31.63.250.



Remarque : Toute entrée générée localement, comme 192.168.250.15, a un prochain saut à 0.0.0.0.

- Un symbole > indique que BGP a choisi la meilleure route. BGP utilise les étapes décisionnelles que le document Algorithme de sélection du meilleur chemin BGP souligne. BGP sélectionne le meilleur chemin pour atteindre une destination, installe le chemin dans la table de routage IP et annonce le chemin aux autres homologues BGP.



Remarque : Remarquez l’attribut du prochain saut. RTB apprend 172.31.10.0 par l'intermédiaire d'un prochain saut de 172.31.63.250, qui est le prochain saut d'eBGP porté dans iBGP.

Regardez la table de routage IP :

<#root>

RTB#

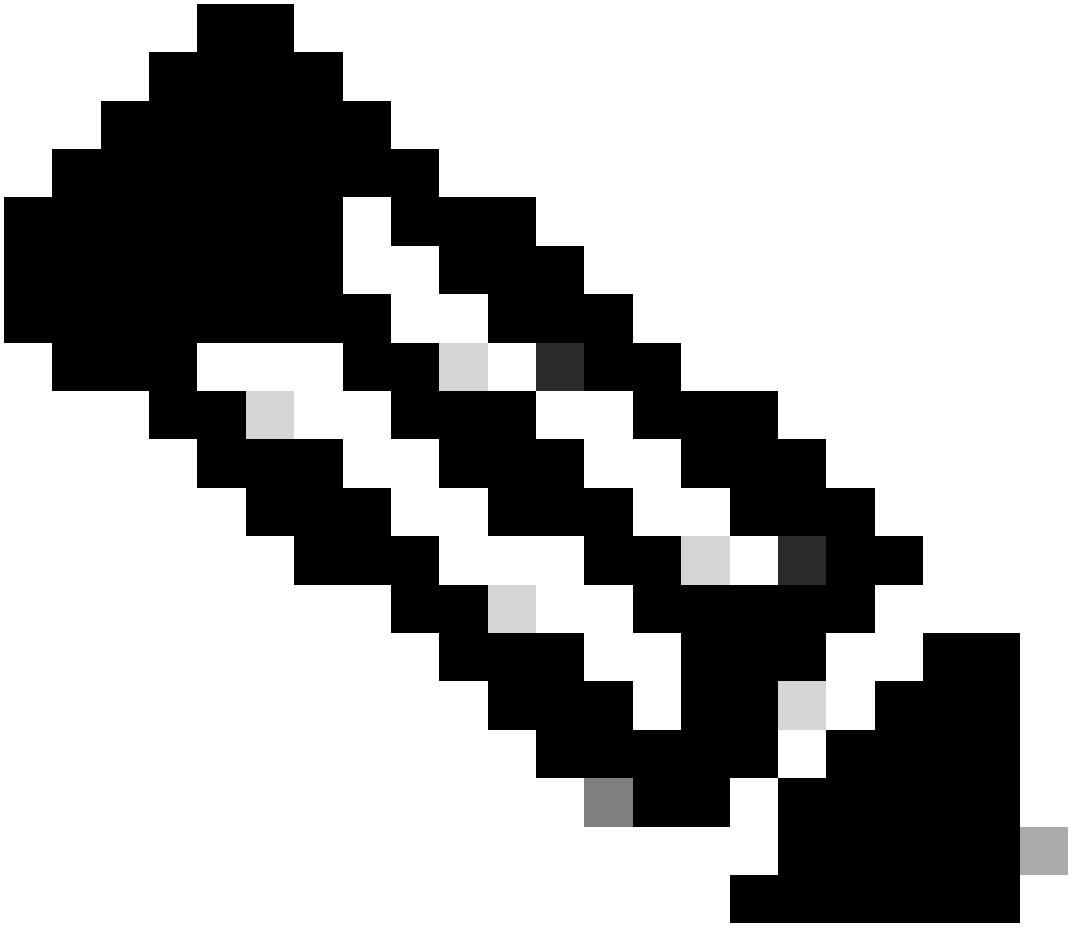
show ip route

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default

Gateway of last resort is not set

     192.168.203.13 255.255.255.255 is subnetted, 1 subnets
O       192.168.203.250 [110/75] via 172.16.15.250, 02:50:45, Serial0
     192.168.250.15 255.255.255.252 is subnetted, 1 subnets
C       192.168.250.15 is directly connected, Serial0
O    192.168.250.14 [110/74] via 172.16.15.250, 02:50:46, Serial0
```

Apparemment, aucune des entrées BGP n'a atteint la table de routage. Deux problèmes se posent.

Le premier problème est que le prochain saut pour ces entrées, 172.31.63.250, est inaccessible. Il n'y a aucun moyen d'atteindre ce prochain saut par l'intermédiaire de cet IGP, qui est OSPF. RTB n'a pas a appris 192.168.213.63 par l'intermédiaire d'OSPF. Vous pouvez exécuter OSPF sur l'interface s0 de RTA et la rendre passive. Ainsi, le RTB sait comment atteindre le prochain saut 172.31.63.250. Cette configuration d'RTA est indiquée ici :

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 192.168.203.250 255.255.255.0

interface Ethernet0
ip address 192.168.203.14 255.255.255.0

interface Serial0
 ip address 192.168.128.63 255.255.255.252

router ospf 10
 passive-interface Serial0
 network 192.168.203.25 0.0.255.255 area 0
 network 172.31.10.0 0.0.255.255 area 0

router bgp 100
 network 192.168.203.25 mask 255.255.0.0
 neighbor 172.31.63.250 remote-as 200
 neighbor 192.168.250.2 remote-as 100
 neighbor 192.168.250.2 update-source Loopback0
```



Remarque : vous pouvez émettre la commandebgp nexthop self entre RTA et RTB afin de modifier le saut suivant.

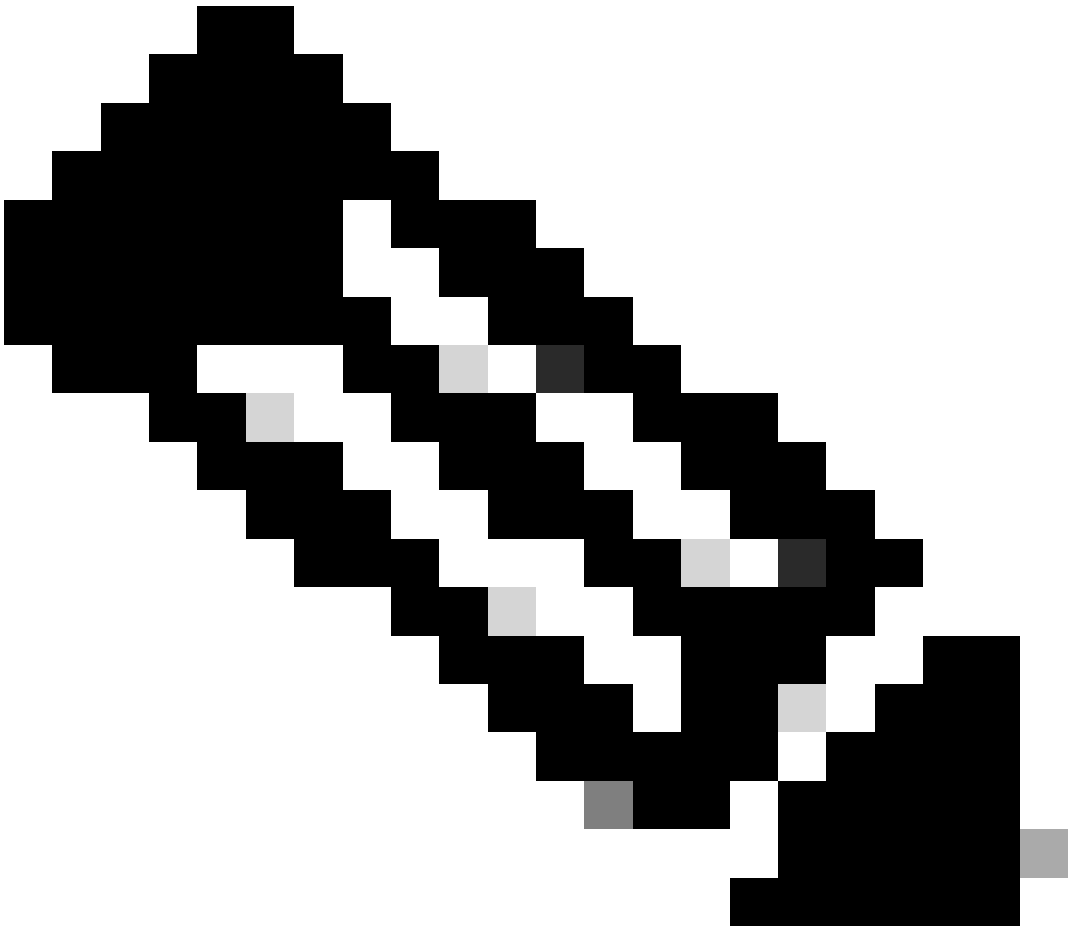
La nouvelle table BGP sur RTB ressemble à ceci :

<#root>

RTB#

show ip bgp

```
BGP table version is 10, local router ID is 192.168.250.2
Status codes: s suppressed, d damped, h history, * valid, > best,
i - internal Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*>i172.31.10.0      172.31.63.250           0    100      0 200 i
*>i192.168.10.0     172.31.63.250                100      0 200 400 500
300 i
*>i192.168.211.10     172.31.63.250                100      0 200 400 500 i
*>i192.168.10.10     172.31.63.250                100      0 200 400 i
*>i192.168.203.13     192.168.203.250          0    100      0 i
*>i192.168.250.14     192.168.203.250          0    100      0 i
*> 192.168.250.15     0.0.0.0                  0         32768 i
```

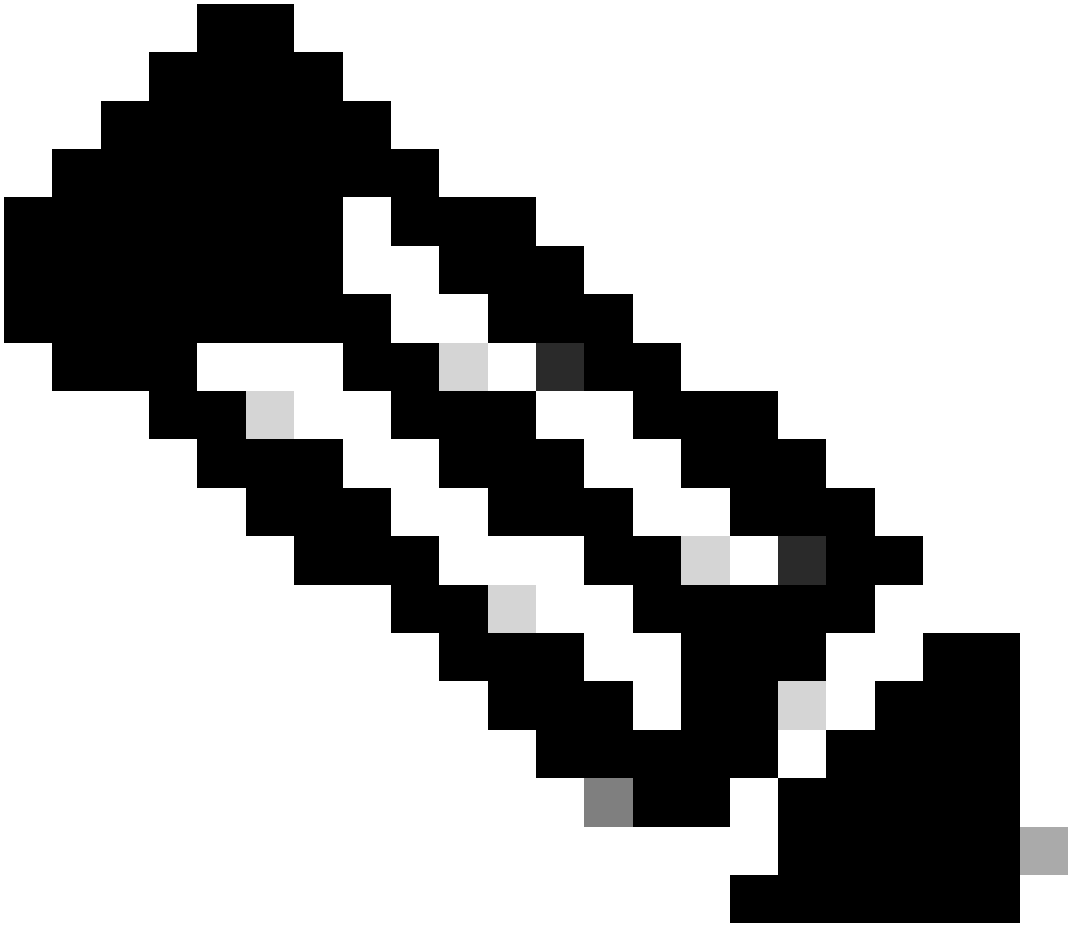


Remarque : Toutes les entrées ont la valeur « > », ce qui signifie que le BGP peut atteindre le prochain saut.

Regardez la table de routage :

```
<#root>

RTB#

show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * -
candidate default

Gateway of last resort is not set

     192.168.203.13 255.255.255.255 is subnetted, 1 subnets
O       192.168.203.250 [110/75] via 172.16.15.250, 00:04:46, Serial0
     192.168.250.15 255.255.255.252 is subnetted, 1 subnets
C       192.168.250.15 is directly connected, Serial0
O    192.168.250.14 [110/74] via 172.16.15.250, 00:04:46, Serial0
     172.31.10.0 255.255.255.252 is subnetted, 1 subnets
O       192.168.213.63 [110/138] via 172.16.15.250, 00:04:47, Serial0
```

Le second problème est que vous ne voyez toujours pas les entrées BGP dans la table de routage. La seule différence est que 192.168.213.63 est maintenant accessible par l'intermédiaire d'OSPF. Ce problème est un problème de synchronization. BGP ne met pas ces entrées dans la table de routage et ne les envoie pas dans les mises à jour BGP en raison d'un manque de synchronization avec IGP.



Remarque : Le RTF n’a aucune connaissance de l’existence des réseaux 192.168.10.0 et 192.168.211.10, car vous n’avez pas encore redistribué le BGP dans OSPF.

Dans ce scénario, si vous désactivez la synchronization, les entrées apparaissent dans la table de routage. Mais la connectivité reste interrompue.

Si vous désactivez la synchronization sur RTB, voici ce qui se produit :

<#root>

RTB#

show ip route

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * -
candidate default

Gateway of last resort is not set

B    192.168.10.10 [200/0] via 172.31.63.250, 00:01:07
B    192.168.211.10 [200/0] via 172.31.63.250, 00:01:07
B    192.168.10.0 [200/0] via 172.31.63.250, 00:01:07
     192.168.203.13 is variably subnetted, 2 subnets, 2 masks
O       192.168.203.250 255.255.255.255
           [110/75] via 172.16.15.250, 00:12:37, Serial0
B       192.168.203.13 255.255.255.0 [200/0] via 192.168.203.250, 00:01:08
     192.168.250.15 255.255.255.252 is subnetted, 1 subnets
C       192.168.250.15 is directly connected, Serial0
O    192.168.250.14 [110/74] via 172.16.15.250, 00:12:37, Serial0
     172.31.10.0 is variably subnetted, 2 subnets, 2 masks
B       172.31.10.0 255.255.0.0 [200/0] via 172.31.63.250, 00:01:08
O       192.168.213.63 255.255.255.252
           [110/138] via 172.16.15.250, 00:12:37, Serial0
```

La table de routage semble correcte, mais il n'y a aucun moyen d'atteindre ces réseaux. RTF au milieu ne sait pas comment atteindre les réseaux :

<#root>

```
RTF#
```

show ip route

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * -
candidate default

Gateway of last resort is not set
```

```
     192.168.203.13 255.255.255.255 is subnetted, 1 subnets
O       192.168.203.250 [110/11] via 192.168.203.14, 00:14:15, Ethernet0
     192.168.250.15 255.255.255.252 is subnetted, 1 subnets
C       192.168.250.15 is directly connected, Serial1
C    192.168.250.14 is directly connected, Ethernet0
     172.31.10.0 255.255.255.252 is subnetted, 1 subnets
O       192.168.213.63 [110/74] via 192.168.203.14, 00:14:15, Ethernet0
```

Lorsque vous désactivez la synchronization dans cette situation, le problème persiste. Mais vous aurez besoin de la synchronization plus tard pour résoudre d'autres problèmes. Redistribuez BGP dans OSPF sur RTA, avec une métrique de 2000 :

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 192.168.203.250 255.255.255.0

interface Ethernet0
ip address 192.168.203.14 255.255.255.0

interface Serial0
 ip address 192.168.128.63 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 2000 subnets
 passive-interface Serial0
 network 192.168.203.25 0.0.255.255 area 0
 network 172.31.10.0 0.0.255.255 area 0

router bgp 100
 network 192.168.203.25 mask 255.255.0.0
 neighbor 172.31.63.250 remote-as 200
 neighbor 192.168.250.2 remote-as 100
 neighbor 192.168.250.2 update-source Loopback0
```

La table de routage ressemble à ceci :


```
RTB#
```

show ip route

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * -
candidate default

Gateway of last resort is not set

O E2 192.168.10.10 [110/2000] via 172.16.15.250, 00:00:14, Serial0
O E2 192.168.211.10 [110/2000] via 172.16.15.250, 00:00:14, Serial0
O E2 192.168.10.0 [110/2000] via 172.16.15.250, 00:00:14, Serial0
     192.168.203.13 is variably subnetted, 2 subnets, 2 masks
O       192.168.203.250 255.255.255.255
           [110/75] via 172.16.15.250, 00:00:15, Serial0
O E2    192.168.203.13 255.255.255.0
           [110/2000] via 172.16.15.250, 00:00:15, Serial0
     192.168.250.15 255.255.255.252 is subnetted, 2 subnets
C       172.31.250.8 is directly connected, Loopback1
C       192.168.250.15 is directly connected, Serial0
O    192.168.250.14 [110/74] via 172.16.15.250, 00:00:15, Serial0
     172.31.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    172.31.10.0 255.255.0.0 [110/2000] via 172.16.15.250,
00:00:15,Serial0
O       192.168.213.63 255.255.255.252
           [110/138] via 172.16.15.250, 00:00:16, Serial0
```

Les entrées BGP ont disparu parce qu'OSPF a une meilleure distance qu'iBGP. La distance OSPF est 110, alors que la distance iBGP est 200.

Désactivez la synchronization sur RTA de sorte que RTA puisse annoncer 192.168.250.15. Cette action est nécessaire parce que RTA ne se synchronise pas avec OSPF en raison de la différence de masques. Gardez la synchronization désactivée sur RTB de sorte que RTB puisse annoncer 192.168.203.13. Cette action est nécessaire sur RTB pour la même raison.

Maintenant, constituez l'interface s1 RTB pour voir ce à quoi ressemblent les routes. En outre, activez OSPF sur la série 1 de RTB pour le rendre passif. Cette étape permet à RTA d'apprendre le prochain saut 192.168.10.5 par l'intermédiaire d'IGP. Si vous ne prenez pas cette précaution, des boucles de routage se produisent car, afin d'atteindre le prochain saut 192.168.10.5, vous devez aller dans l'autre direction via eBGP. Voici les nouvelles configurations de RTA et de RTB :

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 192.168.203.250 255.255.255.0

interface Ethernet0
ip address 192.168.203.14 255.255.255.0

interface Serial0
 ip address 192.168.128.63 255.255.255.252
```

```
router ospf 10
 redistribute bgp 100 metric 2000 subnets
 passive-interface Serial0
 network 192.168.203.25 0.0.255.255 area 0
 network 172.31.10.0 0.0.255.255 area 0

router bgp 100
 no synchronization
 network 192.168.203.13
 network 192.168.250.14
 neighbor 172.31.63.250 remote-as 200
 neighbor 192.168.250.2 remote-as 100
 neighbor 192.168.250.2 update-source Loopback0

RTB#
hostname RTB

ip subnet-zero

interface Serial0
 ip address 192.168.250.2 255.255.255.252

interface Serial1
 ip address 192.168.10.6 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 1000 subnets
 passive-interface Serial1
 network 192.168.203.25 0.0.255.255 area 0
 network 192.168.208.0 0.0.255.255 area 0

router bgp 100
 no synchronization
 network 192.168.250.15
 neighbor 192.168.10.5 remote-as 300
 neighbor 192.168.203.250 remote-as 100
```

Les tables BGP ressemblent à ceci :

<#root>

```
RTA#
```

show ip bgp

```
BGP table version is 117, local router ID is 192.168.203.250
Status codes: s suppressed, d damped, h history, * valid, > best,
```

```
i -internal Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 172.31.10.0      172.31.63.250           0             0 200 i
*>i192.168.10.0     192.168.10.5            0    100      0 300 i
*>i192.168.211.10     192.168.10.5                100      0 300 500 i
*                   172.31.63.250                         0 200 400 500 i
*> 192.168.10.10     172.31.63.250                        0 200 400 i
*> 192.168.203.13     0.0.0.0                 0         32768 i
*> 192.168.250.14     0.0.0.0                 0         32768 i
*>i192.168.250.15     192.168.250.2             0    100      0 i

RTB#
```

show ip bgp

```
BGP table version is 12, local router ID is 172.16.15.2500
Status codes: s suppressed, d damped, h history, * valid, > best,
i -internal Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*>i172.31.10.0      172.31.63.250           0    100      0 200 i
*                   192.168.10.5                          0 300 500 400
200 i
*> 192.168.10.0     192.168.10.5            0             0 300 i
*> 192.168.211.10     192.168.10.5                        0 300 500 i
*>i192.168.10.10     172.31.63.250                100      0 200 400 i
*                   192.168.10.5                          0 300 500 400 i
*>i192.168.203.13     192.168.203.250           0    100      0 i
*>i192.168.250.14     192.168.203.250           0    100      0 i
*> 192.168.250.15     0.0.0.0                 0         32768 i
```

Il y a plusieurs façons de concevoir votre réseau pour parler aux deux différents IPS, AS200 et AS300. L'une consiste à utiliser un ISP principal et un ISP de secours. Vous pouvez apprendre les routes partielles de l'un des ISP et les routes par défaut aux deux ISP. Dans cet exemple, vous recevez les routes partielles d'AS200 et seulement les routes locales d'AS300. RTA et RTB produisent des routes par défaut dans OSPF, avec RTB défini comme préférence en raison de la métrique inférieure. De cette façon, vous pouvez équilibrer le trafic sortant entre les deux ISP.

Une asymétrie potentielle peut se produire si le trafic qui quitte RTA revient par l'intermédiaire de RTB. Cette situation peut se produire si vous utilisez le même pool d'adresses IP, le même réseau principal, quand vous parlez aux deux ISP. En raison de l'agrégation, votre AS global peut apparaître comme une entité globale au monde extérieur. Les points d'entrée à votre network peuvent se produire par l'intermédiaire de RTA ou de RTB. Vous pouvez voir que tout le trafic entrant de votre AS arrive par l'intermédiaire d'un point unique, même si vous ayez plusieurs points à l'Internet. Dans l'exemple, vous avez deux réseaux principaux différents quand vous parlez aux deux ISP.

Une autre raison potentielle d'asymétrie est la longueur différente du chemin annoncé pour atteindre votre AS. Peut-être qu'un fournisseur de services est plus près d'une certaine destination que l'autre. Dans l'exemple, le trafic d'AS400 qui a votre réseau comme destination entre toujours par l'intermédiaire de RTA car le chemin est plus court. Vous pouvez essayer de rendre effective cette décision. Vous pouvez utiliser la commande set as-path prepend pour préfixer des numéros de chemin à vos mises à jour et faire en sorte que la longueur du chemin paraisse plus

longue. Mais, avec des attributs tels que la préférence locale, la métrique ou le poids, AS400 peut avoir défini le point de sortie comme étant AS200. Dans ce cas, il n'y a rien que vous puissiez faire.

Cette configuration est la configuration finale pour tous les routeurs :

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 192.168.203.250 255.255.255.0

interface Ethernet0
 ip address 192.168.203.14 255.255.255.0

interface Serial0
 ip address 192.168.128.63 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 2000 subnets
 passive-interface Serial0
 network 192.168.203.25 0.0.255.255 area 0
 network 172.31.10.0 0.0.255.255 area 0
 default-information originate metric 2000

router bgp 100
 no synchronization
 network 192.168.203.13
 network 192.168.250.14
 neighbor 172.31.63.250 remote-as 200
 neighbor 172.31.63.250 route-map setlocalpref in
 neighbor 192.168.250.2 remote-as 100
 neighbor 192.168.250.2 update-source Loopback0

ip classless
ip default-network 172.31.200.200

route-map setlocalpref permit 10
 set local-preference 200
```

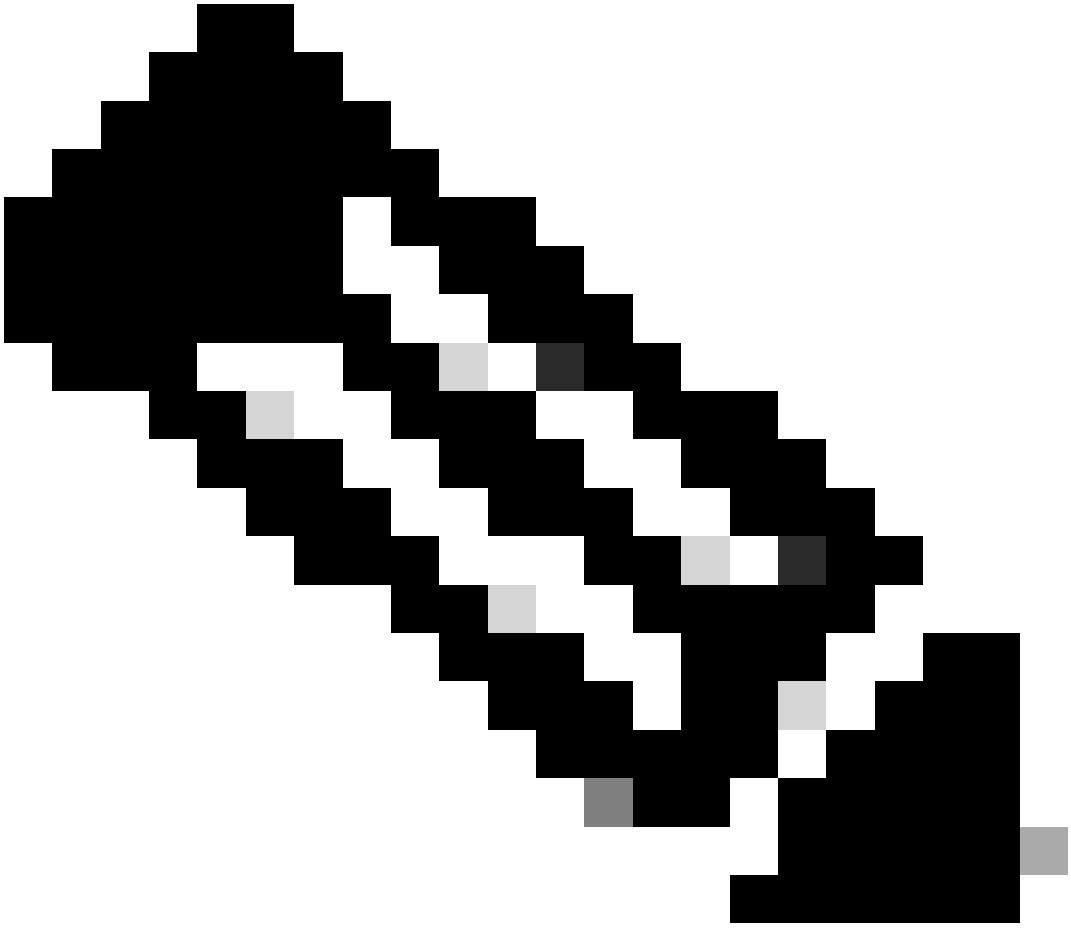
Sur RTA, la préférence locale pour les routes qui viennent d'AS200 est définie à 200. En outre, le réseau 172.31.200.200 est choisi comme candidat par défaut. La commande ip default-network permet de choisir le réseau par défaut.

De plus, dans cet exemple, l'utilisation de la commande default-information originate **avec OSPF injecte la route par default dans le domaine OSPF.** Cette exemple utilise également cette commande avec le protocole IS-IS (Intermediate System-to-Intermediate System) et BGP. Pour RIP, il y a une redistribution automatique dans RIP de 0.0.0.0, sans configuration supplémentaire. Pour IGRP et EIGRP, l'injection des informations par défaut dans le domaine IGP se produit après la redistribution de BGP dans IGRP et EIGRP. En outre, avec IGRP et EIGRP, vous pouvez redistribuer une route statique à 0.0.0.0 dans le domaine IGP.

```
RTF#
```

```
hostname RTF

ip subnet-zero

interface Ethernet0
 ip address 172.31.14.250 255.255.255.0

interface Serial1
 ip address 172.16.15.250 255.255.255.252

router ospf 10
 network 192.168.203.25 0.0.255.255 area 0

ip classless

RTB#
hostname RTB

ip subnet-zero

interface Loopback1
 ip address 172.16.15.2500 255.255.255.252

interface Serial0
 ip address 192.168.250.2 255.255.255.252
!
interface Serial1
 ip address 192.168.10.6 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 1000 subnets
 passive-interface Serial1
 network 192.168.203.25 0.0.255.255 area 0
 network 192.168.10.6 0.0.0.0 area 0
 default-information originate metric 1000
!
router bgp 100
 no synchronization
 network 192.168.250.15
 neighbor 192.168.10.5 remote-as 300
 neighbor 192.168.10.5 route-map localonly in
 neighbor 192.168.203.250 remote-as 100
!
ip classless
ip default-network 192.168.10.0
ip as-path access-list 1 permit ^300$

route-map localonly permit 10
 match as-path 1
 set local-preference 300
```

Pour RTB, la préférence locale pour les mises à jour qui viennent d'AS300 est définie à 300. Cette valeur est plus haute que la valeur de la préférence locale des mises à jour iBGP qui viennent d'RTA. De cette façon, AS100 sélectionne RTB pour les routes locales d'AS300. Toutes les autres routes sur RTB, si d'autres routes existent, transmettent en interne avec une préférence locale de 100. Cette valeur est inférieure à la préférence locale de 200, qui vient de RTA. Le RTA est privilégié.



Remarque : Vous n’avez annoncé que les routes locales AS300. Toutes les informations de chemin qui ne correspondent pas à ^300$ sont rejetées. Si vous voulez annoncer les routes locales et les routes voisines, qui sont les clients de l'ISP, utilisez ^300_[0-9]*.

Voici les résultats de l'expression régulière qui indique les routes locales AS300 :

<#root>

RTB#

show ip bgp regexp ^300$

```
BGP table version is 14, local router ID is 172.16.15.2500
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 192.168.10.0     192.168.10.5           0    300      0 300
```

```
RTC#
hostname RTC

ip subnet-zero

interface Loopback0
 ip address 192.168.128.6330 255.255.255.192

interface Serial2/0
 ip address 172.16.31.5 255.255.255.252
!
interface Serial2/1
 ip address 172.31.63.250 255.255.255.252

router bgp 200
 network 172.31.10.0
 neighbor 192.168.128.63 remote-as 100
 neighbor 192.168.128.63 distribute-list 1 out
 neighbor 172.31.63.213 remote-as 400

ip classless
access-list 1 deny   192.168.211.0 0.0.255.255
access-list 1 permit any
```

Sur RTC, vous agrégez 172.31.10.0/16 et indiquez les routes spécifiques pour l'injection dans AS100. Si l'ISP refuse d'exécuter cette tâche, vous devez filtrer sur la fin entrante d'AS100.

```
RTD#
hostname RTD

ip subnet-zero

interface Loopback0
 ip address 192.168.208.174 255.255.255.192
!
interface Serial0/0
 ip address 192.168.10.5 255.255.255.252
!
interface Serial0/1
 ip address 192.168.10.2 255.255.255.252
```

```
router bgp 300
 network 192.168.10.0
 neighbor 192.168.10.1 remote-as 500
 neighbor 192.168.10.6 remote-as 100

RTG#
hostname RTG

ip subnet-zero

interface Loopback0
 ip address 192.168.211.19574 255.255.255.192

interface Serial0
 ip address 192.168.10.1 255.255.255.252

interface Serial1
 ip address 192.168.211.195 255.255.255.252

router bgp 500
 network 192.168.211.10
 aggregate-address 192.168.211.0 255.255.0.0 summary-only
 neighbor 192.168.10.2 remote-as 300
 neighbor 192.168.10.2 send-community
 neighbor 192.168.10.2 route-map setcommunity out
 neighbor 192.168.195.211 remote-as 400
!
ip classless
access-list 1 permit 192.168.211.0 0.0.255.255
access-list 2 permit any
route-map setcommunity permit 20
 match ip address 2
!
route-map setcommunity permit 10
 match ip address 1
 set community no-export
```

Une démonstration de l'utilisation du filtrage de communauté est fournie sur le RTG. Vous ajoutez une no-export communauté aux mises à jour 192.168.211.0 vers RTD. De cette façon, RTD n'exporte pas cette route vers RTB. Cependant, dans ce cas, RTB n'accepte pas ces routes de toute façon.

```
RTE#
hostname RTE

ip subnet-zero

interface Loopback0
 ip address 192.168.200.10 255.255.255.0

interface Serial0
 ip address 192.168.195.211 255.255.255.252

interface Serial1
 ip address 172.31.63.213 255.255.255.252
```

```
router bgp 400
 network 192.168.10.10
 aggregate-address 172.31.200.200 255.255.0.0 summary-only
 neighbor 172.16.31.5 remote-as 200
 neighbor 192.168.211.195 remote-as 500

ip classless
```

RTE agrège 172.31.200.200/16. Voici le BGP final et les tables de routage pour RTA, RTF et RTB :

```
   <#root>


RTA#
```

show ip bgp

```
BGP table version is 21, local router ID is 192.168.203.250
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 172.31.10.0      172.31.63.250           0    200      0 200 i
*>i192.168.10.0     192.168.10.5            0    300      0 300 i
*> 172.31.200.200/16   172.31.63.250               200      0 200 400 i
*> 192.168.203.13     0.0.0.0                  0         32768 i
*> 192.168.250.14     0.0.0.0                  0         32768 i
*>i192.168.250.15     192.168.250.2             0    100      0 i

RTA#
```

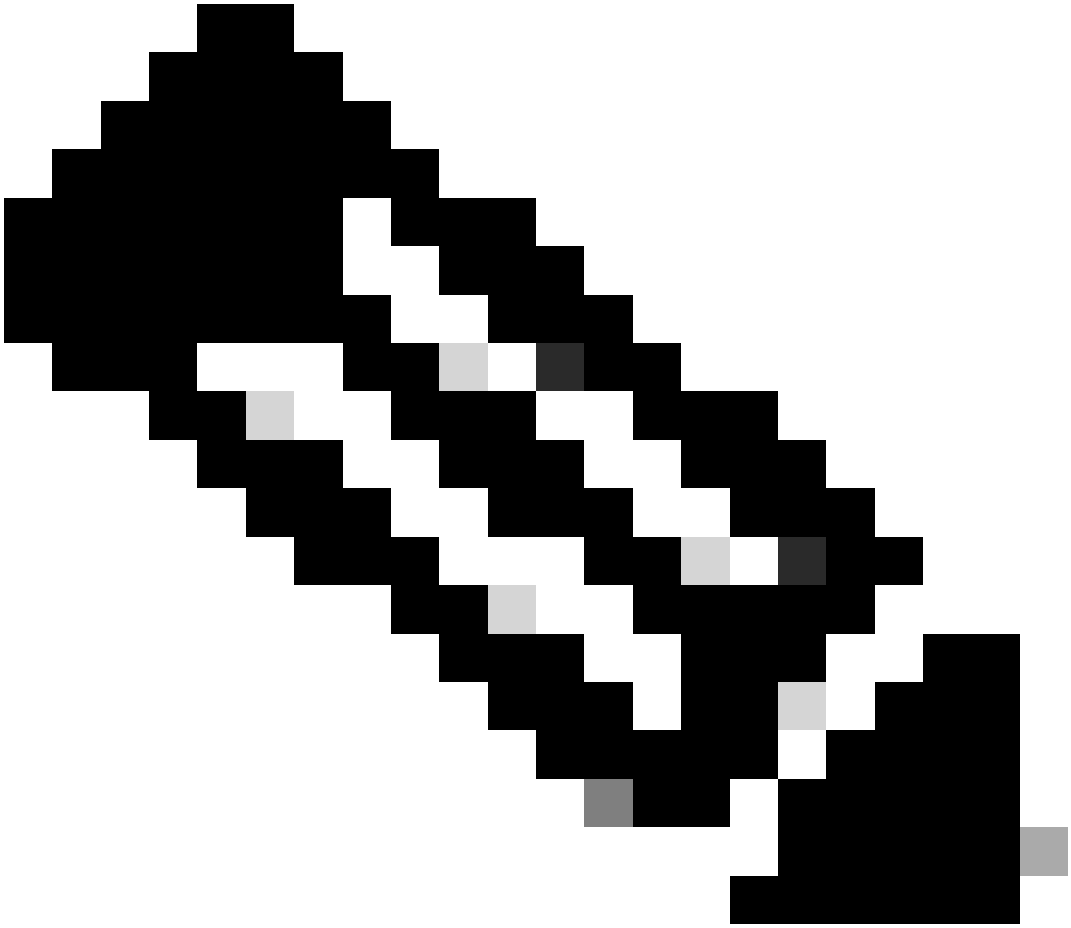
show ip route

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * -
candidate default

Gateway of last resort is 172.31.63.250 to network 172.31.200.200
```

```
     192.168.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    192.168.10.0 255.255.255.0
           [110/1000] via 172.31.14.250, 00:41:25, Ethernet0
O       192.168.10.4 255.255.255.252
           [110/138] via 172.31.14.250, 00:41:25, Ethernet0
C    192.168.203.13 is directly connected, Loopback0
     192.168.250.15 is variably subnetted, 3 subnets, 3 masks
O       172.16.15.2500 255.255.255.255
           [110/75] via 172.31.14.250, 00:41:25, Ethernet0
O       192.168.250.15 255.255.255.252
           [110/74] via 172.31.14.250, 00:41:25, Ethernet0
B       192.168.250.15 255.255.255.0 [200/0] via 192.168.250.2, 00:41:25
C    192.168.250.14 is directly connected, Ethernet0
     172.31.10.0 is variably subnetted, 2 subnets, 2 masks
B       172.31.10.0 255.255.0.0 [20/0] via 172.31.63.250, 00:41:26
C       192.168.213.63 255.255.255.252 is directly connected, Serial0
O*E2 0.0.0.0/0 [110/1000] via 172.31.14.250, Ethernet0/0
B*   172.31.200.200 255.255.0.0 [20/0] via 172.31.63.250, 00:02:38

RTF#
```

show ip route

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * -
candidate default

Gateway of last resort is 192.168.250.2 to network 0.0.0.0

     192.168.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    192.168.10.0 255.255.255.0
           [110/1000] via 192.168.250.2, 00:48:50, Serial1
O       192.168.10.4 255.255.255.252
           [110/128] via 192.168.250.2, 01:12:09, Serial1
     192.168.203.13 is variably subnetted, 2 subnets, 2 masks
O       192.168.203.250 255.255.255.255
           [110/11] via 192.168.203.14, 01:12:09, Ethernet0
O E2    192.168.203.13 255.255.255.0
           [110/2000] via 192.168.203.14, 01:12:09, Ethernet0
     192.168.250.15 is variably subnetted, 2 subnets, 2 masks
O       172.16.15.2500 255.255.255.255
           [110/65] via 192.168.250.2, 01:12:09, Serial1
C       192.168.250.15 255.255.255.252 is directly connected, Serial1
C    192.168.250.14 is directly connected, Ethernet0
     172.31.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    172.31.10.0 255.255.0.0
           [110/2000] via 192.168.203.14, 00:45:01, Ethernet0
O       192.168.213.63 255.255.255.252
           [110/74] via 192.168.203.14, 01:12:11, Ethernet0
O E2 172.31.200.200 255.255.0.0 [110/2000] via 192.168.203.14, 00:03:47, Ethernet0
O*E2 0.0.0.0 0.0.0.0 [110/1000] via 192.168.250.2, 00:03:33, Serial1
```



Remarque : La table de routage RTF indique que les réseaux locaux pour AS300, tels que 192.168.10.0, sont accessibles par RTB. Pour atteindre d'autres réseaux connus, tels que 172.31.200.200, il faut passer par RTA. La passerelle de dernier recours est définie à RTB. Si quelque chose arrive à la connexion entre RTB et RTD, la valeur par défaut que RTA annonce intervient avec une métrique de 2000.

<#root>

RTB#

show ip bgp

```
BGP table version is 14, local router ID is 172.16.15.2500
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*>i172.31.10.0      172.31.63.250           0    200      0 200 i
*> 192.168.10.0     192.168.10.5            0    300      0 300 i
*>i172.31.200.200/16   172.31.63.250               200      0 200 400 i
*>i192.168.203.13     192.168.203.250          0    100      0 i
*>i192.168.250.14     192.168.203.250          0    100      0 i
*> 192.168.250.15     0.0.0.0                  0         32768 i

RTB#
```

show ip route

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * -
candidate default

Gateway of last resort is 192.168.10.5 to network 192.168.10.0

 *   192.168.10.0 is variably subnetted, 2 subnets, 2 masks
B*      192.168.10.0 255.255.255.0 [20/0] via 192.168.10.5, 00:50:46
C       192.168.10.4 255.255.255.252 is directly connected, Serial1
     192.168.203.13 is variably subnetted, 2 subnets, 2 masks
O       192.168.203.250 255.255.255.255
           [110/75] via 172.16.15.250, 01:20:33, Serial0
O E2    192.168.203.13 255.255.255.0
           [110/2000] via 172.16.15.250, 01:15:40, Serial0
     192.168.250.15 255.255.255.252 is subnetted, 2 subnets
C       172.31.250.8 is directly connected, Loopback1
C       192.168.250.15 is directly connected, Serial0
O    192.168.250.14 [110/74] via 172.16.15.250, 01:20:33, Serial0
     172.31.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    172.31.10.0 255.255.0.0 [110/2000] via 172.16.15.250, 00:46:55, Serial0
O       192.168.213.63 255.255.255.252
           [110/138] via 172.16.15.250, 01:20:34, Serial0
O*E2 0.0.0.0/0 [110/2000] via 172.16.15.250, 00:08:33, Serial0
O E2 172.31.200.200 255.255.0.0 [110/2000] via 172.16.15.250, 00:05:42, Serial0
```

Informations connexes

- **BGP : Foire aux questions**
- **Exemples de configuration de BGP à travers un pare-feu PIX**
- **Comment utiliser HSRP pour assurer la redondance dans un réseau BGP multihébergé**
- **Configurer la redondance du mode routeur unique et le BGP sur un MSFC de Cat6000**
- **Optimisation du routage et réduction de la consommation de mémoire au niveau des routeurs BGP**
- **Dépannage des problèmes courants du protocole BGP**
- **Dépannage d'un CPU élevé causé par l'analyseur BGP ou le processus du routeur**
- **Comprendre le partage de charge avec le BGP dans les environnements à un ou plusieurs réseaux**
- **Page de support BGP**
- **Assistance technique de Cisco et téléchargements**

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.