Études de cas BGP

Contenu

Introduction

Ce document contient cinq études de cas sur Border Gateway Protocol (BGP).

Conditions préalables

Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

Composants utilisés

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

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.

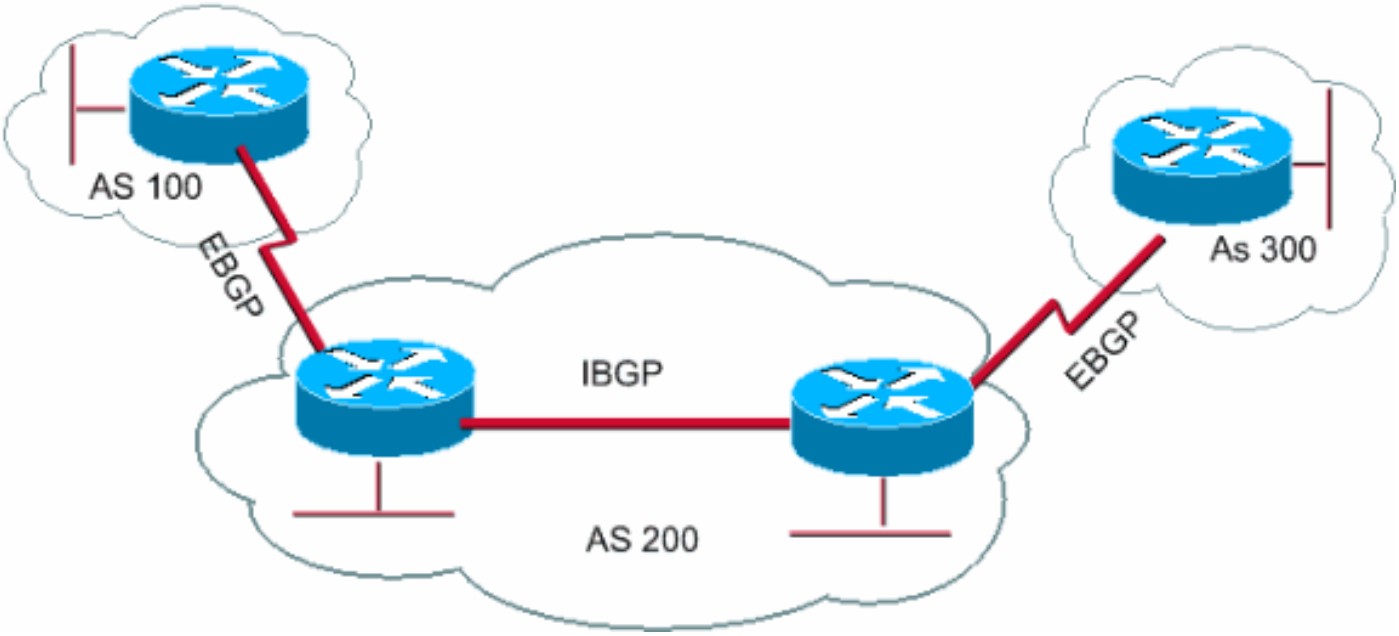
eBGP et iBGP

Si un AS comporte plusieurs speakers BGP, il peut servir de service de transit pour d'autres AS. Comme le montre le diagramme de 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.



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.

- 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 :

```
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.

Définissez les voisins BGP.

La formation de voisins BGP indique les routeurs qui essaient de communiquer via BGP. La section Former des voisins BGP 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 commande **neighbor** pour établir une connexion TCP :

```
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 commande **neighbor** des routeurs homologues *doivent* pouvoir accéder l'une à l'autre. Une manière de vérifier l'accessibilité consiste à faire un test ping étendu entre les deux adresses IP. Le test ping étendu force le routeur sur lequel le test ping porte à utiliser comme source l'adresse IP que la commande **neighbor** spécifie. 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 adresse

Remarque: ***adresse*** est l'adresse du voisin.

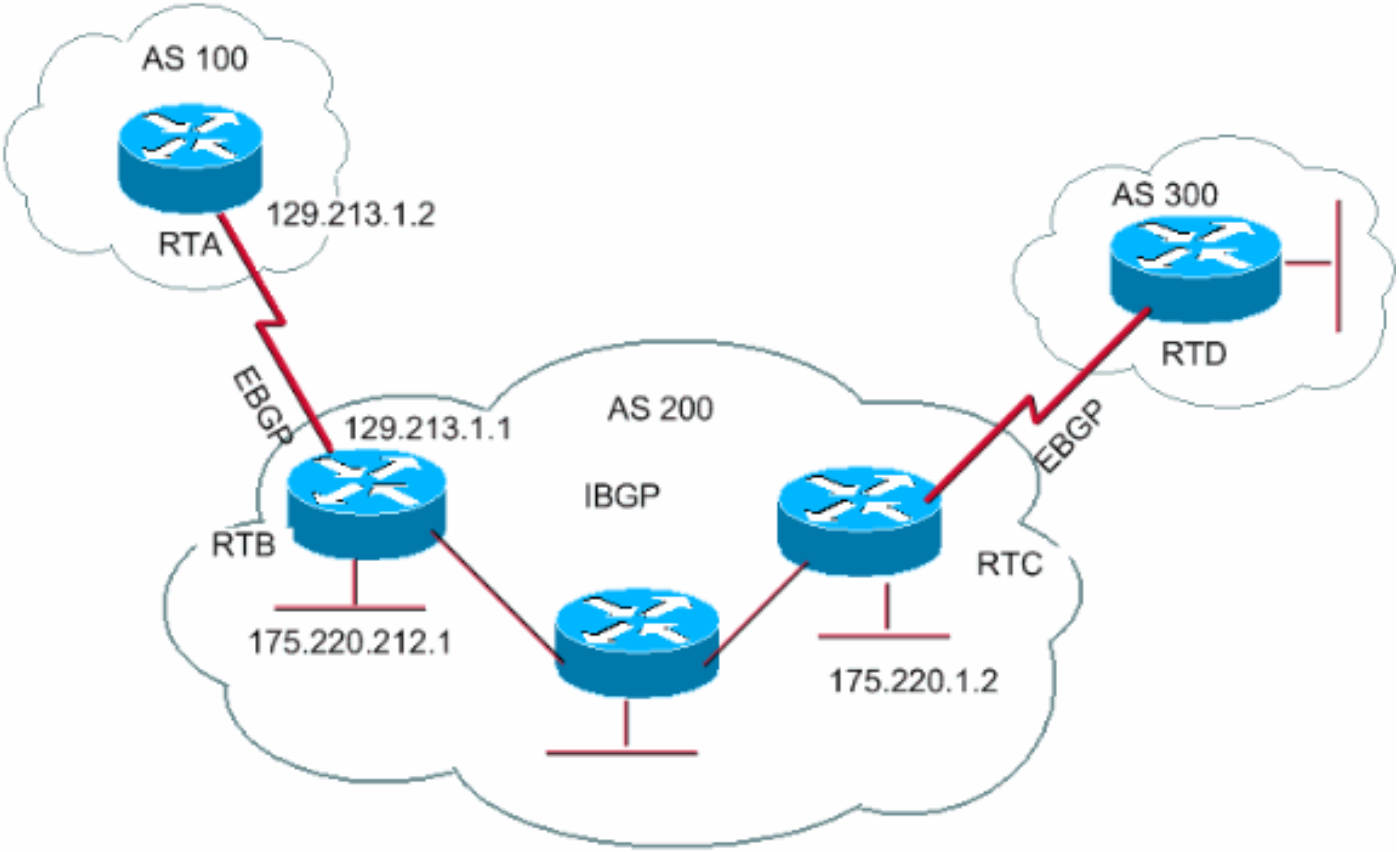
clear ip bgp *

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 :

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

Voici un exemple de configuration de la commande **neighbor** :



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

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. En outre, les homologues eBGP ont une connexion directe, mais les homologues iBGP n'en ont pas. Les routeurs iBGP n'ont pas besoin de connexion directe. Toutefois, il doit exister un IGP qui s'exécute et permet aux deux voisins de communiquer.

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

Remarque: Faites particulièrement attention à l'état de BGP. Une valeur autre que l'état Established indique que les homologues ne sont pas actifs.

Remarque: En outre, notez ce qui suit :

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.

```
# show ip bgp neighbors
     BGP neighbor is 129.213.1.1, remote AS 200, external link
     BGP version 4, remote router ID 175.220.12.1
     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
```

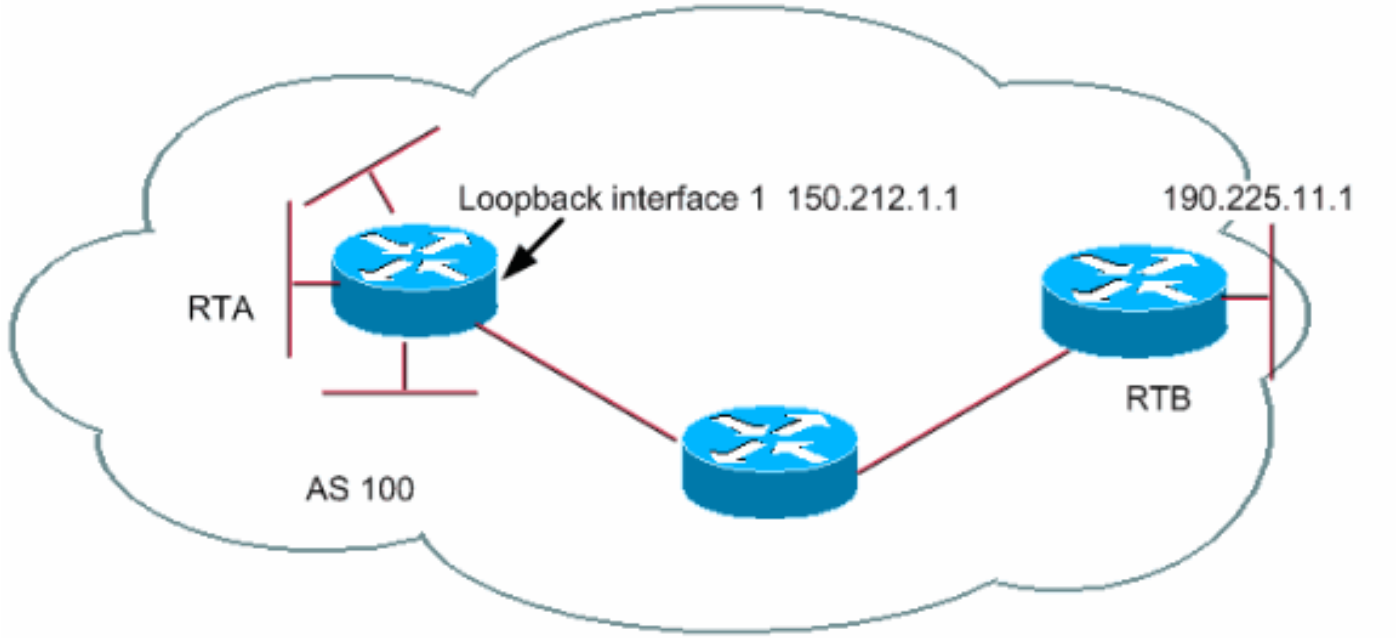
BGP et interfaces de bouclage

L'utilisation d'une interface de bouclage pour définir les voisins est commune 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 commande **neighbor**, 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 :

```
neighbor ip-address update-source interface
```

Cet exemple illustre l'utilisation de cette commande :



neighbor *ip-address* **update-source** *interface*

Dans cet exemple, RTA et RTB exécutent iBGP dans AS100. Dans la commande **neighbor**, RTB utilise l'interface de bouclage de RTA, 150.212.1.1. 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** de sorte que la commande soit **neighbor 190.225.11.1 update-source loopback 1.** Cette instruction force BGP à utiliser l'adresse IP de l'interface de bouclage quand BGP communique avec neighbor 190.225.11.1.

Remarque: RTA a utilisé l'adresse IP de l'interface physique de RTB, 190.225.11.1, comme 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

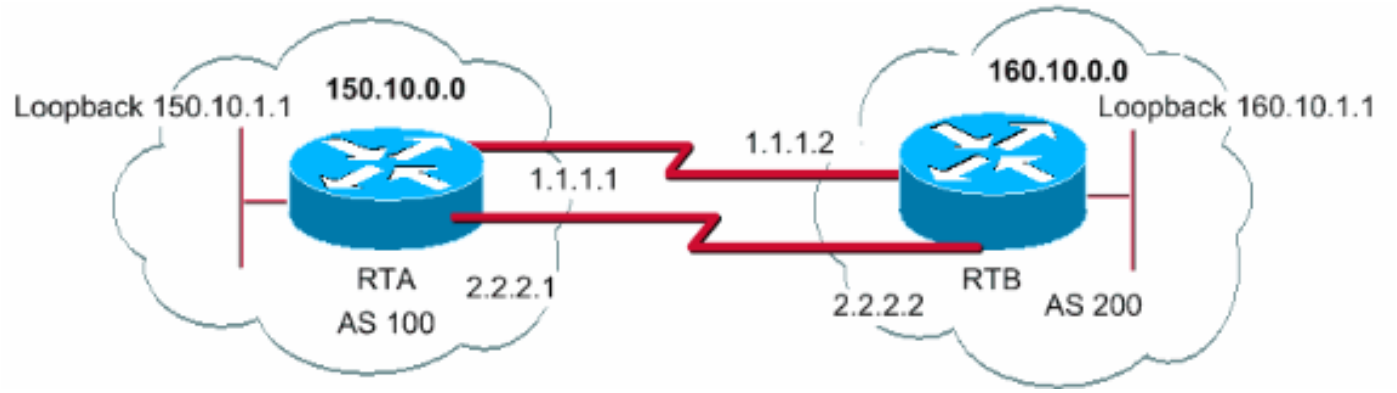
Dans certains cas, un routeur Cisco peut exécuter eBGP avec un routeur tiers qui ne permet pas la connexion directe des deux homologues externes. Pour établir la connexion, vous pouvez utiliser le saut multiple eBGP. Le saut multiple eBGP permet d'établir une connexion de voisinage entre deux homologues 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 :

neighbor *ip-address* **update-source** *interface*

RTA indique un voisin externe qui n'a pas de connexion directe. RTA doit indiquer son utilisation de la commande de **neighbor ebgp-multihop**. D'autre part, RTB indique un voisin avec une connexion directe, qui est 129.213.1.2. En raison de cette liaison directe, RTB n'a pas besoin de la commande de **neighbor ebgp-multihop**. Vous devez également configurer un routage IGP ou statique pour permettre aux voisins sans connexion de communiquer.

L'exemple de la section Saut multiple eBGP (équilibrage de charge) montre comment réaliser l'équilibrage de charge avec BGP en cas d'utilisation d'eBGP sur des lignes parallèles.

Saut multiple eBGP (équilibrage de charge)



neighbor *ip-address* **update-source** *interface*

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. RTA a deux possibilités pour atteindre le prochain saut 160.10.1.1 : un chemin par l'intermédiaire de 1.1.1.2 et l'autre chemin par l'intermédiaire de 2.2.2.2. 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. Le format d'une mise en correspondance de route est le suivant :

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

Chacune mise en correspondance de route se compose d'une liste de commandes de configuration **match** et **set**. match spécifie un **critère de correspondance**, et set spécifie une action **set** si les critères que la commande **match** impose sont respectés.

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 1.1.1.1, la métrique de cette mise à jour est définie sur 5. Les commandes suivantes illustrent l'exemple :

```
match ip address 1.1.1.1
set metric 5
```

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

Si les critères de correspondance sont respectés et que avez un **refus**, il n'y a pas de redistribution ou de contrôle de la route. Vous sortez de la liste.

Si les critères de correspondance ne sont pas remplis et que vous avez une **autorisation ou un refus**, l'instance suivante de la mise en correspondance de route 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 trouver de correspondance, la route n'est **ni acceptée ni transférée**.

Dans les versions du logiciel Cisco IOS® antérieures à la version 11.2, si vous utilisez des mises en correspondance de route pour filtrer les mises à jour BGP au lieu de les redistribuer entre les protocoles, vous *ne pouvez pas* filtrer sur les données entrantes lorsque 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 relatives à **match** sont :

match as-path

match community

match clns

match interface

match ip address

match ip next-hop

match ip route-source

match metric

match route-type

match tag

Les commandes relatives à **set** sont :

set as-path

set clns

set automatic-tag

set community

set interface

set default interface

set ip default next-hop

set level

set local-preference

set metric

set metric-type

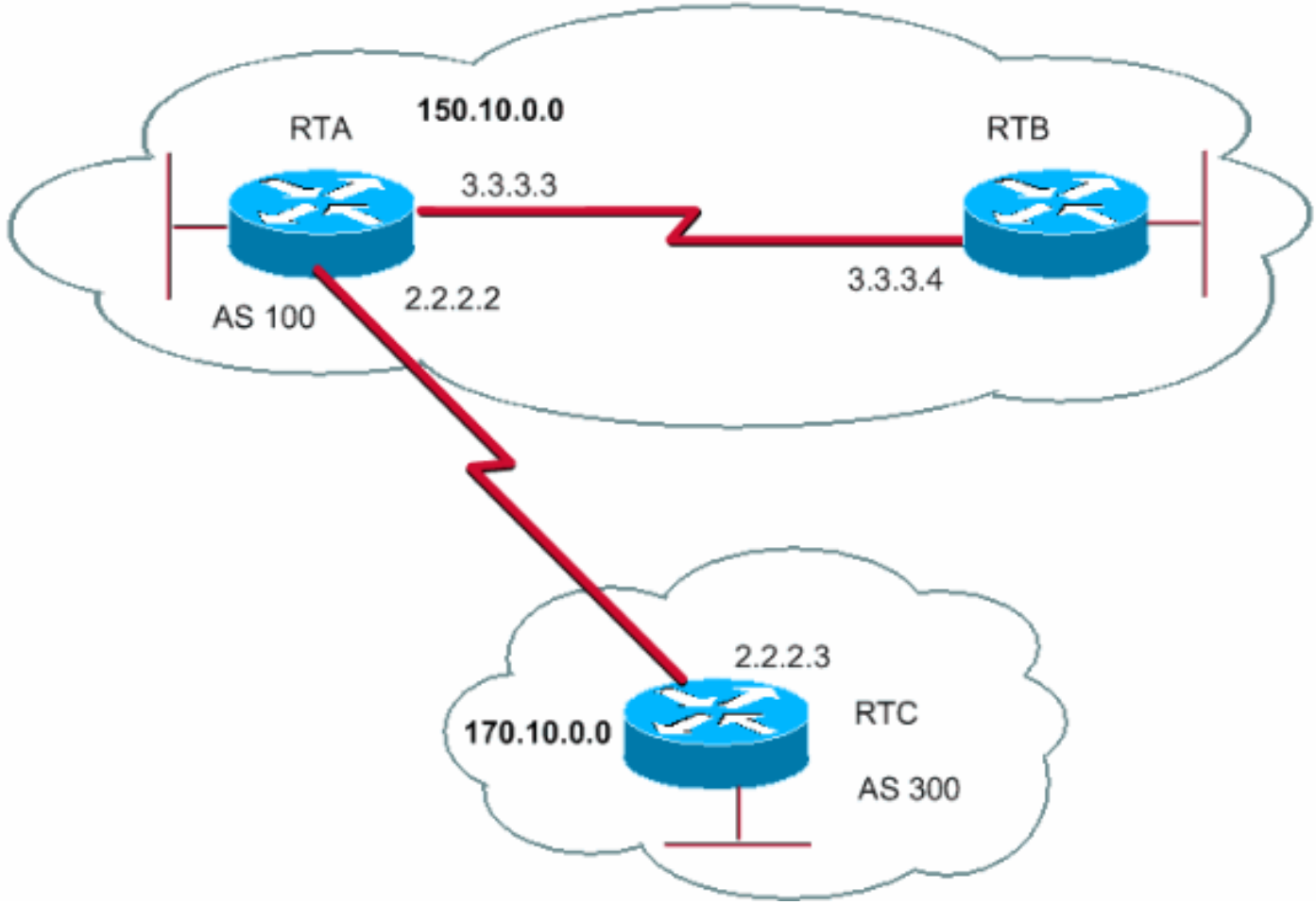
set next-hop

set origin

set tag

set weight

Voici quelques exemples de mises en correspondance de route :



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. Supposez que RTA veut redistribuer aux artères RTB au sujet de 170.10.0.0 avec une mesure de 2 et à toutes autres artères avec une mesure de 5. dans ce cas, vous pouvez utiliser cette configuration :

```
match ip address 1.1.1.1
set metric 5
```

Dans cet exemple, si une route correspond à l'adresse IP 170.10.0.0, elle a une métrique de 2. Ensuite, vous sortez de la liste des mises en correspondance de route. En l'absence de correspondance, vous continuez à parcourir vers le bas la liste des mises en correspondance de route, qui indique de définir une métrique de 5 pour tout le reste.

Remarque: Posez toujours la question « Que se passe-t-il pour les routes qui ne correspondent à aucune des instructions de correspondance ? » Ces routes sont ignorées par défaut.

Exemple 2

Supposons que, dans l'exemple 1, vous ne voulez pas qu'AS100 accepte les mises à jour relatives à 170.10.0.0. 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 :

```
match ip address 1.1.1.1
set metric 5
```

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 commande **network** est :

```
network network-number [mask network-mask]
```

La commande **network** contrôle les réseaux qui proviennent de ce routeur. 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. Au lieu de cela, vous essayez d'indiquer à BGP quels réseaux BGP devraient provenir de ce routeur. 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 pour la commande **network** est acceptable.

La commande **network** 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** :

```
network network-number [mask network-mask]
```

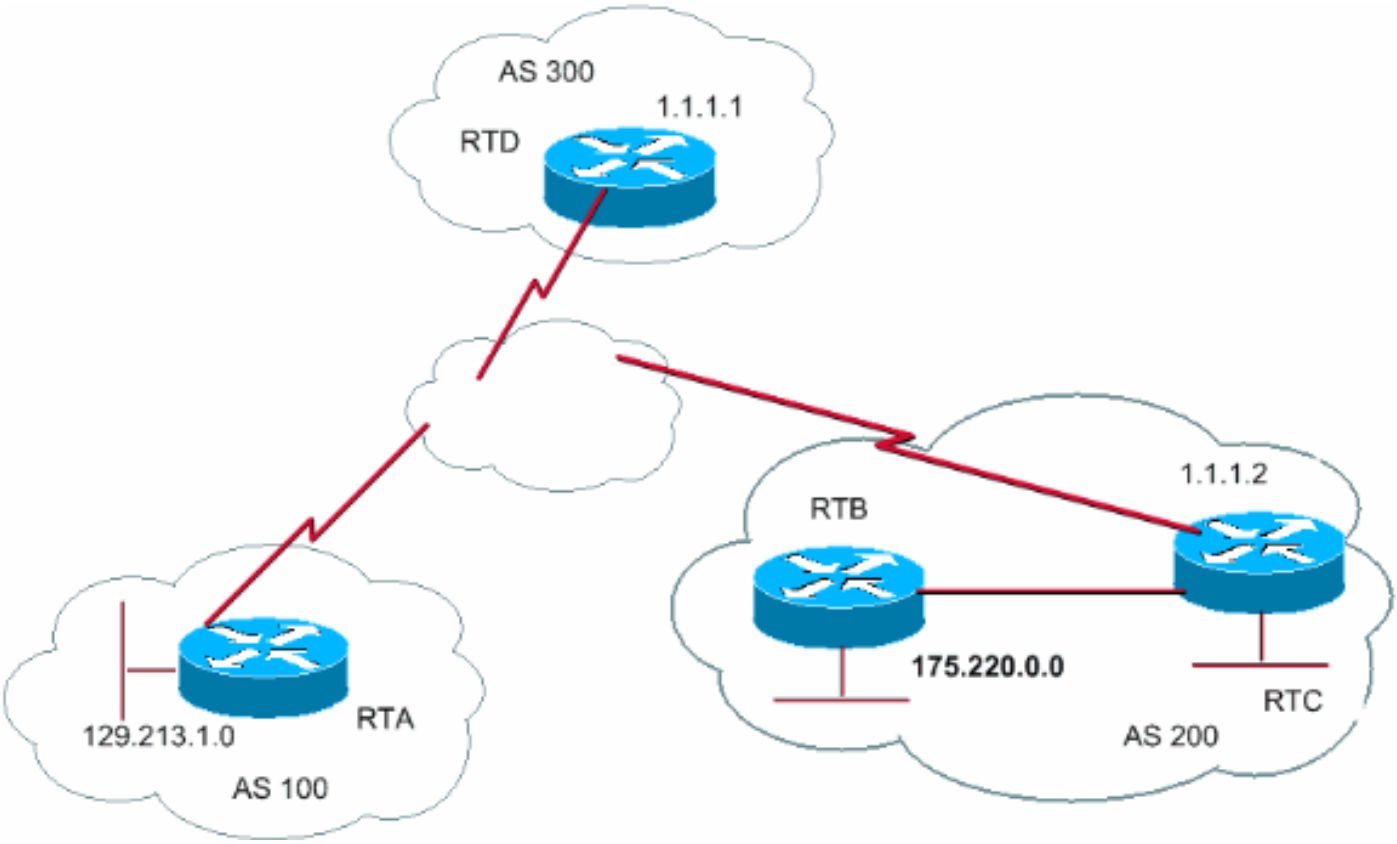
Cet exemple montre que le routeur A génère une entrée réseau pour 192.213.0.0/16. /16 indique que vous utilisez un super-réseau d'adresse de classe C et que vous annoncez les deux premiers octets, ou les 16 premiers bits.

Remarque: Vous avez besoin de la route statique pour que le routeur génère 192.213.0.0 étant donné que la route statique place une entrée correspondante dans la table de routage.

Redistribution

La commande **network** est l'une des méthodes permettant 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 effrayante parce que videz toutes vos routes internes dans BGP ; certaines de ces routes peuvent avoir été apprises via BGP et vous n'avez pas besoin de les renvoyer. Soyez prudent lorsque vous appliquez un filtre pour être sûr d'envoyer aux routes Internet que vous voulez annoncer et pas à toutes les routes dont vous disposez. Voici un exemple :

RTA annonce 129.213.1.0 et RTC annonce 175.220.0.0. Regardez la configuration RTC :



Si vous émettez la commande **network**, vous obtenez :

```
network network-number [mask network-mask]
```

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

```
network network-number [mask network-mask]
```

Cette redistribution entraîne la création de 129.213.1.0 par votre AS. Vous n'êtes pas la source de 129.213.1.0 ; AS100 est la source. Par conséquent, vous devez utiliser des filtres pour empêcher la sortie de la source de ce réseau par votre AS. La configuration correcte est la suivante :

```
network network-number [mask network-mask]
```

Vous utilisez la commande **access-list** 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 émission de **redistribute ospf 1** sous **router bgp** ne fonctionne pas. Des mots clés spécifiques tels qu'**internal**, **external** et **nssa-external** sont nécessaires pour redistribuer les routes respectives. Référez-vous à Présentation de la redistribution des routes OSPF dans BGP pour plus de détails.

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. Vous pouvez obtenir le même résultat que l'exemple de la section Redistribution avec ceci :

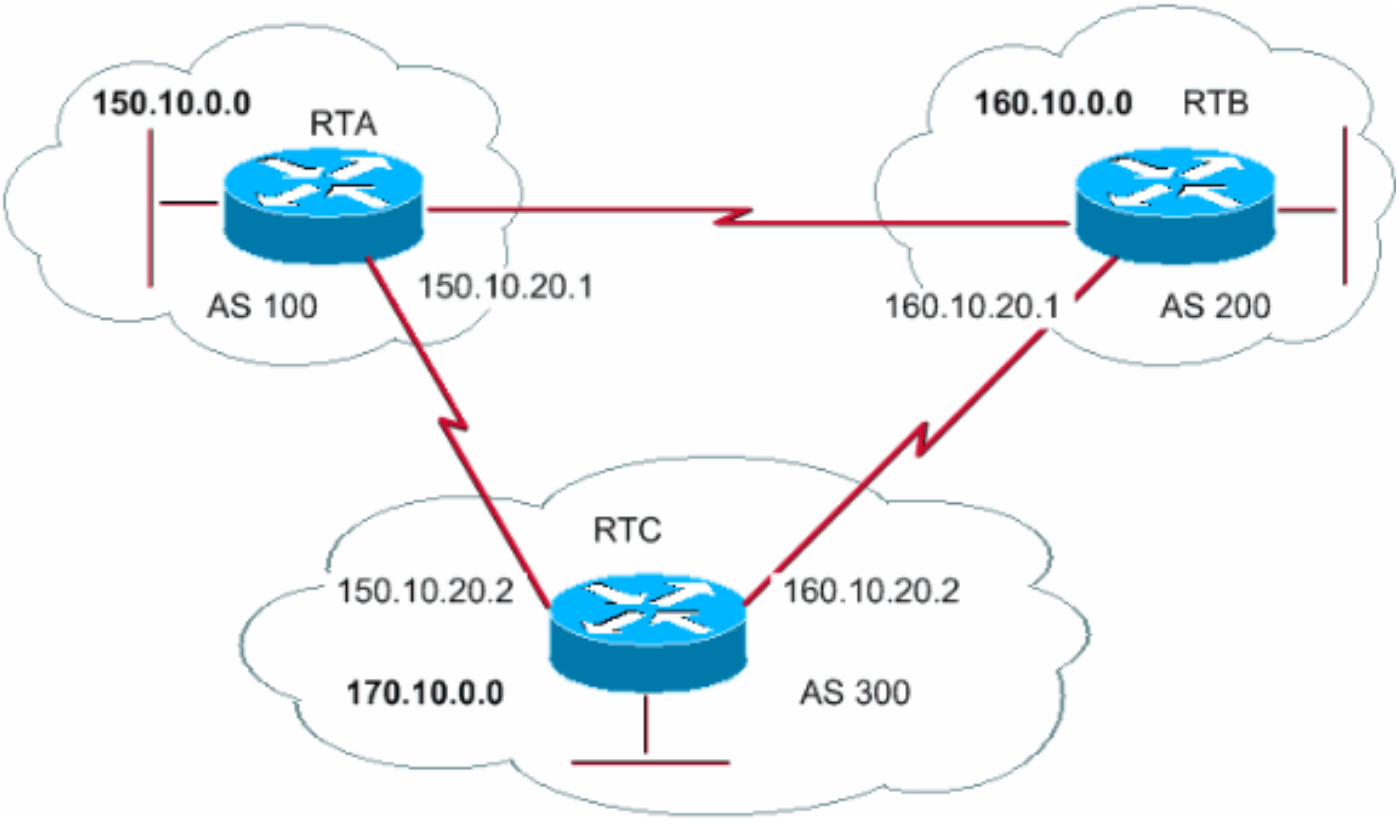
```
network network-number [mask network-mask]
```

L'interface **null0** signifie que le paquet est ignoré. Par conséquent, si vous obtenez le paquet et qu'il y a une correspondance plus spécifique que 175.220.0.0 (qui existe), le routeur envoie le paquet à la correspondance spécifique. 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 générées par la commande **network**, la redistribution, ou les routes statiques indiquent votre AS en tant qu'origine de ces réseaux.

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

Voici un exemple :



network *network-number* **[mask** *network-mask***]**

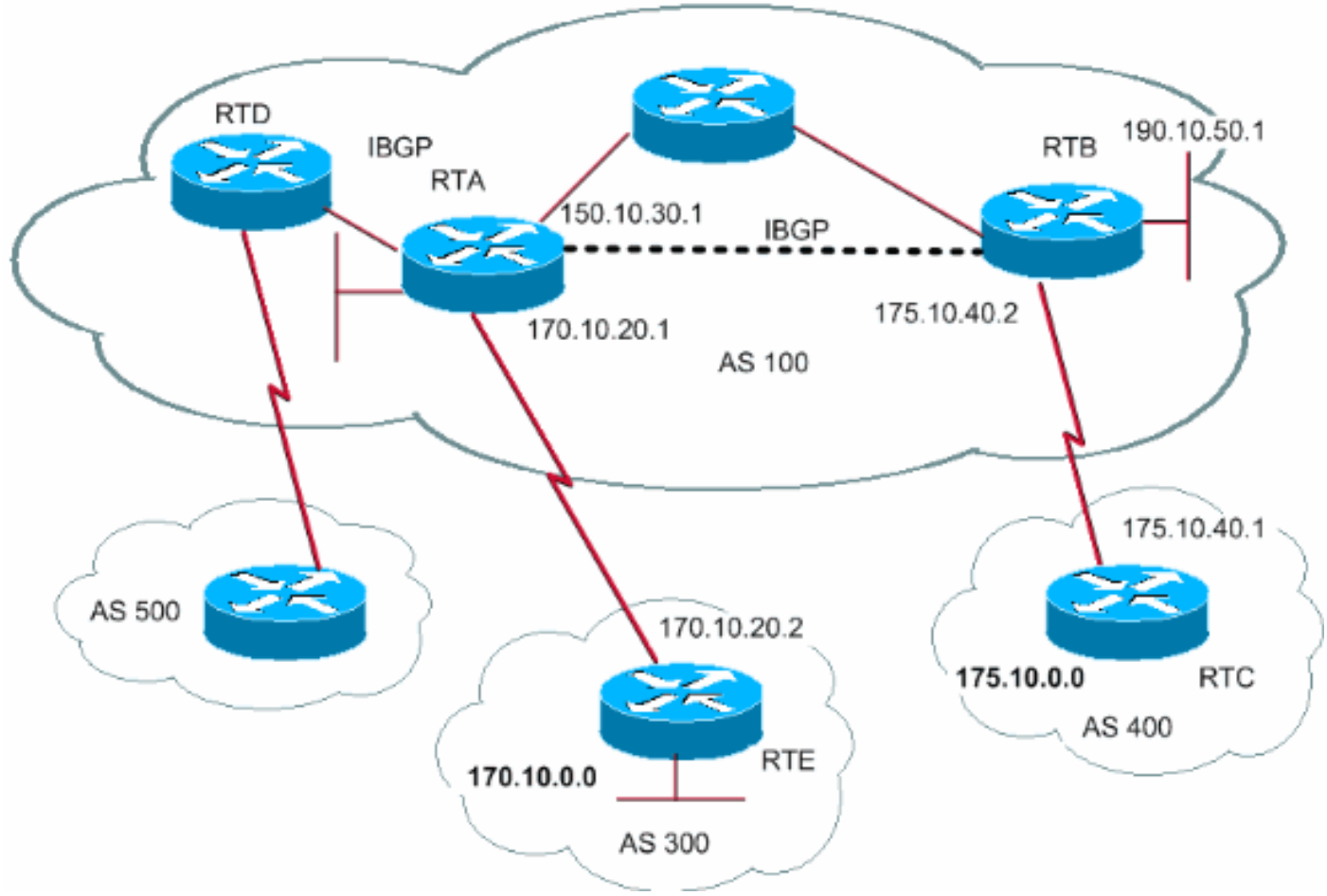
Remarque: Vous n'avez pas besoin du réseau 150.10.0.0 ou du réseau 160.10.0.0 dans RTC, à moins que vous vouliez que RTC génère ces réseaux et leur transmette les informations lorsqu'elles arrivent d'AS100 et AS200. De nouveau, la différence est que la commande **network** ajoute une annonce supplémentaire 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 ont pour origine son propre 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 150.10.0.0 et l'envoie à AS300. Ensuite, RTC passe cette route à AS200 et conserve son origine comme étant AS100. RTB passe 150.10.0.0 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 l'iBGP lorsqu'un AS souhaite agir en tant que système de transit vers d'autres AS. Est-il vrai que vous pouvez faire la même chose en apprenant par l'intermédiaire d'eBGP, en redistribuant dans IGP, puis en redistribuant vers un autre AS ? Oui, mais iBGP offre plus de souplesse et des méthodes plus efficaces pour échanger des informations 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 Attribut local preference fournit plus d'informations sur la préférence locale.



network *network-number* **[mask** *network-mask***]**

Remarque: Rappelez-vous que lorsqu'un speaker BGP reçoit une mise à jour d'autres speakers BGP de son propre AS (iBGP), le speaker BGP qui reçoit la mise à jour ne redistribue pas cette information aux autres speakers BGP situés dans son propre 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.

Dans le diagramme de cette section, RTA et 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.

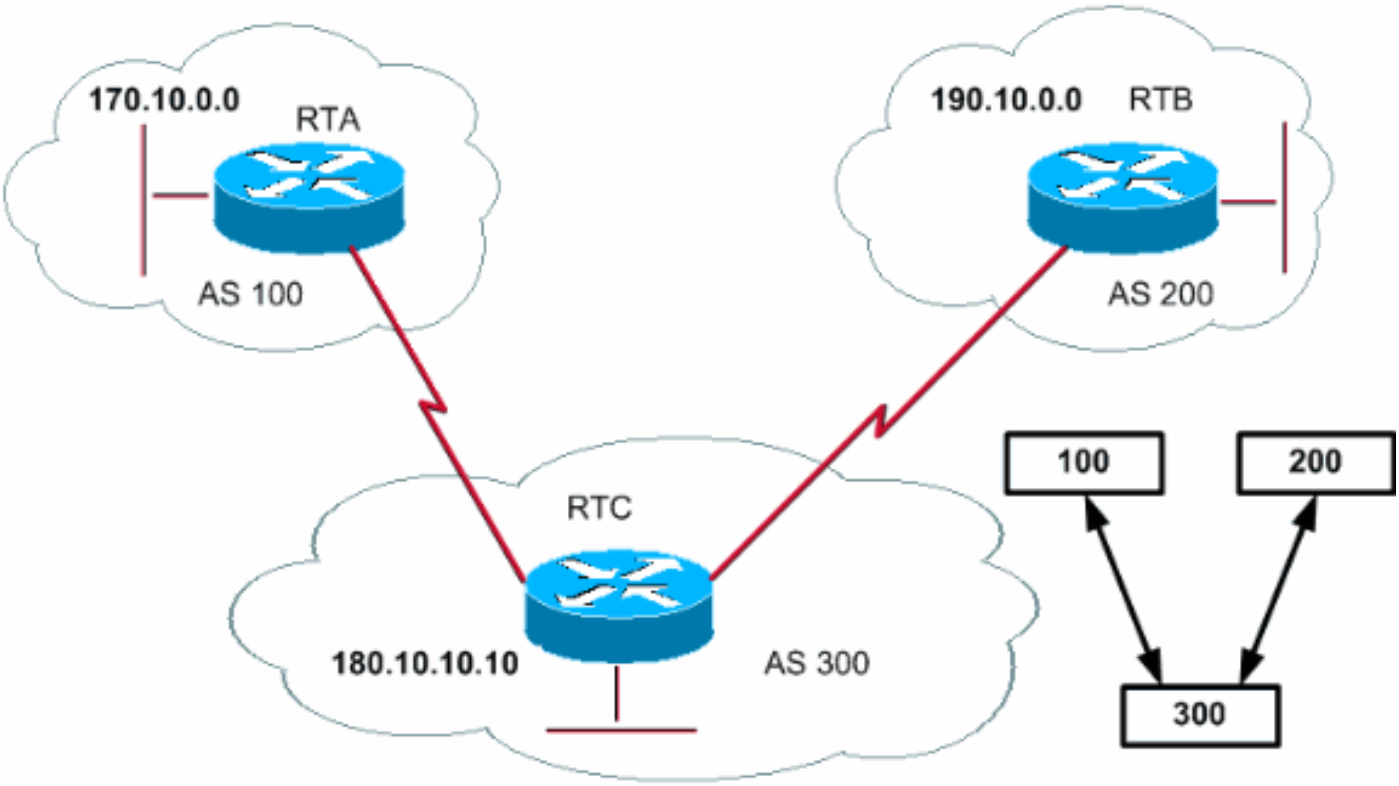
BGP base sa décision sur différents **attributs**, tels que le prochain saut, les poids administratifs, la préférence locale, l'origine de la route, la longueur du chemin d'accès, le code d'origine, la métrique, et d'autres attributs.

BGP propage toujours le meilleur chemin aux voisins. Reportez-vous à l'Algorithme de sélection du meilleur chemin BGP pour plus d'informations.

La section Études de cas BGP 2 explique ces attributs et leur utilisation.

Études de cas BGP 2

Attribut AS_PATH



Àchaque 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 Exemple CIDR 2 (as-set) de ce document offre un exemple d'AS_SET.

Dans l'exemple de cette section, RTB annonce le réseau 190.10.0.0 dans AS200. Quand cette route traverse AS300, RTC ajoute son propre numéro d'AS au réseau. Par conséquent, quand 190.10.0.0 atteint RTA, deux numéros d'AS sont attachés au réseau : 200, puis 300. Pour RTA, le chemin pour atteindre 190.10.0.0 est (300, 200).

Le même processus s'applique à 170.10.0.0 et à 180.10.0.0. RTB doit prendre le chemin (300, 100) ; RTB traverse AS300 puis AS100 pour atteindre 170.10.0.0. RTC doit traverser le chemin (200) afin d'atteindre 190.10.0.0 et le chemin (100) afin d'atteindre 170.10.0.0.

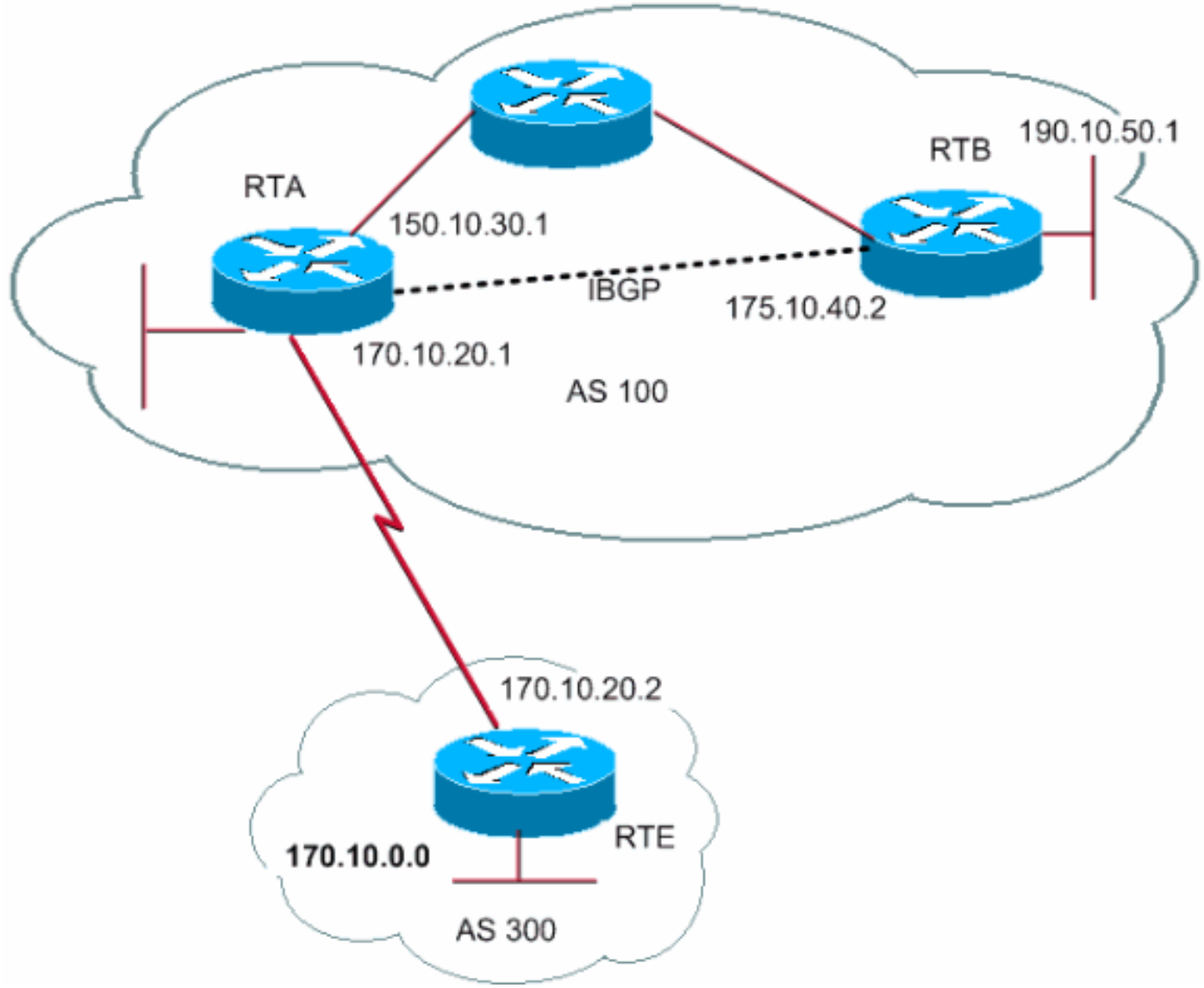
Attribut origin

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 intérieures au en date des origines. Ceci se produit normalement quand vous émettez la commande **bgp network**. Un i dans la table BGP indique IGP.

EGP : les informations NLRI sont apprises par l'intermédiaire de l'Exterior Gateway Protocol (EGP). Un e dans la table BGP indique EGP.

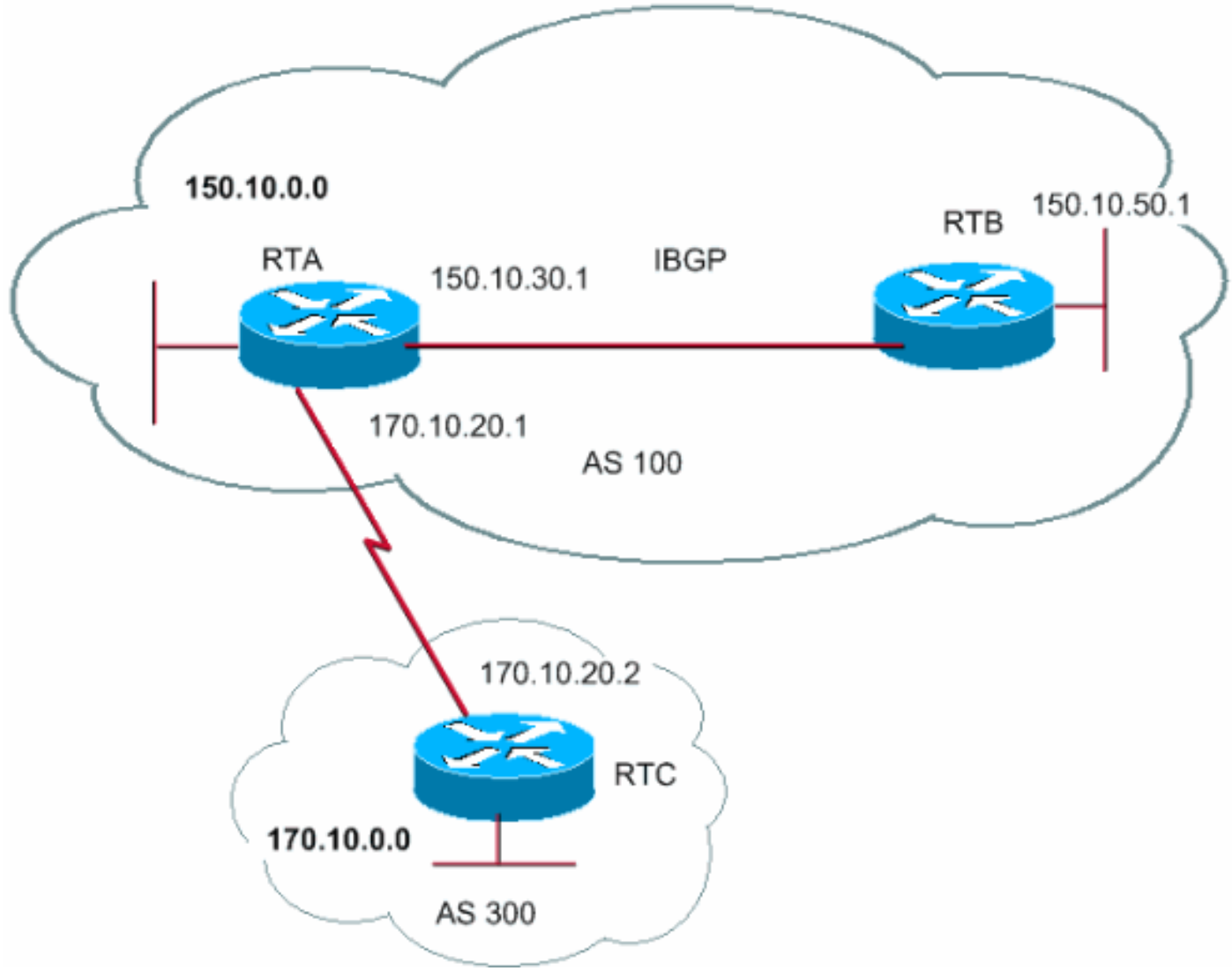
INCOMPLETE : les informations NLRI sont inconnues ou apprises par un autre moyen. INCOMPLETE se produit habituellement quand vous redistribuez les routes d'autres protocoles de routage dans BGP et que l'origine de la route est incomplète. ? dans la table BGP indique INCOMPLETE.



network *network-number* **[mask** *network-mask***]**

RTA atteint 170.10.0.0 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 190.10.50.0 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 150.10.0.0 par l'intermédiaire de 100 i. « 100 i » signifie que l'AS suivant est 100 et que l'origine est IGP. RTE atteint également 190.10.0.0 par l'intermédiaire de 100 ?. « 100 ? » signifie que l'AS suivant est 100 et que l'origine est incomplète et provient d'une route statique.

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 prochain saut est toujours l'adresse IP du voisin spécifié par la commande **neighbor**. Dans l'exemple de cette section, RTC annonce 170.10.0.0 à RTA avec un prochain saut de 170.10.20.2. RTA annonce 150.10.0.0 à RTC avec un prochain saut de 170.10.20.1. Pour iBGP, le protocole indique que le prochain saut annoncé par eBGP doit être effectué vers iBGP. En raison de cette règle, RTA annonce 170.10.0.0 à son homologue iBGP RTB avec un prochain saut de 170.10.20.2. Toujours selon RTB, le prochain saut pour atteindre 170.10.0.0 est 170.10.20.2 et *pas* 150.10.30.1.

Assurez-vous que RTB peut atteindre 170.10.20.2 par l'intermédiaire d'IGP. Sinon, RTB ignore les paquets avec la destination 170.10.0.0 parce que l'adresse du prochain saut est inaccessible. Par exemple, si RTB exécute iGRP, vous pouvez également exécuter iGRP sur le réseau RTA 170.10.0.0. Vous voulez rendre iGRP passif sur le lien à RTC de sorte que BGP soit seulement échangé.

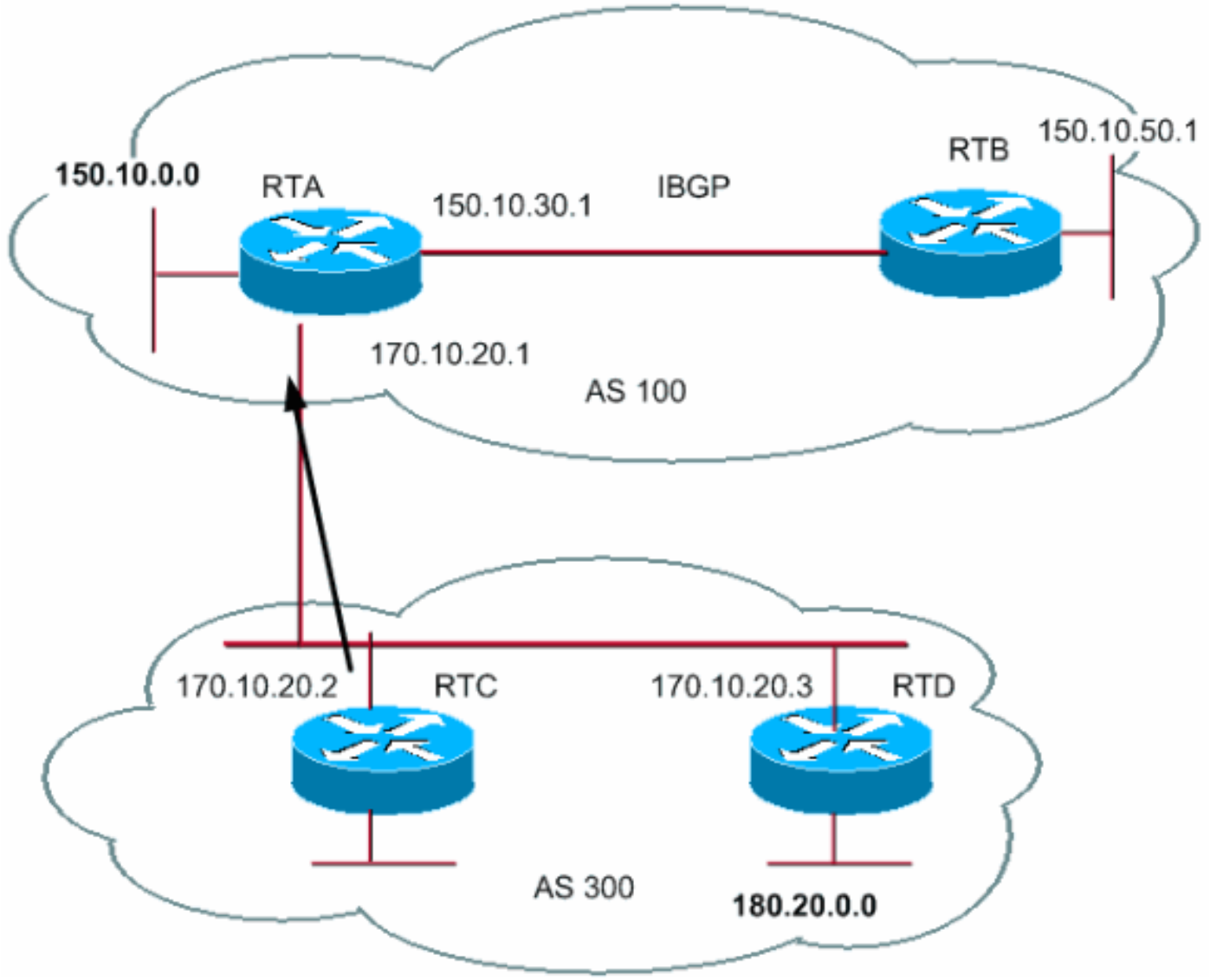
```
network network-number [mask network-mask]
```

Remarque: RTC annonce 170.10.0.0 à RTA avec un prochain saut égal à 170.10.20.2.

Remarque: RTA annonce 170.10.0.0 à RTB avec un prochain saut égal à 170.10.20.2. Le prochain saut eBGP est effectué dans iBGP.

Soyez particulièrement prudent quand vous gérez des réseaux multi-accès et NBMA (Non-Broadcast Multi-Access). Les sections Prochain saut BGP (réseaux multi-accès) et Prochain saut BGP (NBMA) fournissent plus de détails.

Prochain saut BGP (réseaux multi-accès)



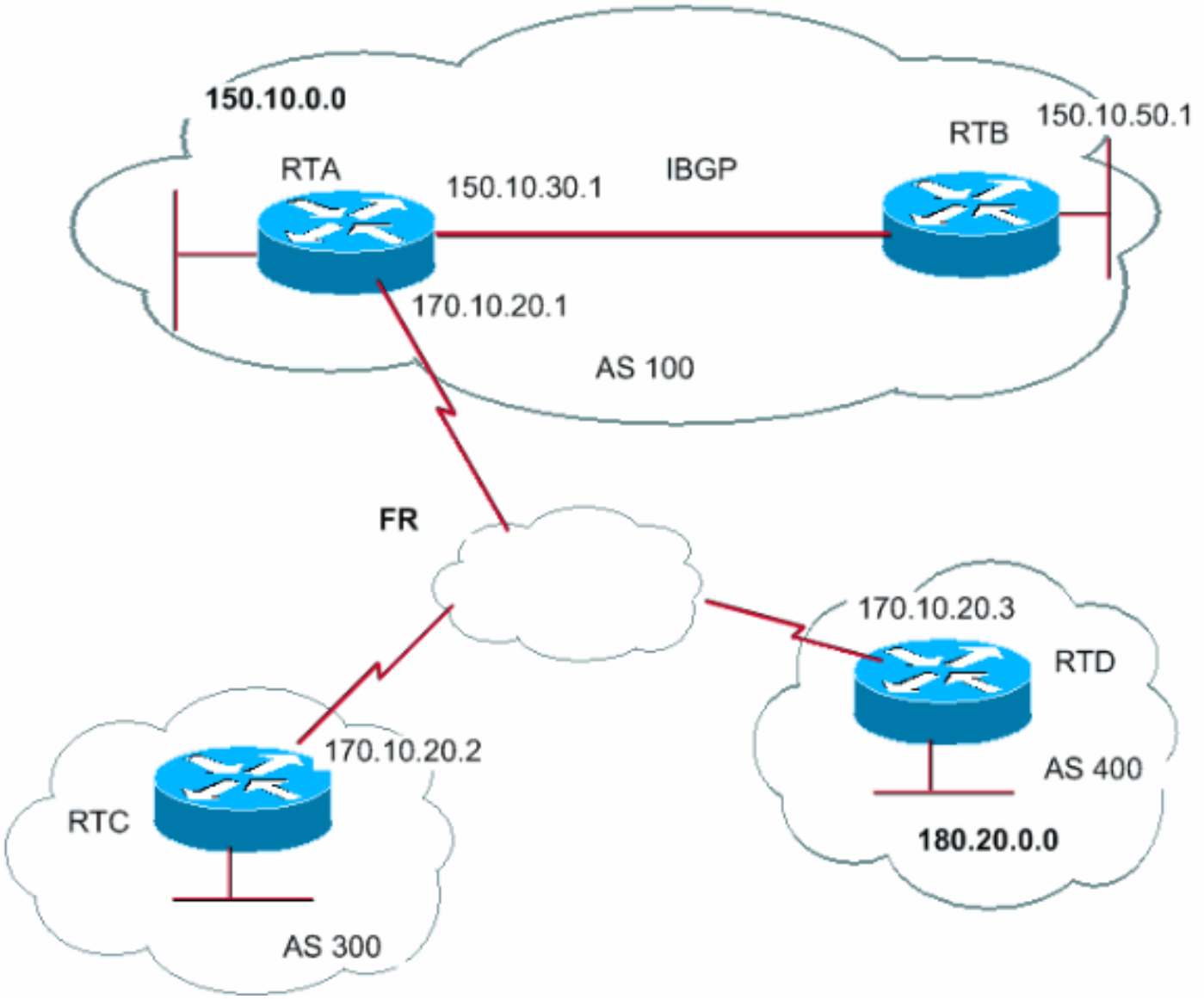
Cet exemple montre comment le prochain saut se comporte sur un réseau multi-accès tel qu'Ethernet.

Supposons que RTC et RTD dans AS300 exécutent OSPF. RTC exécute BGP avec RTA. RTC peut atteindre le réseau 180.20.0.0 par l'intermédiaire de 170.10.20.3. Quand RTC envoie une mise à jour BGP à RTA concernant 180.20.0.0, RTC utilise comme prochain saut 170.10.20.3. RTC n'utilise pas sa propre adresse IP, 170.10.20.2. 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 180.20.0.0 est plus raisonnable que le saut supplémentaire par l'intermédiaire de RTC.

Remarque: RTC annonce 180.20.0.0 à RTA avec un prochain saut de 170.10.20.3.

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 180.20.0.0 à RTA avec un prochain saut de 170.10.20.3.

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 commande **next-hop-self** remédie à cette situation.

Commande next-hop-self

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

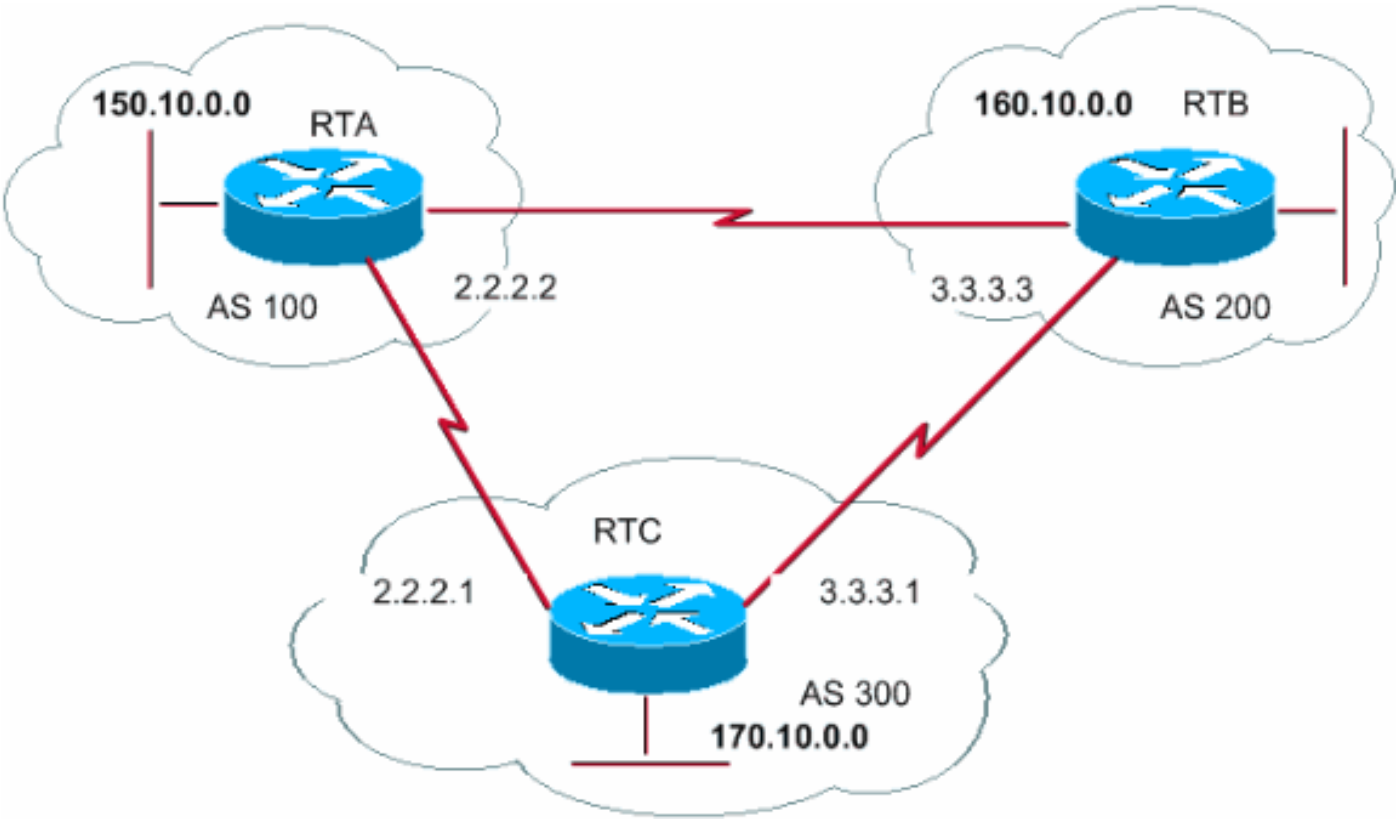
```
neighbor {ip-address | peer-group-name} next-hop-self
```

La commande **next-hop-self** vous permet de forcer BGP à utiliser une adresse IP spécifique comme prochain saut.

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

neighbor {*ip-address* | *peer-group-name*} **next-hop-self**
RTA annonce 180.20.0.0 avec un prochain saut égal à 170.10.20.2.

Porte dérobée BGP



Dans ce diagramme, 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 160.10.0.0 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 commande **distance** pour changer les distances par défaut :

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 160.10.0.0 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.

Émettez la commande network address backdoor.

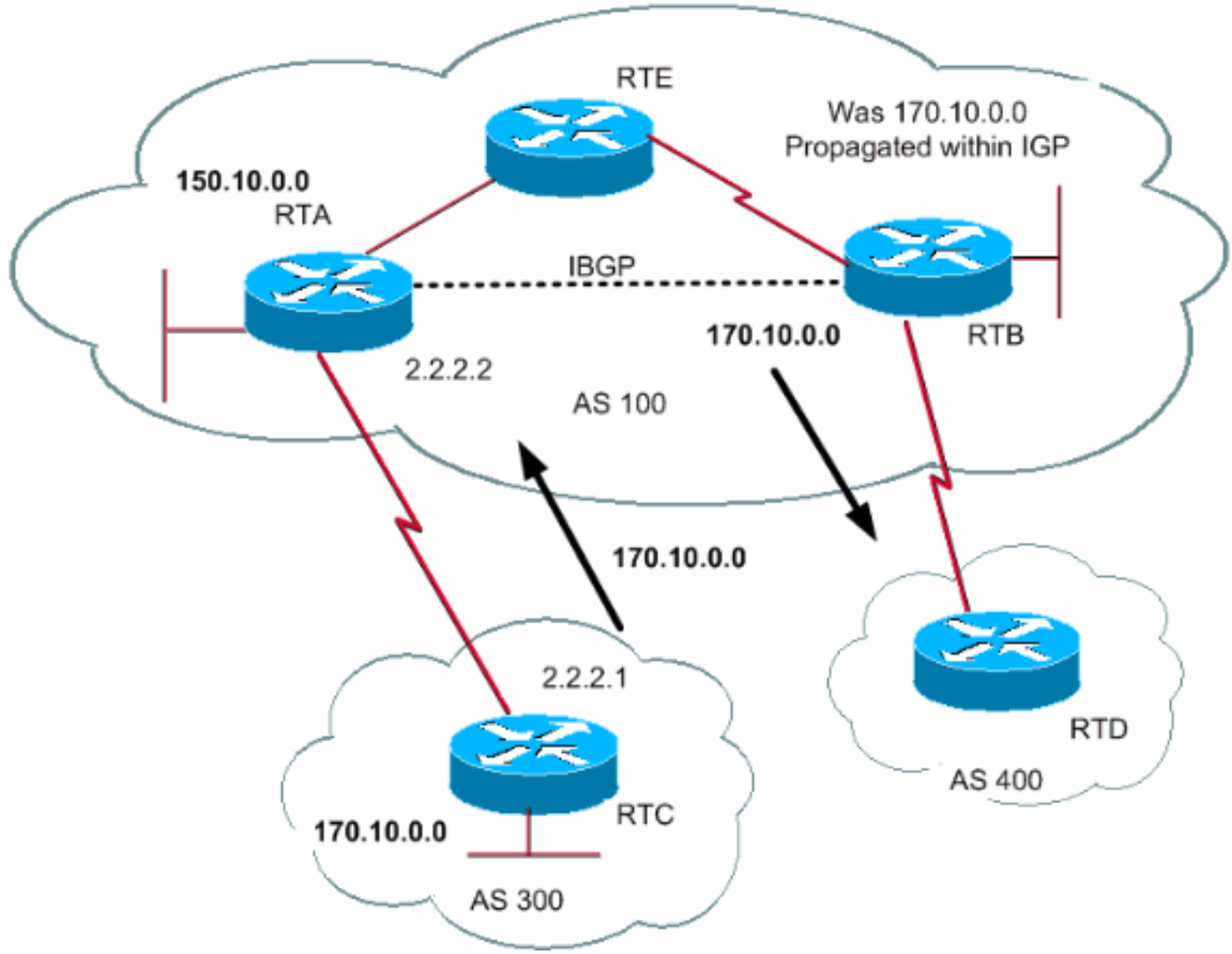
Le réseau configuré est le réseau qui vous voulez atteindre par l'intermédiaire d'IGP. Pour BGP, ce réseau bénéficie du même traitement qu'un réseau assigné localement, à ceci près que les mises à jour BGP n'annoncent pas ce réseau.

distance bgp *external-distance internal-distance local-distance*

Le réseau 160.10.0.0 est traité comme une entrée locale, mais n'est pas annoncé comme une entrée réseau normale.

RTA apprend 160.10.0.0 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 l'eBGP est la préférence, mais en raison de la commande de **network backdoor**, l'EIGRP est la préférence.

Synchronisation



Avant la discussion sur la synchronisation, examinez ce scénario. RTC dans AS300 envoie des mises à jour au sujet de 170.10.0.0. RTA et RTB exécutent iBGP, par conséquent RTB obtient la mise à jour et peut atteindre 170.10.0.0 par l'intermédiaire du prochain saut 2.2.2.1. 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 170.10.0.0 dans IGP. À ce niveau, RTE ne sait même pas que 170.10.0.0 existe.

Si RTB commence à annoncer à AS400 que RTB peut atteindre 170.10.0.0, le trafic entre RTD et RTB avec la destination 170.10.0.0 circule et s'arrête à RTE.

La synchronisation stipule que, si votre AS transmet le trafic d'un autre AS à un AS tiers, BGP ne doit pas annoncer de route avant que tous les routeurs de votre AS aient appris la route par l'intermédiaire d'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 170.10.0.0 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 170.10.0.0. Assurez-vous que les autres routeurs peuvent atteindre 170.10.0.0.

Désactiver la synchronisation

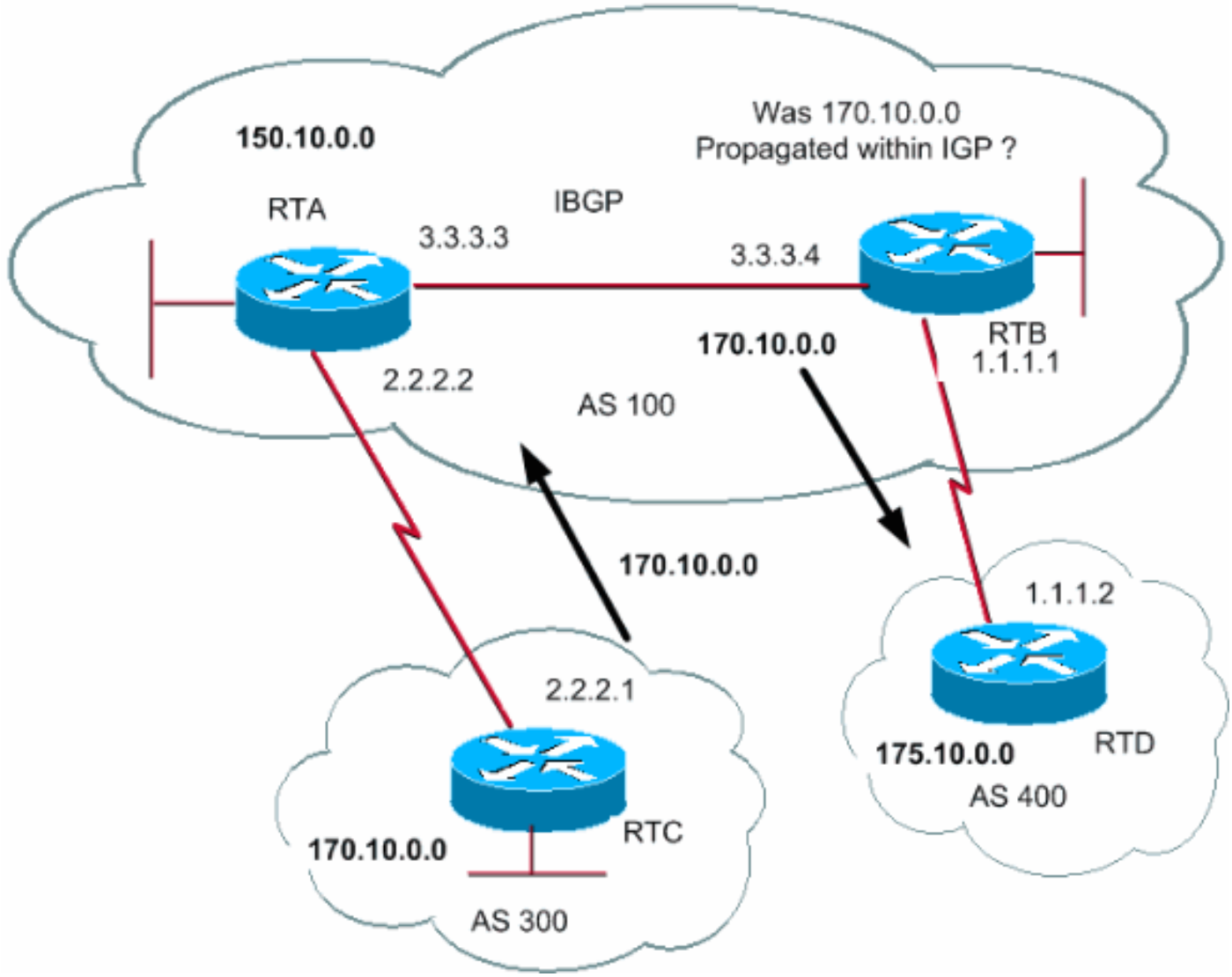
Dans certains cas, vous n'avez pas besoin de la synchronisation. Si aucun trafic issu d'un autre

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 :

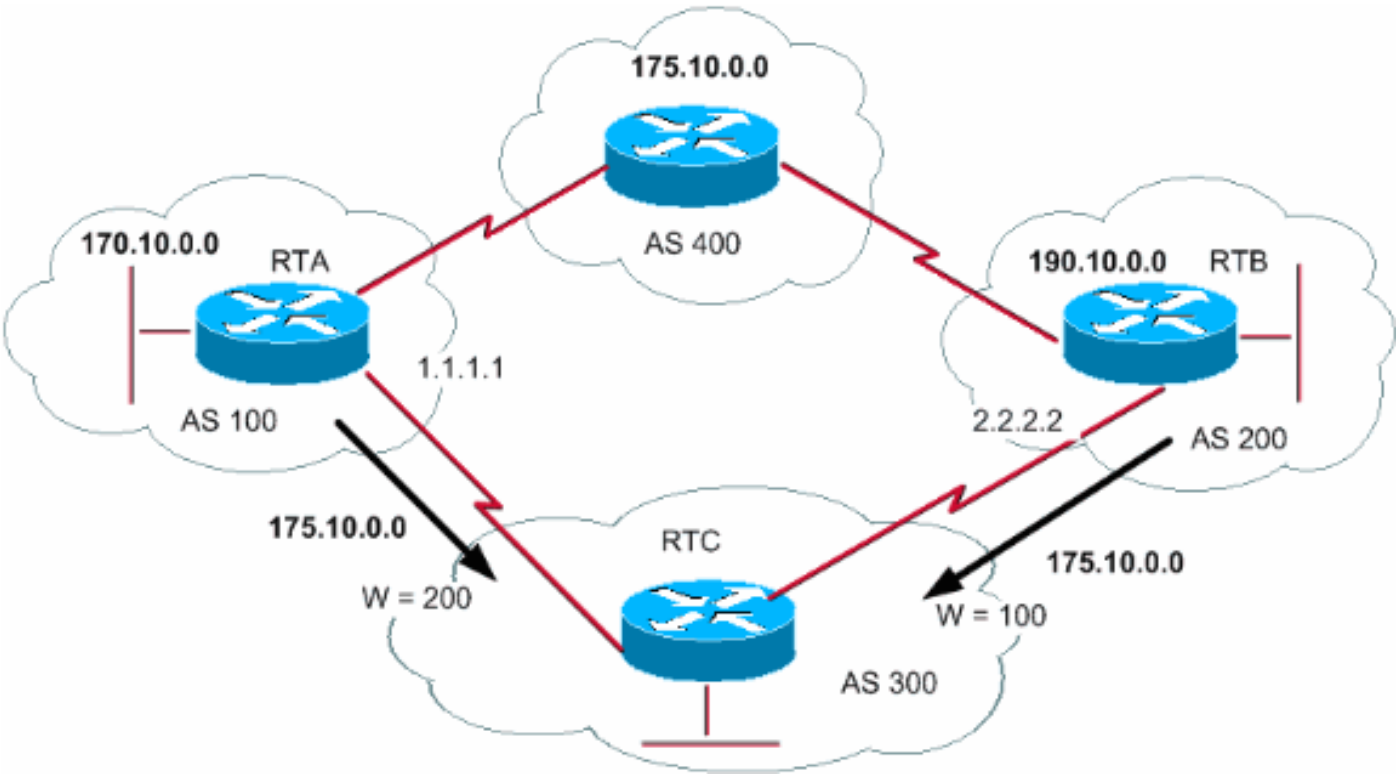
distance bgp *external-distance internal-distance local-distance*

Remarque: Assurez-vous que vous émettez la commande **clear ip bgp adress** pour réinitialiser la session.



distance bgp *external-distance internal-distance local-distance*

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 175.10.0.0 d'AS4. RTA propage la mise à jour à RTC. RTB a également appris le réseau 175.10.0.0 d'AS4. RTB propage la mise à jour à RTC. RTC dispose désormais de deux chemins pour atteindre 175.10.0.0 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 175.10.0.0. Plusieurs méthodes permettent de définir ce poids :

Utilisez la commande **neighbor**.

neighbor {adresse-ip | groupe-homologue} weight poids

Utilisez les listes d'accès AS_PATH.

ip as-path access-list access-list-number {permit | deny} as-regular-expression neighbor ip-address filter-list access-list-number weight weight

Utilisez des mises en correspondance de route.

distance bgp *external-distance internal-distance local-distance*

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.

distance bgp *external-distance internal-distance local-distance*

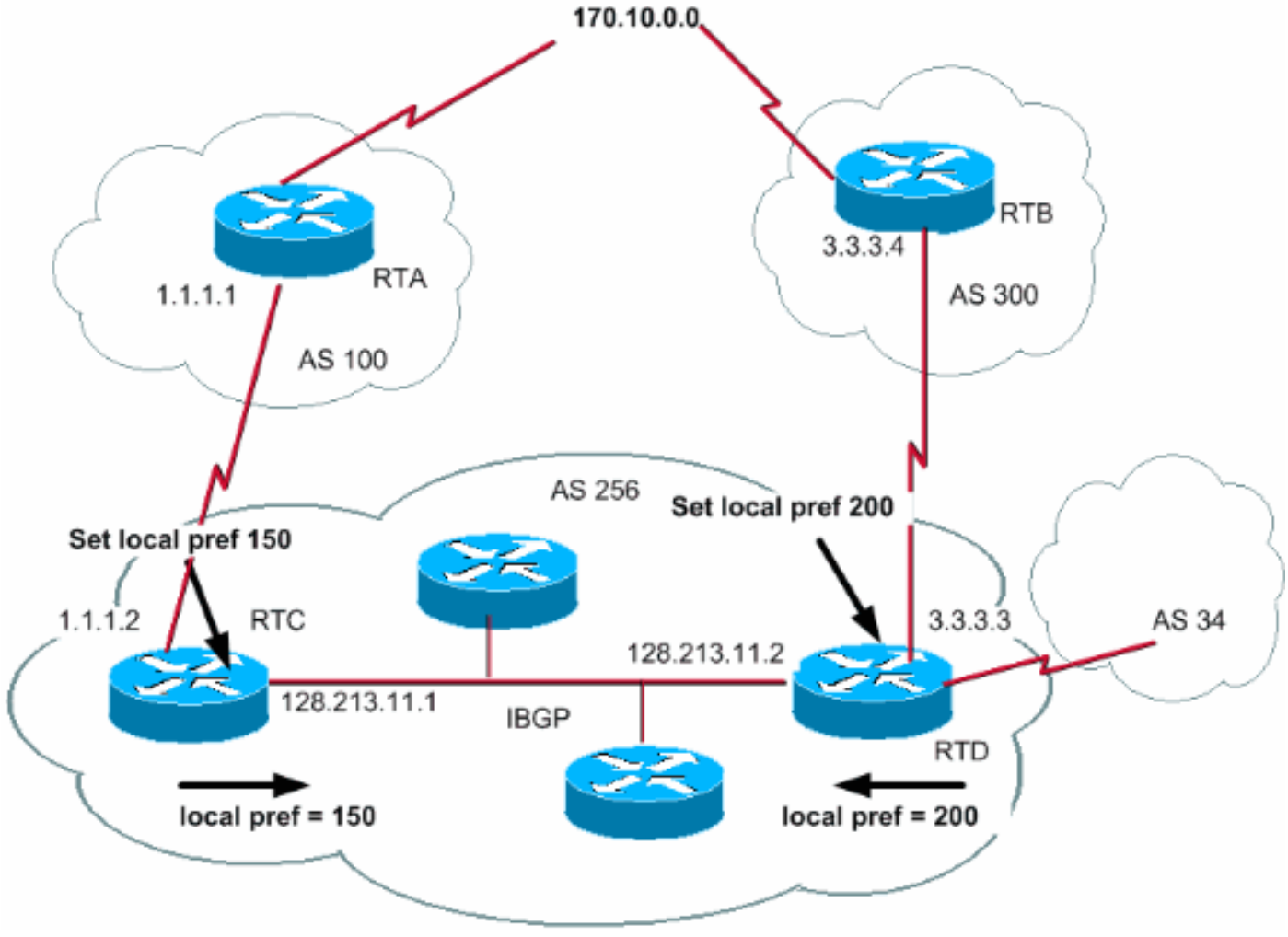
Vous pouvez également obtenir les mêmes résultats en utilisant des mises en correspondance de route.

distance bgp *external-distance internal-distance local-distance*

Remarque: Vous pouvez modifier le poids pour préférer le chemin BGP MPLS VPN avec le chemin d'IGP comme sauvegarde.

Remarque: Le pour en savoir plus, se rapportent à ce document de la Communauté de support de Cisco qui décrit comment configurer le routeur pour avoir un chemin préférentiel sur primaire et des conditions de panne et pour le rerouter sur la reprise de chemin primaire : Préférence du chemin BGP MPLS VPN avec la sauvegarde d'IGP

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.

Àla 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'exécuter une étiquette logicielle (c'est-à-dire, effacez le processus BGP sur le routeur) pour que les modifications soient rentrées à la considération. Afin d'effacer le processus BGP, utilisez le clear ip bgp **[doux] [**commande d'**entrée/sortie]** où le **doux** indique une étiquette logicielle sans déchirer la session et **[l'entrée/sortie]** spécifie la configuration d'arrivée ou sortante. Si l'**entrée/sortie** n'est pas spécifié des sessions d'arrivée et sortantes sont remises à l'état initial.

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 170.10.0.0 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 :

```
distance bgp external-distance internal-distance local-distance
```

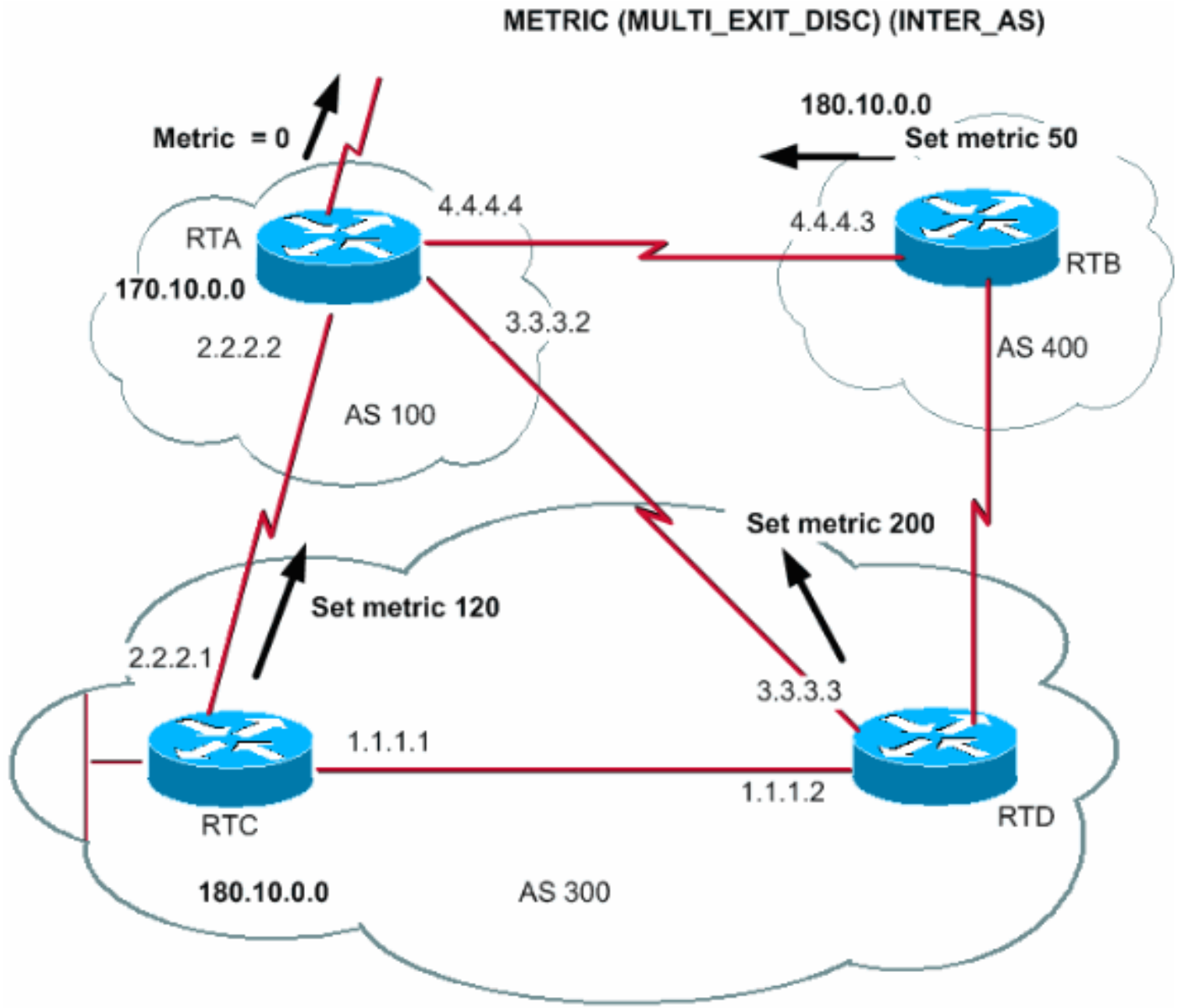
Dans cette configuration, RTC définit la préférence locale de toutes les mises à jour sur 150. Le même RTD définit la préférence locale de toutes les mises à jour sur 200. Il y a un échange de préférence locale dans AS256. Par conséquent, RTC et RTD se rendent compte que le réseau 170.10.0.0 a une préférence 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 :

```
distance bgp external-distance internal-distance local-distance
```

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.

Àla 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.

Àmoins 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 BGP peuvent influencer la sélection de chemin basée sur le discriminateur de sorties multiples (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 diagramme de cette section, AS100 obtient des informations sur le réseau 180.10.0.0 par l'intermédiaire de trois routeurs différents : RTC, RTD et RTB. RTC et RTD sont dans AS300, et RTB dans AS400.

Dans cet exemple, la comparaison d'AS-Path sur RTA par **bgp bestpath as-path ignore de** commande est ignorée. Il est configuré pour forcer le BGP pour tomber en fonction au prochain attribut pour la comparaison d'artère (dans ce cas mesure ou MED). Si la commande est omise, le BGP installera l'artère 180.10.0.0 du routeur RTC comme cela a l'AS-Path 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. Quand RTA obtient une mise à jour de RTB avec la mesure 50, RTA ne peut pas comparer la mesure à 120 parce que le RTC et les RTB sont dans Ass. différente RTA doivent choisir basé sur quelques 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 :

```
distance bgp external-distance internal-distance local-distance
```

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 :

```
distance bgp external-distance internal-distance local-distance
```

Dans ce cas, RTA sélectionne RTB comme meilleur prochain saut afin d'atteindre le réseau 180.10.0.0.

Vous pouvez également définir la métrique pendant la redistribution des routes vers BGP en émettant la commande **default-metric number**.

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

```
distance bgp external-distance internal-distance local-distance
```

Attribut community

L'attribut community est un attribut transitif facultatif situé entre 0 et 4 294 967 200. L'attribut

community permet de grouper les destinations d'une communauté donnée et d'appliquer des décisions de routage en fonction de ces communautés. 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 :

```
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 :

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

  ou

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

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 :

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

Dans le Logiciel Cisco IOS Version 12.0 et ultérieure, vous pouvez configurer les 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 au format AA : NN, émettez la commande **ip bgp-community new-format global configuration**. La première partie d'AA: NN représente le numéro de l'AS, la seconde partie représente un numéro à 2 octets.

Voici un exemple :

Sans la commande ip bgp-community new-format en configuration globale, l'émission de la commande show ip bgp 6.0.0.0 affiche la valeur de l'attribut community au format décimal. Dans cet exemple, la valeur de l'attribut community apparaît sous la forme 6553620.

```
Router# show ip bgp 6.0.0.0
BGP routing table entry for 6.0.0.0/8, 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 (200.200.200.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: 6553620
```

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

```
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 **ip bgp-community new-format global configuration**, la valeur de la communauté s'affiche au format AA: NN. La valeur apparaît sous la forme 100:20 dans la sortie de la commande **show ip bgp 6.0.0.0** de cet exemple :

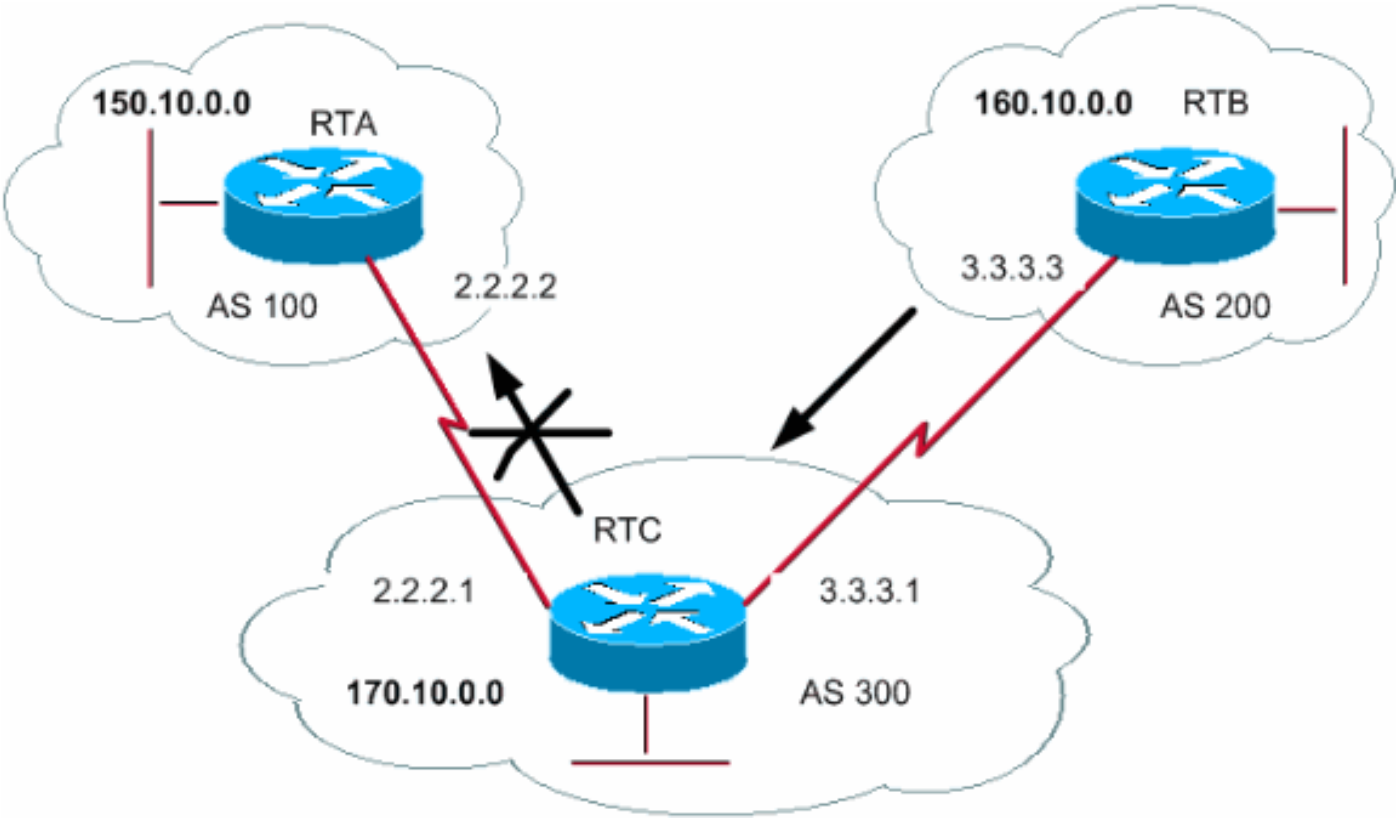
```
Router# show ip bgp 6.0.0.0
BGP routing table entry for 6.0.0.0/8, version 9
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  1
    10.10.10.1 from 10.10.10.1 (200.200.200.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: 100:20
```

Études de cas BGP 3

Filtrage des 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.

Filtrage de route



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 :

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

Dans cet exemple, RTB initie le réseau 160.10.0.0 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 :

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

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 160.0.0.0/8.

Remarque: La notation /8 signifie que vous utilisez 8 bits de masque de sous-réseau, à partir de l'extrême gauche de l'adresse IP. Cette adresse est équivalente à 160.0.0.0 255.0.0.0.

La commande **access-list 1 permit 160.0.0.0 0.255.255.255** autorise 160.0.0.0/8, 160.0.0.0/9, etc. Pour restreindre la mise à jour uniquement à 160.0.0.0/8, vous devez utiliser une liste d'accès étendue au format suivant :

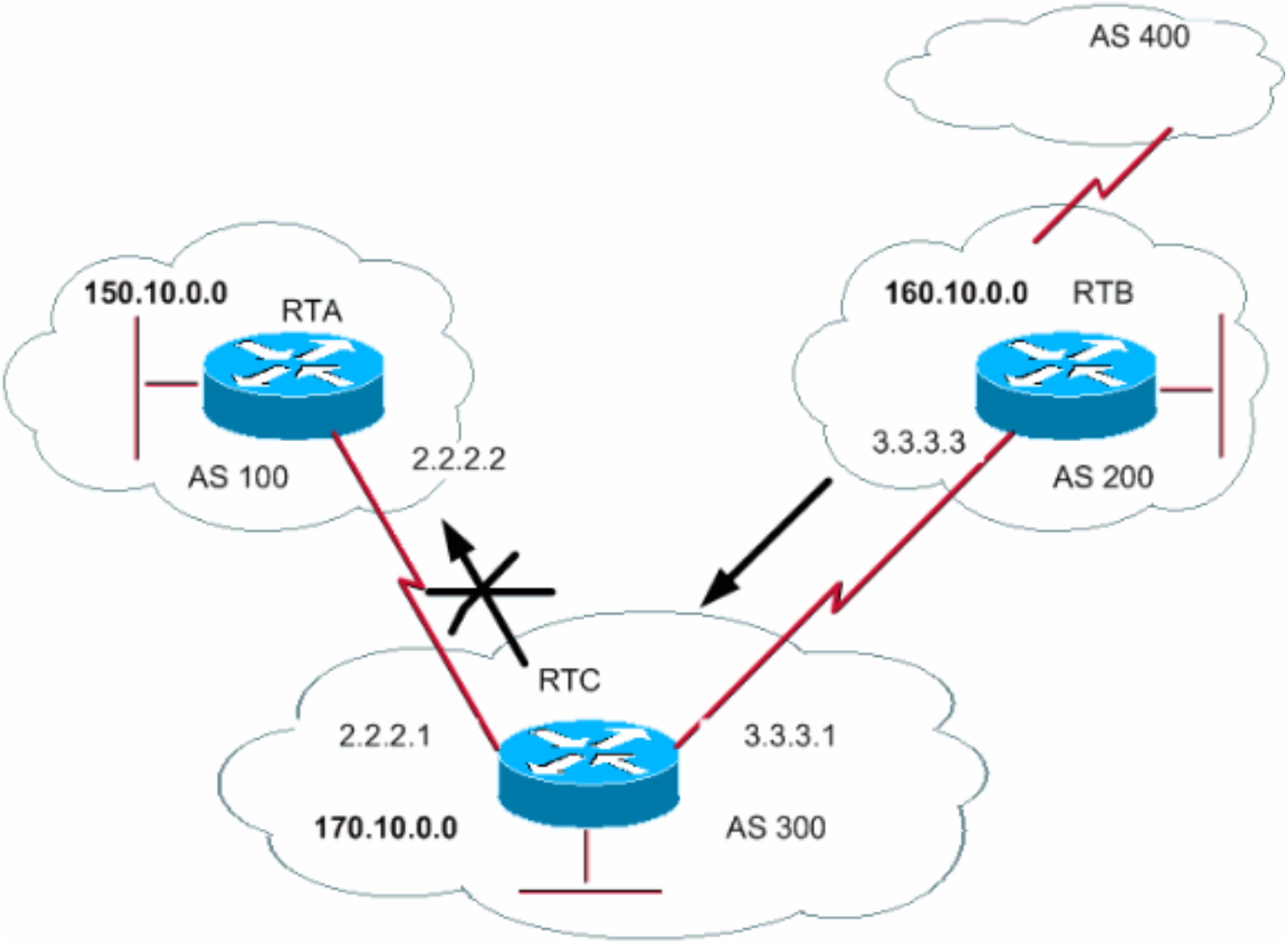
```
access-list 101 permit ip 160.0.0.0 0.255.255.255 255.0.0.0 0.0.0.0.
```

Cette liste autorise uniquement 160.0.0.0/8.

Référez-vous à Comment bloquer un ou plusieurs réseaux d'un homologue BGP pour obtenir des exemples de configuration sur la façon de filtrer les réseaux des homologues BGP. La méthode utilise la commande **distribute-list** avec des listes de contrôle d'accès (ACL) standard et étendues ainsi que le filtrage des listes de préfixes.

Filtrage de chemin

Un autre type de filtrage est le filtrage de chemin.



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 160.10.0.0 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 :

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

```
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 160.10.0.0 à RTA :

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

La commande **access-list 1** de cet exemple force le refus des mises à jour avec des informations de chemin commençant par 200 et se terminant par 200. ^200$ dans la commande est une « expression régulière » dans laquelle ^ signifie « commence par » et $ signifie « se termine par ». Puisque RTB envoie les mises à jour sur 160.10.0.0 avec les informations de chemin qui commencent par 200 et se terminent par 200, les mises à jour correspondent à la liste d'accès. La liste d'accès refuse ces mises à jour.

. ** est une autre expression régulière dans laquelle . signifie « n'importe quel caractère » et * signifie « la répétition de ce caractère ». Ainsi. ** représente n'importe quelle information de chemin nécessaire pour permettre la transmission de toutes les autres mises à jour.

Que se passe-t-il si, au lieu d'utiliser ^200$, vous utilisez ^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 parce que les informations de 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.

Afin de contrôler si vous avez mis en application l'expression régulière correcte, émettez la commande show ip bgp regexp regular-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 Filtrage de chemin, vous avez spécifié la chaîne ^200$. Vous vouliez que les informations de chemin intégrées aux mises à jour correspondent à la chaîne afin de prendre une décision.

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 :


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

. correspond à n'importe quel caractère unique.

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

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

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

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

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

Le \ correspond au caractère.

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

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

Partie

Une partie est l'un de ces symboles qui suit un atome :

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

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

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

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

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

? correspond à l'atome ou à la chaîne Null.

Branchement

Une branche est composée de 0 ou plusieurs parties concaténées.

Voici quelques exemples d'expressions régulières :

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

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

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

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

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

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

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

Cette expression signifie par l'intermédiaire d'AS100.

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

Cette expression indique l'origine AS100.

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

Cette expression indique une transmission depuis AS100.

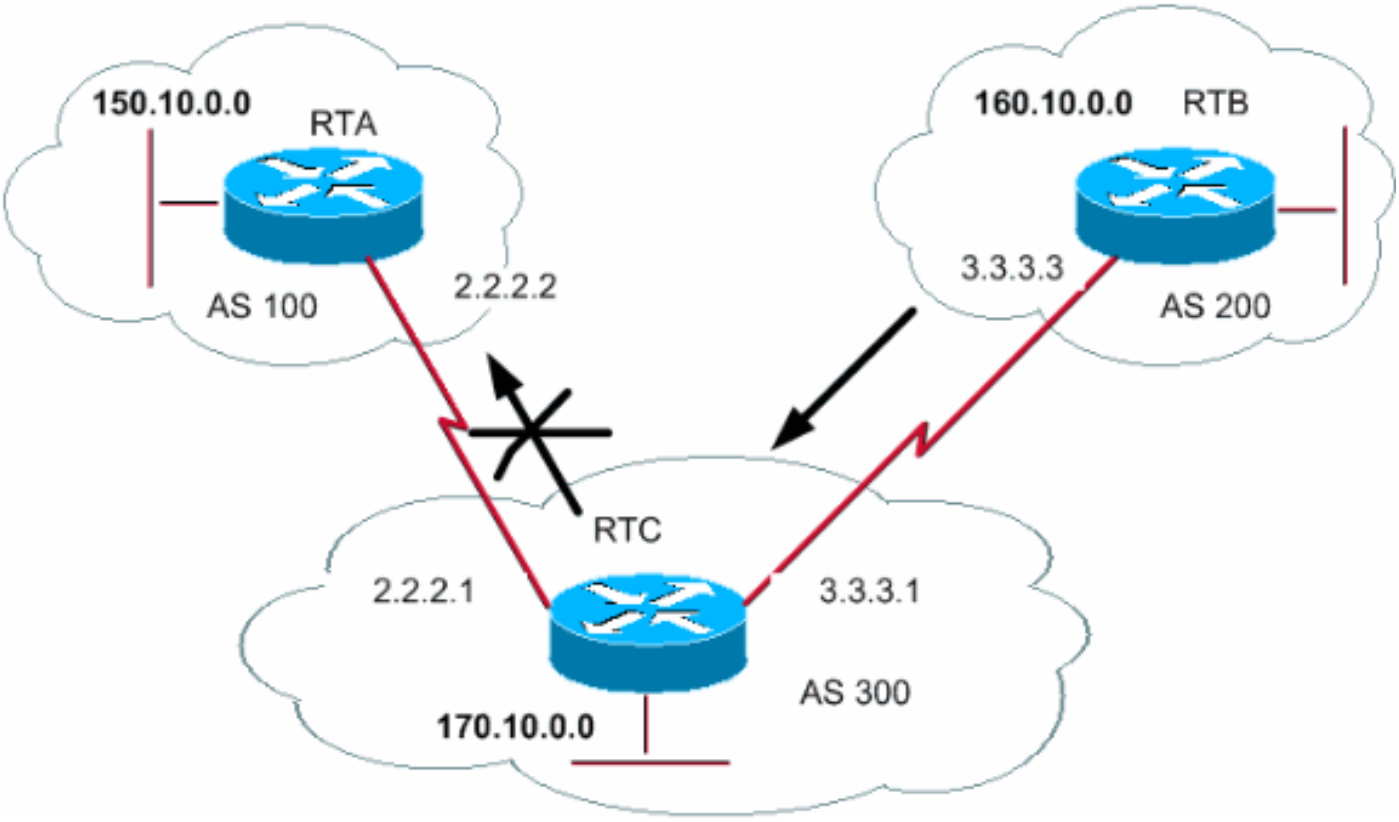
```
neighbor {ip-address | peer-group-name} filter-list access-list-number {in | out}
```

Cette expression indique une origine depuis cet AS.

Référez-vous à Utilisation d'expressions régulières dans BGP pour obtenir des exemples de configuration de filtrage d'expression régulière.

Filtrage de la 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 Attribut community présente la communauté, et cette section fournit quelques exemples d'utilisation de la 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-export community**.

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

Remarque: Cet exemple utilise la commande **route-map setcommunity** pour définir la communauté sur **no-export**.

Remarque: La commande **neighbor send-community** est nécessaire afin d'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 example, RTB a défini l'attribut community sur **100 200 additive**. Cette action ajoute la valeur 100 200 à n'importe quelle valeur existante de la communauté avant la transmission à RTC.

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

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.

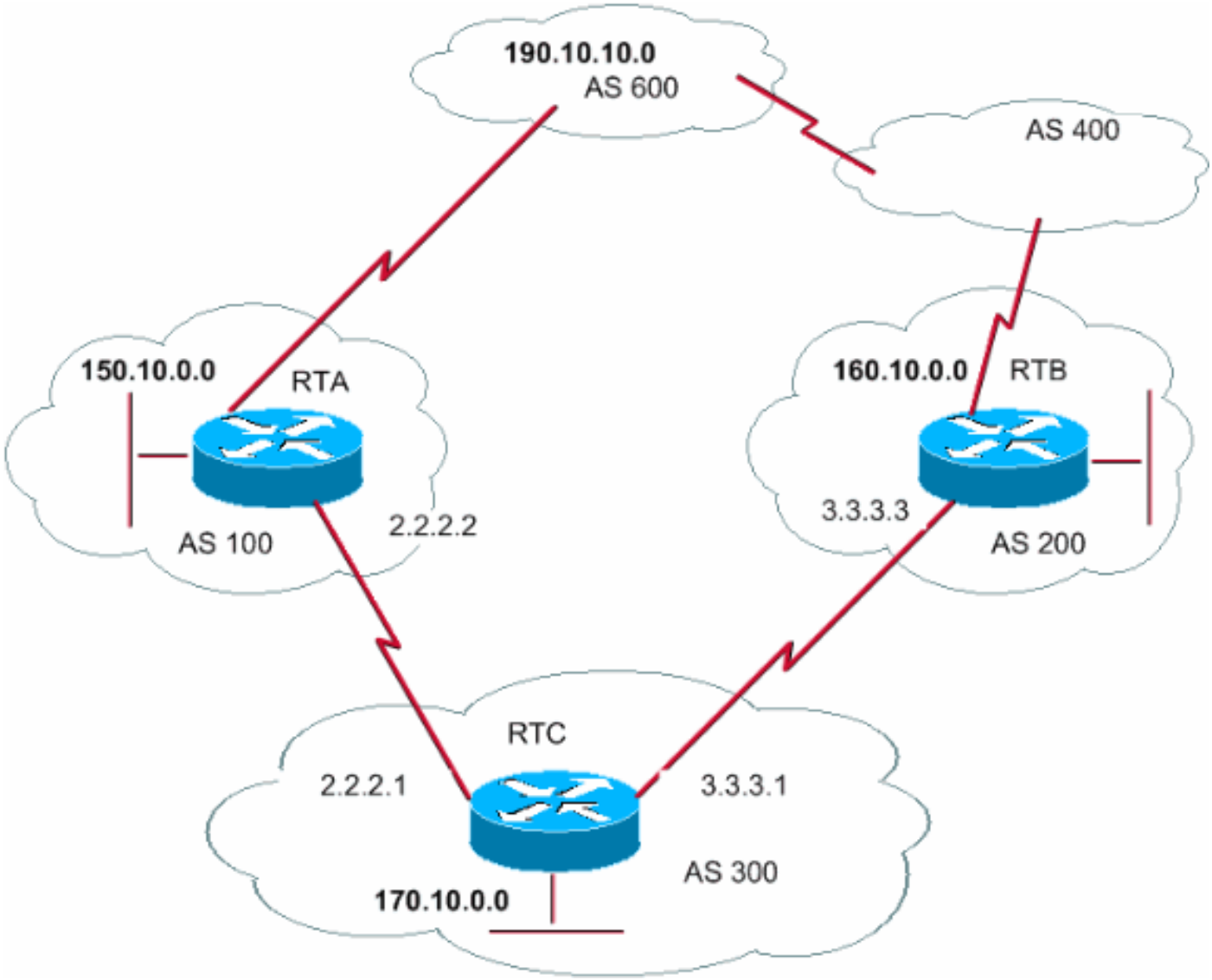
ip community-list *community-list-number* **{permit | deny}** *community-number*
Par exemple, vous pouvez définir cette mise en correspondance de route, **match-on-community** :

ip community-list *community-list-number* **{permit | deny}** *community-number*
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 :

ip community-list *community-list-number* **{permit | deny}** *community-number*
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.

Référez-vous à Utilisation des valeurs de la communauté BGP pour contrôler la politique de routage dans le réseau du fournisseur en amont pour plus d'informations.

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 :

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** :

neighbor *ip-address* **route-map** *route-map-name*

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.

neighbor *ip-address* **route-map** *route-map-name*
Cette instruction définit un poids de 20 pour les mises à jour qui sont locales à AS200. L'instruction définit également un poids de 10 pour les mises à jour qui sont derrière AS400, et rejette les mises à jour qui viennent d'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 :

set as-path prepend *as-path# as-path#*

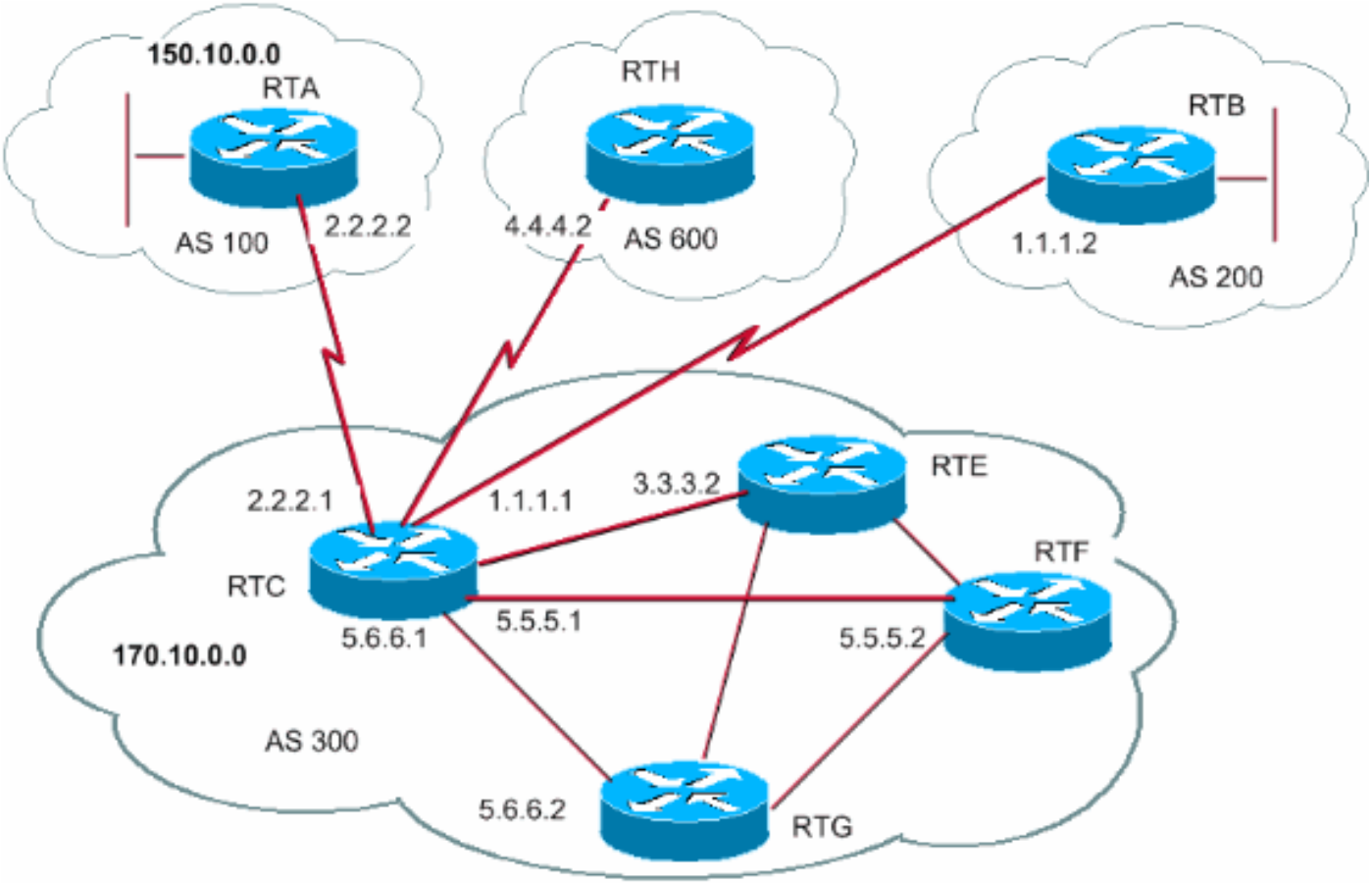
Supposez que, dans le diagramme de la section BGP Neighbors et Route Maps, RTC annonce son propre réseau 170.10.0.0 sur deux différents AS, AS100 et AS200. Quand les informations sont propagées à AS600, les Routeurs dans AS600 ont les informations sur l'accessibilité de réseau sur 170.10.0.0 par l'intermédiaire 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. Vous pouvez faire ceci si vous préfixez des numéros AS aux informations du chemin d'accès existant qui est annoncé à AS100. Une pratique courante est de répéter vos propres numéros AS de cette façon :

set as-path prepend *as-path# as-path#*

Àcause de cette configuration, AS600 reçoit des mises à jour au sujet de 170.10.0.0 par l'intermédiaire d'AS100 avec les informations de chemin : (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 ; au lieu de cela, vous définissez un nom de groupe d'homologues et assignez ces stratégies au groupe d'homologues.

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 :

```
neighbor peer-group-name peer-group
```

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

```
neighbor peer-group-name peer-group
```

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 affectent 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.

neighbor *peer-group-name* **peer-group**

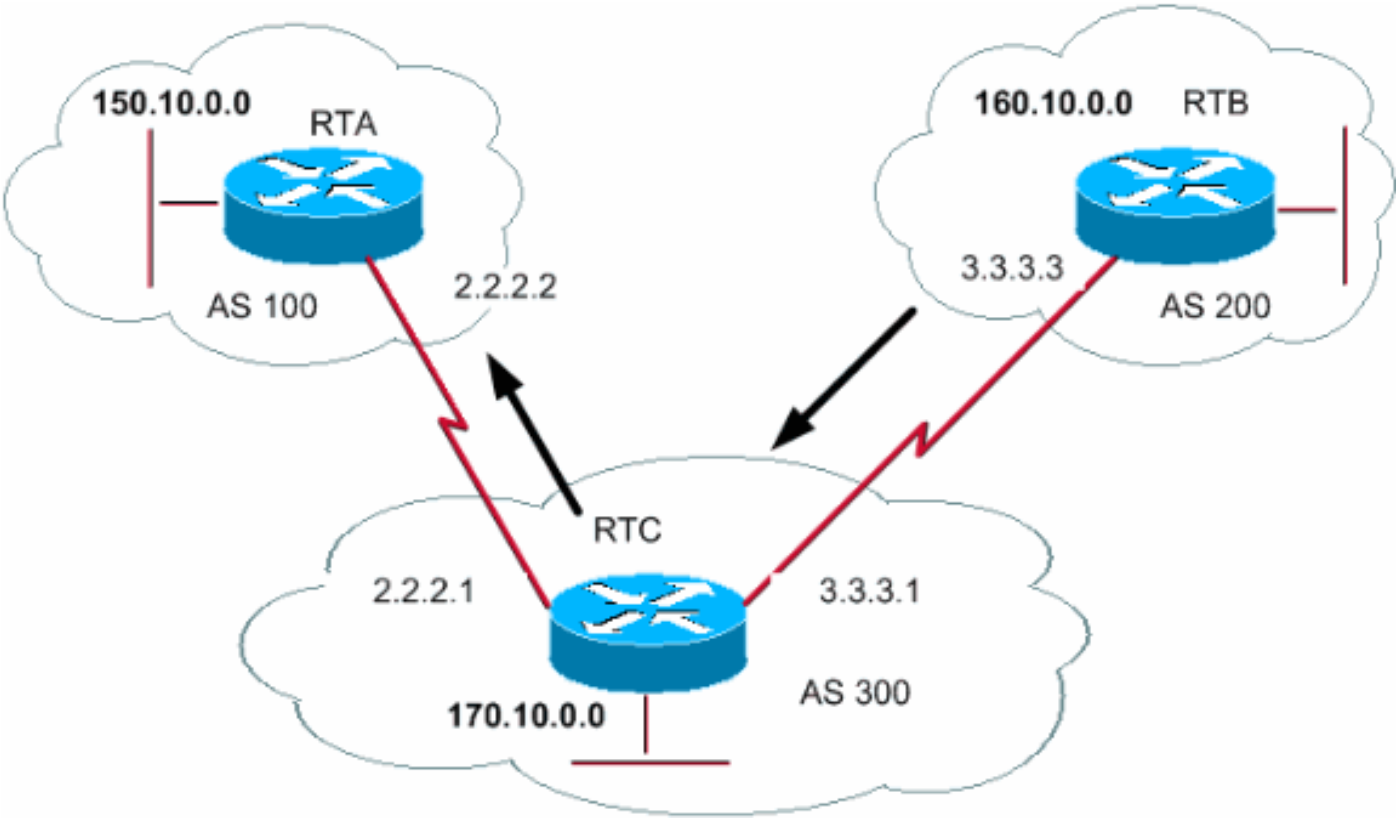
Remarque: Dans ces configurations, vous définissez les instructions **remote-as** en dehors du groupe d'homologues parce que vous devez définir des AS externes différents. En outre, vous remplacez les mises à jour entrantes du voisin 1.1.1.2 avec l'attribution de la liste de filtres 3.

Pour plus d'informations sur les groupes d'homologues, référez-vous à la section Groupes d'homologues BGP.

Remarque: Dans le logiciel Cisco IOS Version 12.0(24)S, Cisco introduit la fonctionnalité de groupes d'homologues de mise à jour dynamiques BGP. 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.

Études de cas BGP 4

CIDR et adresses agrégées



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 des classes, telles que la classe A, B, ou le C. par exemple, réseau 192.213.0.0 était par le passé un réseau de classe C illégal. Maintenant, le réseau est un super-réseau légal, 192.213.0.0/16. Le « 16 » représente le nombre de bits dans le masque de sous-réseau, quand vous comptez à partir de l'extrême gauche de l'adresse IP. Cette représentation est semblable à 192.213.0.0 255.255.0.0.

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 160.10.0.0. Vous configurez RTC pour propager un super-réseau de cette route 160.0.0.0 à RTA :

neighbor *peer-group-name* **peer-group**
RTC propage l'adresse agrégée 160.0.0.0 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 dans la section CIDR et adresses agrégées :

aggregate-address *address-mask*

Cette commande annonce la route du préfixe et toutes les routes plus spécifiques. La commande **aggregate-address 160.0.0.0** propage un réseau 160.0.0.0 supplémentaire mais n'empêche pas la propagation de 160.10.0.0 à RTA. Les résultats sont la propagation des réseaux 160.0.0.0 et 160.10.0.0 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 une route plus spécifique pour cette adresse dans la table de routage BGP.

Par exemple, RTB ne peut pas produire un agrégat pour 160.0.0.0 si RTB n'a pas une entrée de 160.0.0.0 plus spécifique dans la table 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 160.10.0.0**.

Si vous voulez que RTC propage le réseau 160.0.0.0 seulement et **non** la route plus spécifique, exécutez cette commande :

aggregate-address *address mask* **summary-only**

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

La commande **aggregate 160.0.0.0 255.0.0.0 summary-only** propage le réseau 160.0.0.0 et supprime la route plus spécifique 160.10.0.0.

Remarque: Si vous agrégez un réseau qui a été injecté dans votre BGP par l'intermédiaire de l'instruction **network**, l'entrée dans le réseau s'injecte toujours dans les mises à jour 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.

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.

aggregate 129.0.0.0 255.0.0.0 as-set

La section Exemple CIDR 2 (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.

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 160.0.0.0, supprimer la route 160.20.0.0 plus spécifique et permettre la propagation de 160.10.0.0. Utilisez cette mise en correspondance de route :

aggregate-address *address-mask* **suppress-map** *map-name*
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.

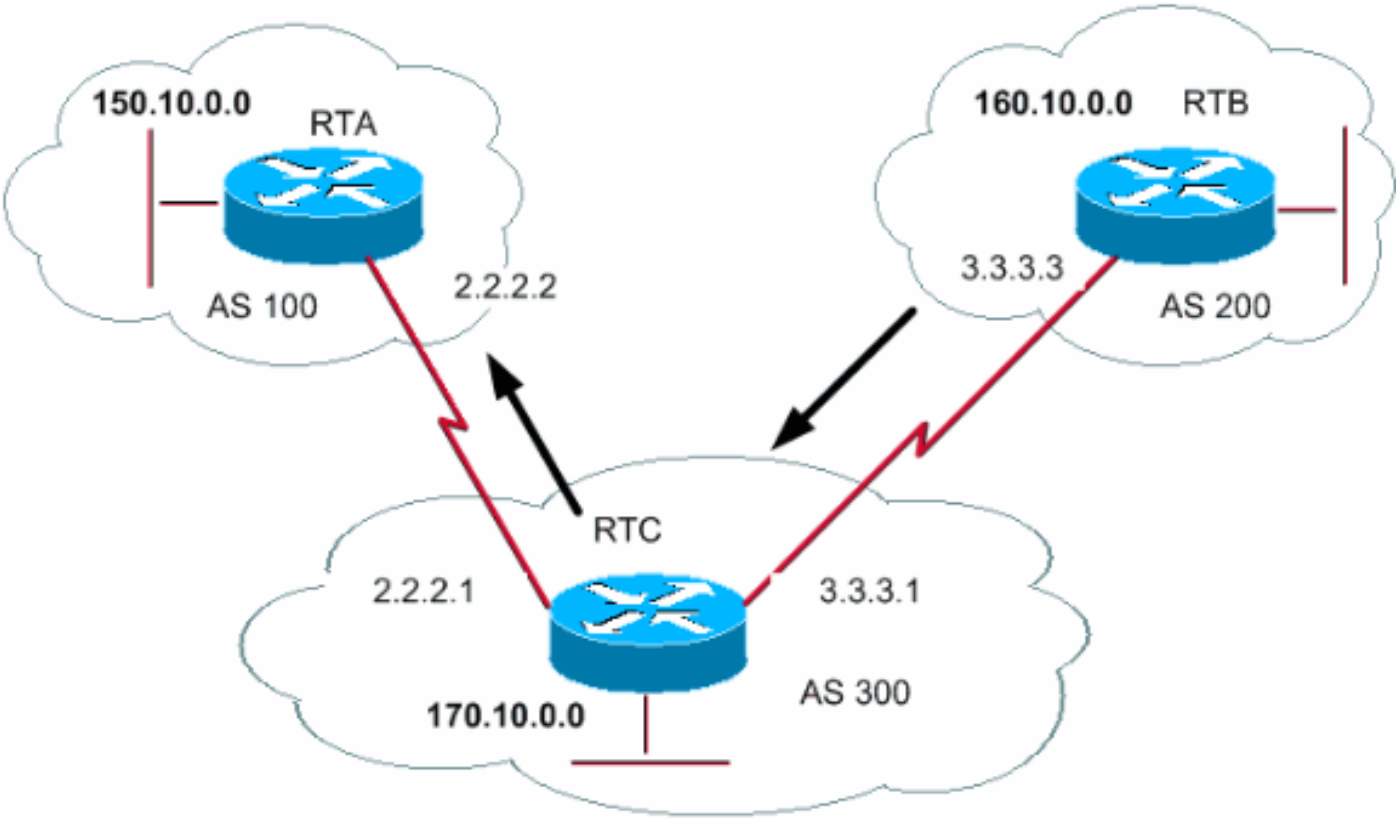
Appliquez ensuite la feuille de route à l'instruction **aggregate**.

aggregate-address *address-mask* **suppress-map** *map-name*
Voici une autre variante :

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** :

aggregate-address *address-mask* **attribute-map** *map-name*
Pour plus d'informations, référez-vous à la section Comprendre l'agrégation de routes dans BGP.

Exemple CIDR 1



Demande : Permettez à RTB d'annoncer le préfixe 160.0.0.0 et de supprimer toutes les routes plus spécifiques. Le problème avec cette requête est que le réseau 160.10.0.0 est local à AS200, ce qui signifie qu'AS200 est le créateur de 160.10.0.0. Vous ne pouvez pas obtenir que RTB génère un préfixe pour 160.0.0.0 sans génération d'une entrée pour 160.10.0.0, même si vous utilisez la commande **aggregate summary-only** . RTB produit les deux réseaux parce que RTB est le créateur de 160.10.0.0. Il y a deux solutions à ce problème.

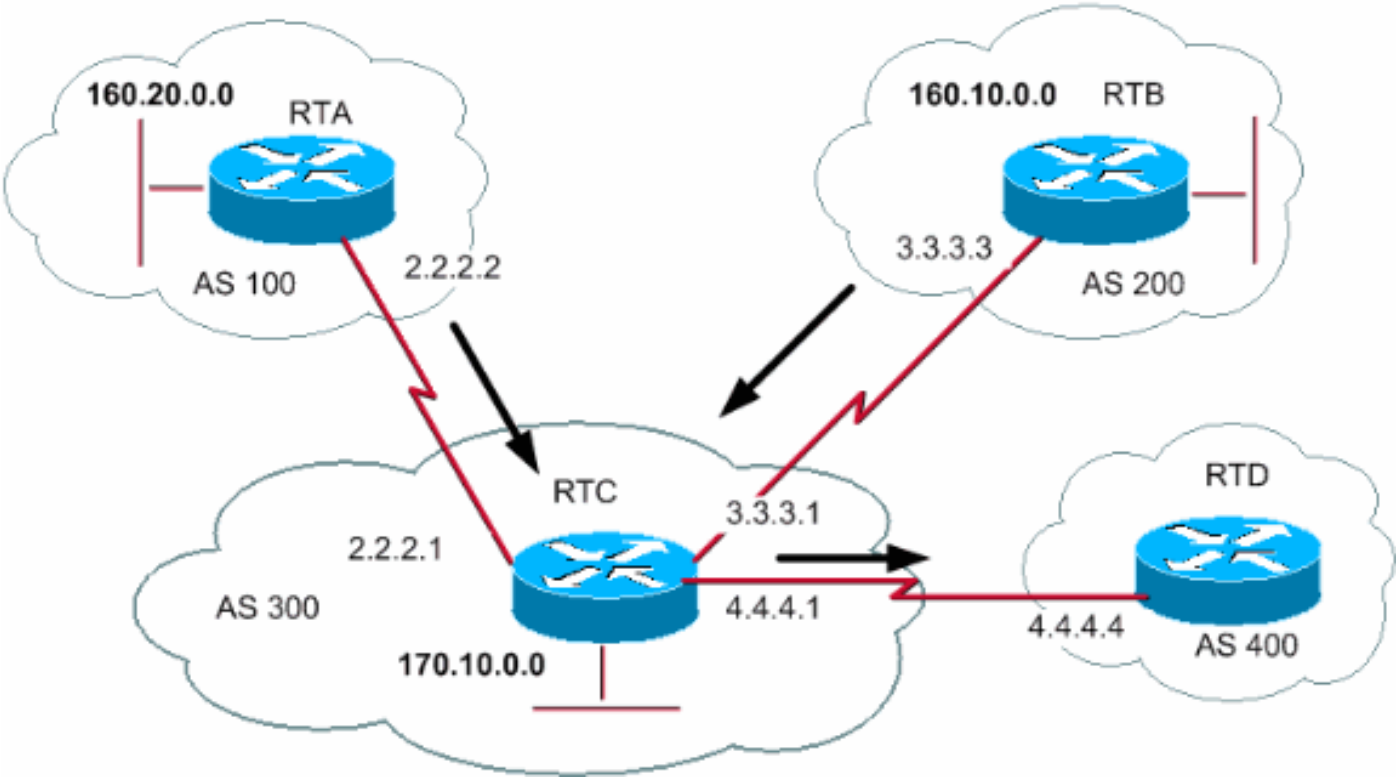
La première solution est d'utiliser une route statique et de redistribuer dans BGP. Les résultats sont que RTB annonce l'agrégat avec une origine inachevée (?).

aggregate-address *address-mask* **attribute-map** *map-name*
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.

aggregate-address *address-mask* **attribute-map** *map-name*

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 160.20.0.0 de RTA et des mises à jour sur 160.10.0.0 de RTB. Supposez que RTC veuille agréger le réseau 160.0.0.0/8 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.



aggregate-address *address-mask* **attribute-map** *map-name*
Cas 1 :

RTC n'a pas une instruction **as-set**. RTC envoie une mise à jour 160.0.0.0/8 à RTD avec les informations du chemin (300), comme si la route provenait d'AS300.

aggregate-address *address-mask* **attribute-map** *map-name*
Cas 2 :

aggregate-address *address-mask* **attribute-map** *map-name*
Les deux prochaines rubriques, Confédération BGP et Réflecteurs de route, s'adressent aux fournisseurs de services Internet (ISP) qui souhaitent contrôler davantage l'explosion de l'appairage iBGP dans leur 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 :

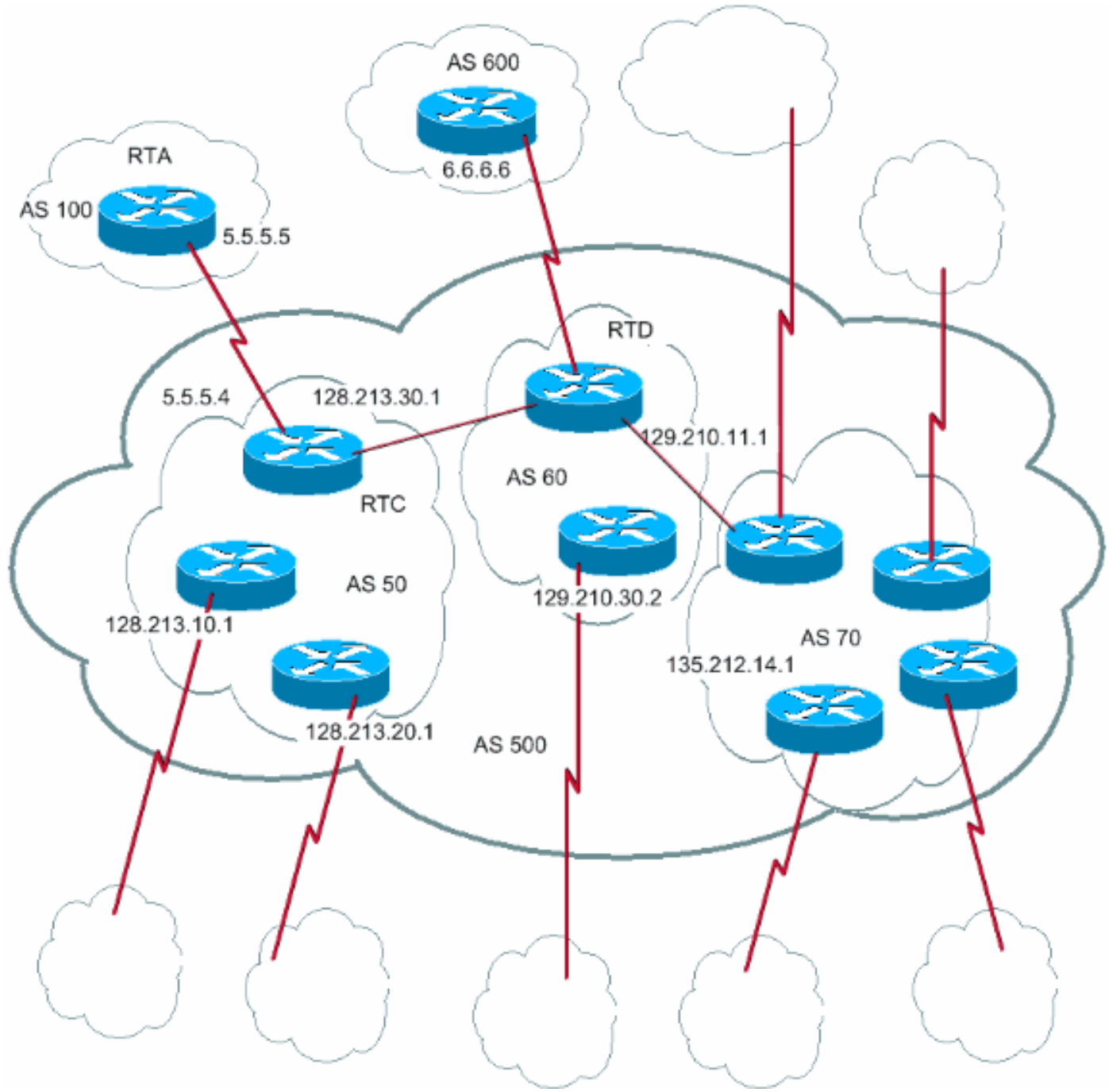
```
bgp confederation identifier autonomous-system
```

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 :

```
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**.

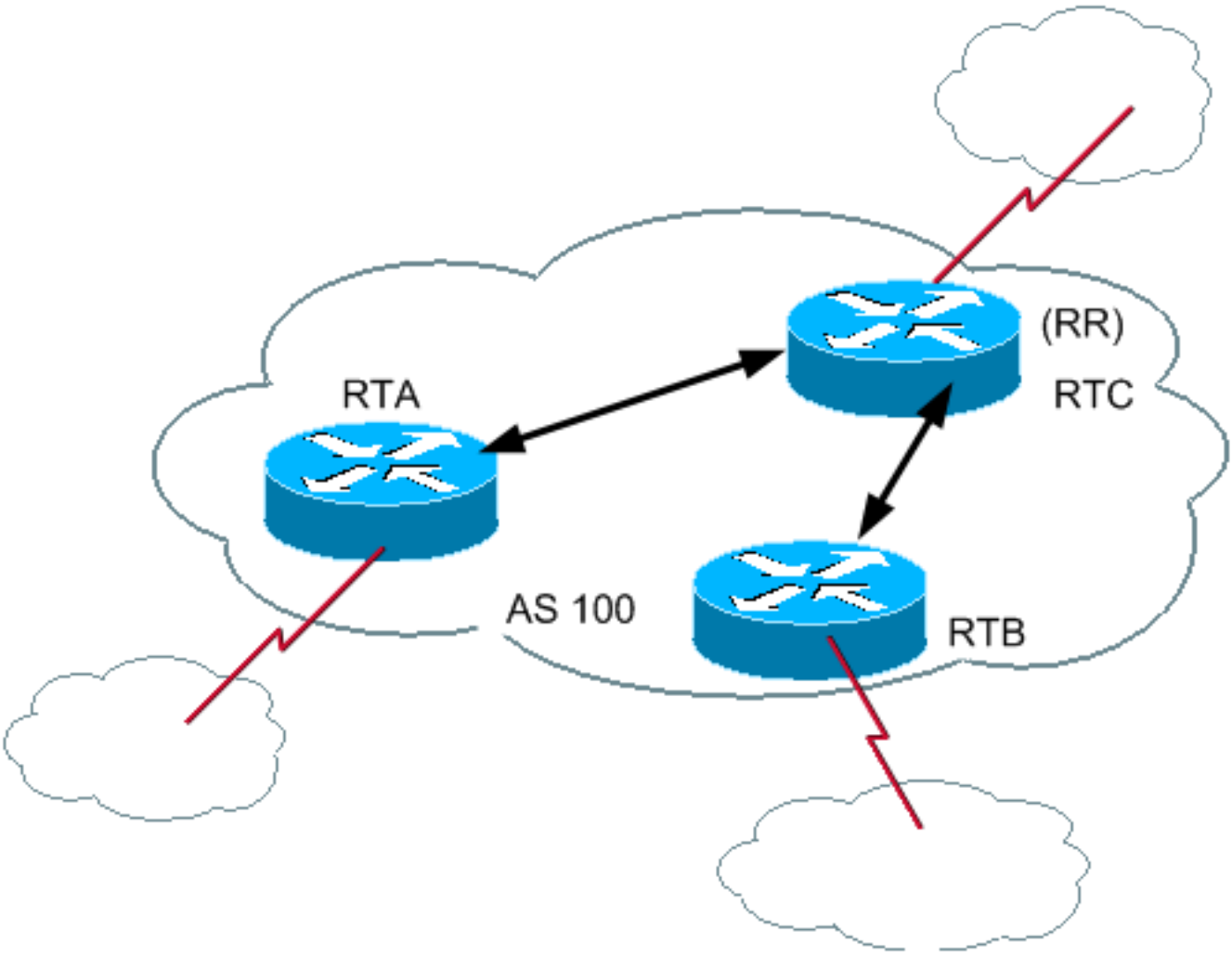
Voici un exemple de configuration des routeurs RTC, RTD et RTA :

Remarque: RTA ne connaît pas AS50, AS60 ou AS70. RTA connaît seulement AS500.

bgp confederation peers autonomous-system [*autonomous-system***]**

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 expliqué dans la section iBGP , un speaker BGP n'annonce pas une route qu'il a apprise par l'intermédiaire d'un autre speaker iBGP sur un troisième speaker 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.

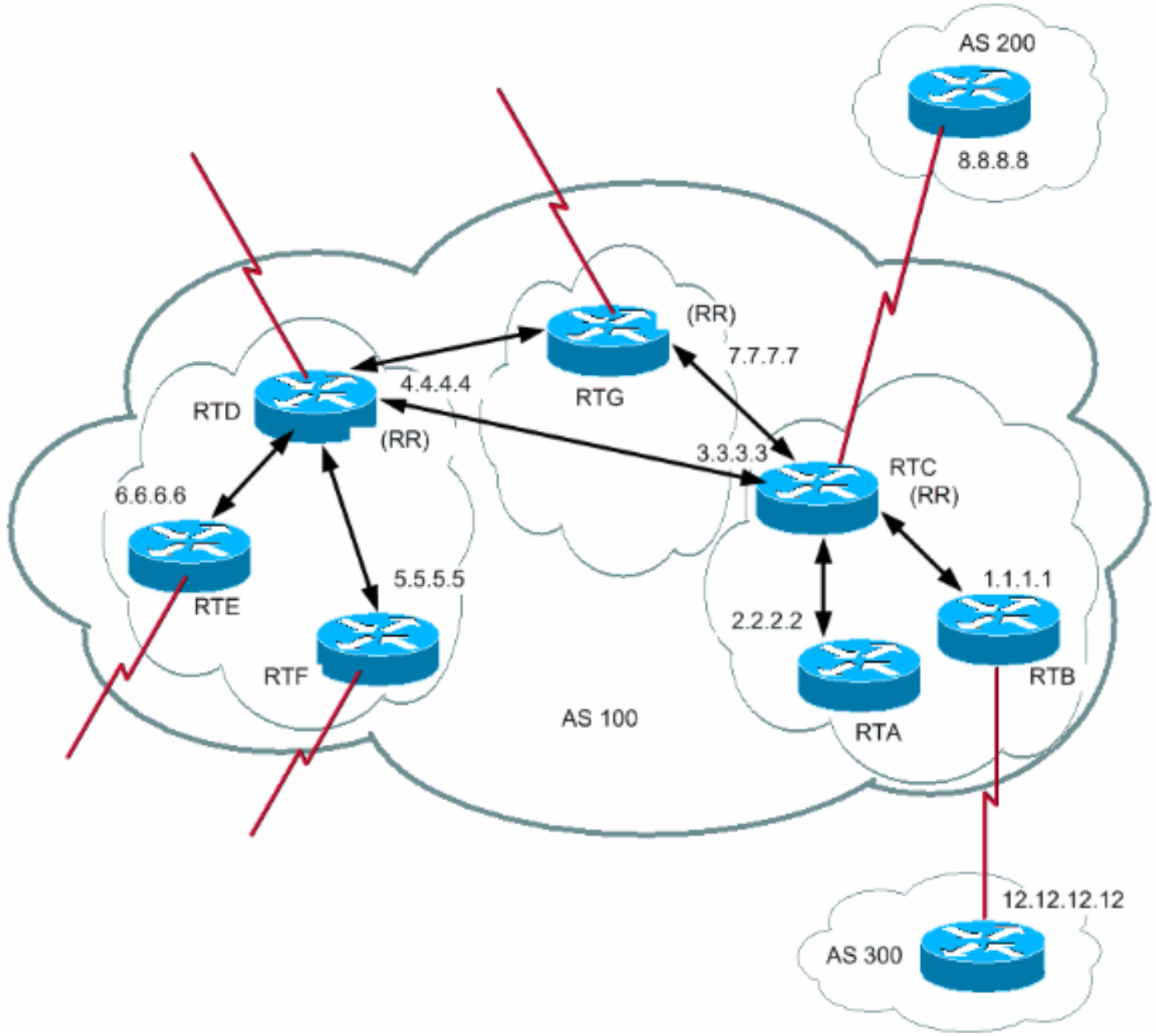


Dans des cas normaux, maintenez un maillage iBGP complet entre RTA, RTB et RTC dans AS100. Si vous utilisez le concept RR, RTC peut être choisi en tant que RR. De cette façon, RTC a un iBGP partiel pour l'appairage avec RTA et RTB. L'appairage entre RTA et RTB n'est pas nécessaire parce que RTC est un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

neighbor 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.

D'autres homologues iBGP du RR qui ne sont pas des clients sont des « nonclients ».



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 devraient pas avoir un appairage avec des speakers iBGP en dehors du cluster client.

Considérez ce diagramme. RTA, RTB et RTC forment un cluster unique. 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: RTD, RTC et RTG sont entièrement maillés, mais les routeurs dans un cluster ne le sont pas. 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 :

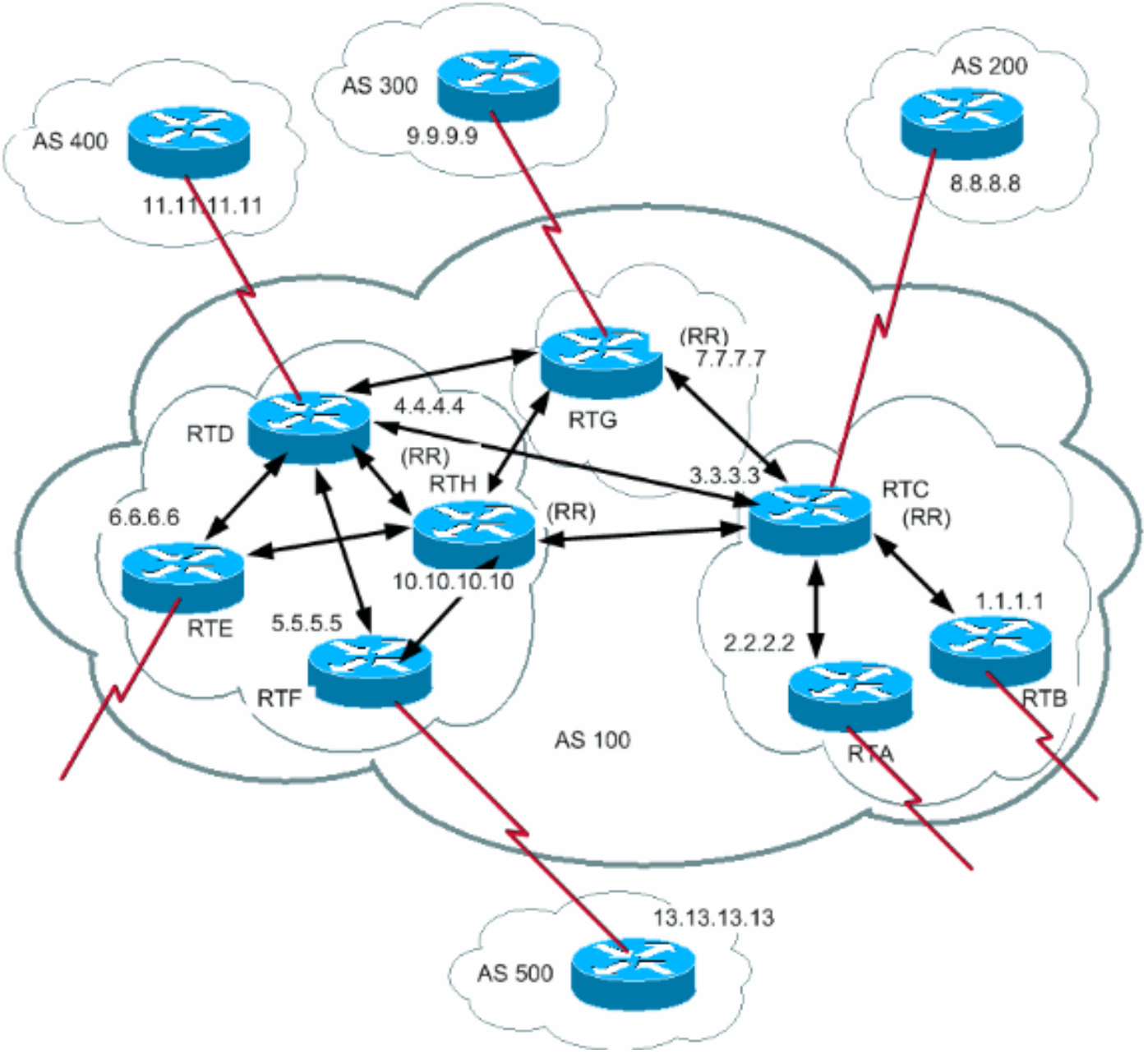
`neighbor route-reflector-client`

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 Plusieurs RR dans un cluster traite de la liste de clusters.

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 une redondance parce que RTH a entièrement maillé l'appairage avec tous les RRs. Si RTD s'arrête, RTH remplace RTD.

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

`neighbor route-reflector-client`

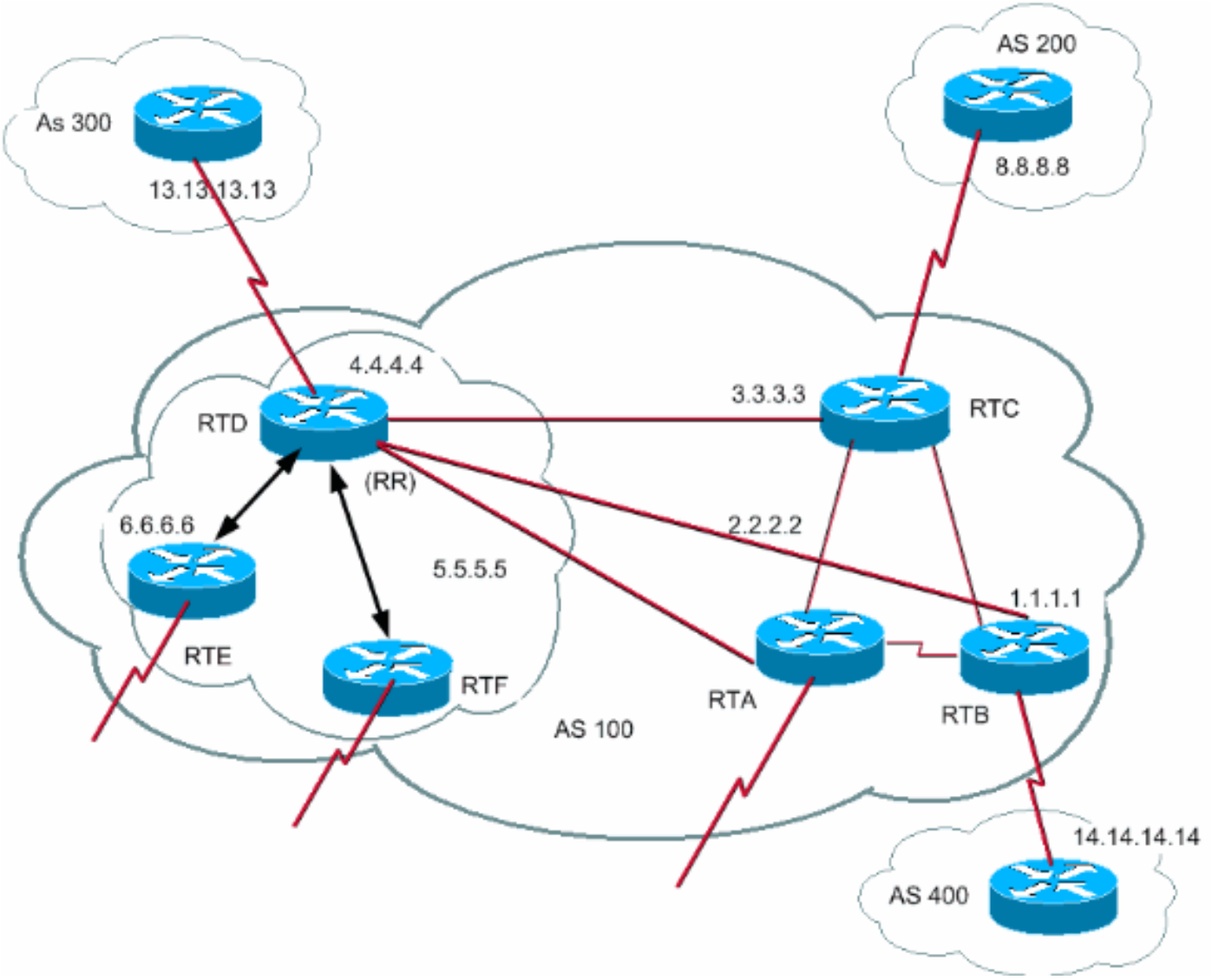
Remarque: Vous n'avez pas besoin de la commande bgp cluster-id pour RTC parce que seul un RR existe dans ce cluster.

Remarque importante : 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. Référez-vous aux limites de pour en savoir plus de groupes de homologues.

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 « conventionnels ». 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 ne doivent pas apprendre le schéma de réflexion de route ; seulement les RRs ont besoin de la mise à niveau.

Voici la configuration de RTD et RTC :

neighbor route-reflector-client

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'ici, ce document a mentionné deux attributs que vous pouvez utiliser pour empêcher le bouclage potentiel de l'information : **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 utiliser plus de restrictions pour **nexthop-self**, qui est une option de configuration par voisin. Quand vous utilisez **nexthop-self** sur RR, la clause affecte seulement le prochain saut des routes eBGP acquises parce que le prochain saut des routes reflétées ne devrait pas être changé.

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 cumulée atteint une « limite de suppression » prédéfinie, la suppression de l'annonce de la route se produit. La pénalité décline exponentiellement en fonction d'une « moitié de durée de vie » préconfigurée. Une fois que la pénalité décroît au-dessous d'une « limite de réutilisation » prédéfinie, la non suppression de l'annonce de route se produit.

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. La non suppression des routes se fait à une granularité de 10 secondes. Le routeur conserve les informations de dampening 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. À l'avenir, en cas de besoin, cette fonctionnalité peut être activée par défaut. Ces commandes contrôlent le route dampening :

bgp dampening - Active l'atténuation.

no bgp dampening — Désactive l'atténuation.

bgp dampening half-life-time — Modifie la moitié de la durée de vie.

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

bgp dampening half-life-time reuse suppress maximum-suppress-time

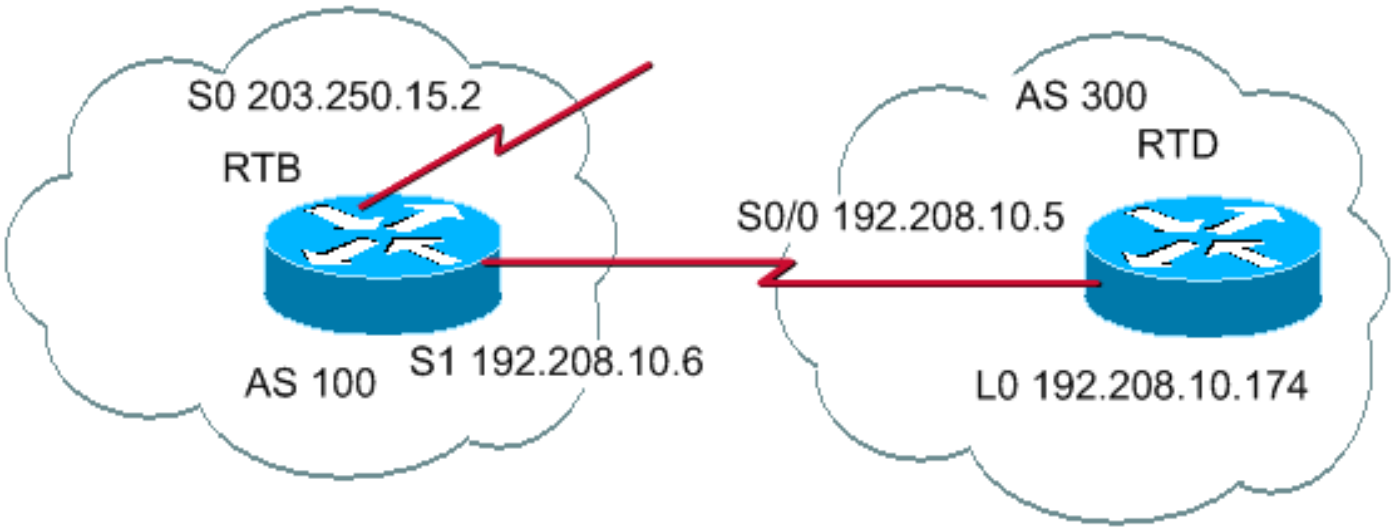
Cette liste détaille la syntaxe :

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

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

suppress-value — La plage est 1 – 20 000, et la valeur par défaut est 2000.

max-suppress-time — C'est la durée maximale de la suppression d'une route. La plage est 1 – 255 minutes, et la valeur par défaut est de 4 fois la moitié de la durée de vie.



neighbor route-reflector-client

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 :

```
RTB# show ip bgp
BGP table version is 24, local router ID is 203.250.15.2 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
*> 192.208.10.0     192.208.10.5           0             0 300 i
*> 203.250.15.0     0.0.0.0                0         32768 i
```

Pour simuler un affollement de la route, exécutez la commande **clear ip bgp 192.208.10.6** sur RTD. La table BGP RTB ressemble à ceci :

```
RTB# show ip bgp
BGP table version is 24, local router ID is 203.250.15.2 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 192.208.10.0     192.208.10.5           0             0 300 i
*> 203.250.15.0     0.0.0.0                0         32768 i
```

L'entrée BGP pour 192.208.10.0 est dans un état d'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.

```
RTB# show ip bgp 192.208.10.0
BGP routing table entry for 192.208.10.0 255.255.255.0, version 25
Paths: (1 available, no best path)
300 (history entry)
```

```
    192.208.10.5 from 192.208.10.5 (192.208.10.174)
Origin IGP, metric 0, external
Dampinfo: penalty 910, flapped 1 times in 0:02:03
```

La route a reçu une pénalité pour l'oscillation, mais la pénalité est toujours inférieure à 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 :

```
RTB# show ip bgp
BGP table version is 32, local router ID is 203.250.15.2 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 192.208.10.0     192.208.10.5           0             0  300 i
*> 203.250.15.0     0.0.0.0                0       32768  i

RTB# show ip bgp 192.208.10.0
BGP routing table entry for 192.208.10.0 255.255.255.0, version 32
Paths: (1 available, no best path)
300, (suppressed due to dampening)
192.208.10.5 from 192.208.10.5 (192.208.10.174)
      Origin IGP, metric 0, valid, external
Dampinfo: penalty 2615, flapped 3 times in 0:05:18 , reuse in 0:27:00
```

La route a été atténuée, ou supprimée. 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 regexp regular-expression — Affiche les statistiques de l'oscillation pour tous les chemins qui correspondent à l'expression régulière.

show ip bgp flap-statistics filtre-list list - Affiche les statistiques de l'affollement pour tous les chemins qui passent le filtre.

show ip bgp flap-statistics A.B.C.D m.m.m.m — Affiche les statistiques de l'affollement pour une seule entrée.

show ip bgp flap-statistics A.B.C.D m.m.m.m long-prefix - Affiche les statistiques de l' affollement pour les entrées plus spécifiques.

show ip bgp neighbor [dampened-routes] | [[flap-statistics] — Affiche les statistiques de l'affollement pour tous les chemins depuis un voisin.

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

clear ip bgp flap-statistics regexp regular-expression — Efface les statistiques de l'oscillation pour tous les chemins qui correspondent à l'expression régulière.

clear ip bgp flap-statistics filtre-list list - Efface les statistiques de l'affollement pour tous les chemins qui passent le filtre.

clear ip bgp flap-statistics A.B.C.D m.m.m.m — Efface les statistiques de l'affollement pour une seule entrée.

clear ip bgp A.B.C.D flap-statistics — Efface les statistiques de l'affollement pour tous les chemins à partir d'un voisin.

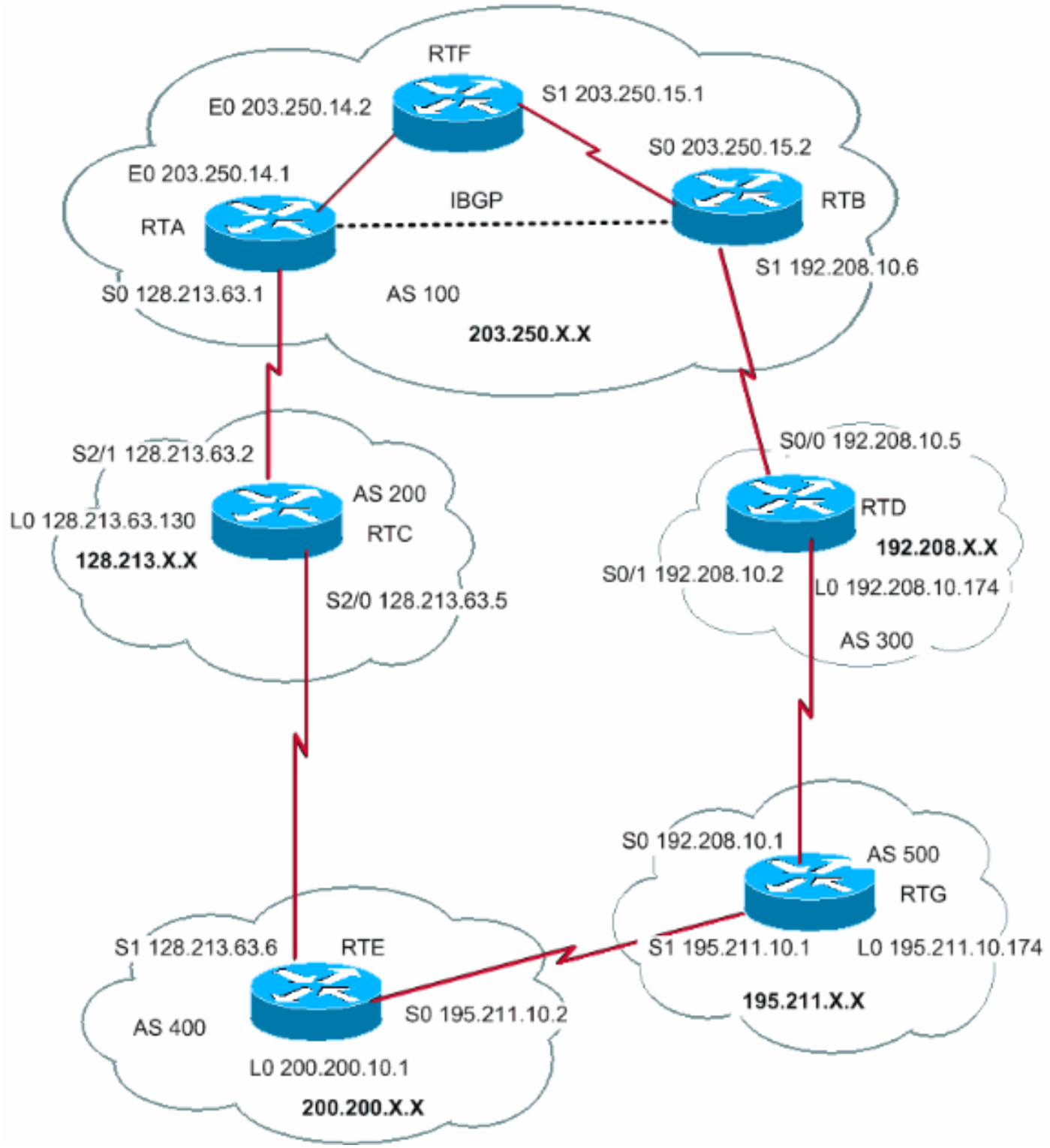
Comment BGP sélectionne un chemin

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.

```
RTB# show ip bgp
BGP table version is 32, local router ID is 203.250.15.2 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 192.208.10.0     192.208.10.5           0             0  300 i
*> 203.250.15.0     0.0.0.0                0       32768  i

RTB# show ip bgp 192.208.10.0
BGP routing table entry for 192.208.10.0 255.255.255.0, version 32
Paths: (1 available, no best path)
300, (suppressed due to dampening)
192.208.10.5 from 192.208.10.5 (192.208.10.174)
      Origin IGP, metric 0, valid, external
Dampinfo: penalty 2615, flapped 3 times in 0:05:18 , reuse in 0:27:00
```

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

```
RTB# show ip bgp BGP
table version is 4, local router ID is 203.250.15.2 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
*i128.213.0.0      128.213.63.2             0    100      0 200 i
*i192.208.10.0     128.213.63.2                  100      0 200 400 500
300 i
*i195.211.10.0     128.213.63.2                  100      0 200 400 500 i
*i200.200.10.0     128.213.63.2                  100      0 200 400 i
*>i203.250.13.0    203.250.13.41            0    100      0 i
*>i203.250.14.0    203.250.13.41            0    100      0 i
*>203.250.15.0     0.0.0.0                  0         32768 i
```

Dans cette table, les notations suivantes apparaissent :

Un i au début indique que l'entrée a été apprise via un homologue iBGP.

Un i à la fin indique que l'origine des informations de chemin est IGP.

Path information — Cette information est intuitive. Par exemple, le réseau 128.213.0.0 est appris par l'intermédiaire du chemin 200 avec un prochain saut de 128.213.63.2.

Remarque: N'importe quelle entrée produite localement, telle que 203.250.15.0, a un prochain saut 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: Notez l'attribut Next Hop. RTB apprend 128.213.0.0 par l'intermédiaire d'un prochain saut de 128.213.63.2, qui est le prochain saut d'eBGP porté dans iBGP.

Regardez la table de routage IP :

```
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

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/75] via 203.250.15.1, 02:50:45, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 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, 128.213.63.2, 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 128.213.63.0 par l'intermédiaire d'OSPF. Vous pouvez exécuter OSPF sur l'interface s0 RTA et le rendre passif ; de cette façon, RTB sait comment atteindre le prochain saut 128.213.63.2. Cette configuration d'RTA est indiquée ici :

```
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

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/75] via 203.250.15.1, 02:50:45, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 02:50:46, Serial0
```

Remarque: Vous pouvez exécuter la commande **bgp nexthopself** entre RTA et RTB afin de changer le prochain saut.

La nouvelle table BGP sur RTB ressemble à ceci :

```
RTB# show ip bgp
BGP table version is 10, local router ID is 203.250.15.2
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
*>i128.213.0.0      128.213.63.2           0    100      0 200 i
*>i192.208.10.0     128.213.63.2                100      0 200 400 500
300 i
*>i195.211.10.0     128.213.63.2                100      0 200 400 500 i
*>i200.200.10.0     128.213.63.2                100      0 200 400 i
*>i203.250.13.0     203.250.13.41          0    100      0 i
```

```
*>i203.250.14.0     203.250.13.41          0    100      0 i
*> 203.250.15.0     0.0.0.0                0         32768 i
```

Remarque: Toutes les entrées ont **>**, ce qui signifie que BGP peut atteindre le prochain saut.

Regardez la table de routage :

```
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

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/75] via 203.250.15.1, 00:04:46, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 00:04:46, Serial0
     128.213.0.0 255.255.255.252 is subnetted, 1 subnets
O       128.213.63.0 [110/138] via 203.250.15.1, 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 128.213.63.0 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: RTF n'a aucune notion des réseaux 192.208.10.0 et 195.211.10.0 parce que vous n'avez pas encore redistribué 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 :

```
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    200.200.10.0 [200/0] via 128.213.63.2, 00:01:07
B    195.211.10.0 [200/0] via 128.213.63.2, 00:01:07
B    192.208.10.0 [200/0] via 128.213.63.2, 00:01:07
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
           [110/75] via 203.250.15.1, 00:12:37, Serial0
B       203.250.13.0 255.255.255.0 [200/0] via 203.250.13.41, 00:01:08
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 00:12:37, Serial0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
B       128.213.0.0 255.255.0.0 [200/0] via 128.213.63.2, 00:01:08
O       128.213.63.0 255.255.255.252
```

```
          [110/138] via 203.250.15.1, 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 :

```
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

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/11] via 203.250.14.1, 00:14:15, Ethernet0
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial1
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 255.255.255.252 is subnetted, 1 subnets
O       128.213.63.0 [110/74] via 203.250.14.1, 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 :

```
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

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/11] via 203.250.14.1, 00:14:15, Ethernet0
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial1
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 255.255.255.252 is subnetted, 1 subnets
O       128.213.63.0 [110/74] via 203.250.14.1, 00:14:15, Ethernet0
```

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 200.200.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
O E2 195.211.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
O E2 192.208.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
          [110/75] via 203.250.15.1, 00:00:15, Serial0
```

```
O E2    203.250.13.0 255.255.255.0
           [110/2000] via 203.250.15.1, 00:00:15, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 2 subnets
C       203.250.15.8 is directly connected, Loopback1
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 00:00:15, Serial0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
O E2    128.213.0.0 255.255.0.0 [110/2000] via 203.250.15.1,
00:00:15,Serial0
O       128.213.63.0 255.255.255.252
           [110/138] via 203.250.15.1, 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 203.250.15.0. 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 203.250.13.0. 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 192.208.10.5 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 192.208.10.5, vous devez aller dans l'autre direction via eBGP. Voici les nouvelles configurations de RTA et de RTB :

```
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 200.200.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
O E2 195.211.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
O E2 192.208.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
           [110/75] via 203.250.15.1, 00:00:15, Serial0
O E2    203.250.13.0 255.255.255.0
           [110/2000] via 203.250.15.1, 00:00:15, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 2 subnets
C       203.250.15.8 is directly connected, Loopback1
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 00:00:15, Serial0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
O E2    128.213.0.0 255.255.0.0 [110/2000] via 203.250.15.1,
00:00:15,Serial0
O       128.213.63.0 255.255.255.252
           [110/138] via 203.250.15.1, 00:00:16, Serial0
```

Les tables BGP ressemblent à ceci :

```
RTA# show ip bgp
BGP table version is 117, local router ID is 203.250.13.41
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
*> 128.213.0.0      128.213.63.2           0             0 200 i
*>i192.208.10.0     192.208.10.5           0    100      0 300 i
*>i195.211.10.0     192.208.10.5                100      0 300 500 i
*                   128.213.63.2                         0 200 400 500 i
*> 200.200.10.0     128.213.63.2                         0 200 400 i
*> 203.250.13.0     0.0.0.0                0         32768 i
*> 203.250.14.0     0.0.0.0                0         32768 i
*>i203.250.15.0     203.250.15.2           0    100      0 i

RTB# show ip bgp
BGP table version is 12, local router ID is 203.250.15.10
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
*>i128.213.0.0      128.213.63.2           0    100      0 200 i
*                   192.208.10.5                         0 300 500 400
200 i
*> 192.208.10.0     192.208.10.5           0             0 300 i
*> 195.211.10.0     192.208.10.5                         0 300 500 i
*>i200.200.10.0     128.213.63.2                100      0 200 400 i
*                   192.208.10.5                         0 300 500 400 i
*>i203.250.13.0     203.250.13.41          0    100      0 i
*>i203.250.14.0     203.250.13.41          0    100      0 i
*> 203.250.15.0     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# show ip bgp
BGP table version is 117, local router ID is 203.250.13.41
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
*> 128.213.0.0      128.213.63.2           0             0 200 i
*>i192.208.10.0     192.208.10.5           0    100      0 300 i
*>i195.211.10.0     192.208.10.5                100      0 300 500 i
*                   128.213.63.2                         0 200 400 500 i
*> 200.200.10.0     128.213.63.2                         0 200 400 i
*> 203.250.13.0     0.0.0.0                0         32768 i
*> 203.250.14.0     0.0.0.0                0         32768 i
*>i203.250.15.0     203.250.15.2           0    100      0 i

RTB# show ip bgp
BGP table version is 12, local router ID is 203.250.15.10
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
*>i128.213.0.0      128.213.63.2           0    100      0 200 i
*                   192.208.10.5                         0 300 500 400
200 i
*> 192.208.10.0     192.208.10.5           0             0 300 i
*> 195.211.10.0     192.208.10.5                         0 300 500 i
*>i200.200.10.0     128.213.63.2                100      0 200 400 i
*                   192.208.10.5                         0 300 500 400 i
*>i203.250.13.0     203.250.13.41          0    100      0 i
*>i203.250.14.0     203.250.13.41          0    100      0 i
*> 203.250.15.0     0.0.0.0                0         32768 i
```

Sur RTA, la préférence locale pour les routes qui viennent d'AS200 est définie à 200. En outre, le réseau 200.200.0.0 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.

```
RTA# show ip bgp
BGP table version is 117, local router ID is 203.250.13.41
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
*> 128.213.0.0      128.213.63.2           0             0 200 i
*>i192.208.10.0     192.208.10.5           0    100      0 300 i
*>i195.211.10.0     192.208.10.5                100      0 300 500 i
*                   128.213.63.2                         0 200 400 500 i
*> 200.200.10.0     128.213.63.2                         0 200 400 i
*> 203.250.13.0     0.0.0.0                0         32768 i
*> 203.250.14.0     0.0.0.0                0         32768 i
*>i203.250.15.0     203.250.15.2           0    100      0 i

RTB# show ip bgp
BGP table version is 12, local router ID is 203.250.15.10
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
*>i128.213.0.0      128.213.63.2           0    100      0 200 i
*                   192.208.10.5                        0 300 500 400
200 i
*> 192.208.10.0     192.208.10.5           0             0 300 i
*> 195.211.10.0     192.208.10.5                        0 300 500 i
*>i200.200.10.0     128.213.63.2                100      0 200 400 i
*                   192.208.10.5                        0 300 500 400 i
*>i203.250.13.0     203.250.13.41          0    100      0 i
*>i203.250.14.0     203.250.13.41          0    100      0 i
*> 203.250.15.0     0.0.0.0                0         32768 i
```

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. Ainsi, RTA est la préférence.

Remarque: Vous avez seulement annoncé 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 :

```
RTB# show ip bgp regexp ^300$
BGP table version is 14, local router ID is 203.250.15.10
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
*> 192.208.10.0     192.208.10.5           0    300      0 300
```

```
RTC#
hostname RTC

ip subnet-zero

interface Loopback0
 ip address 128.213.63.130 255.255.255.192

interface Serial2/0
 ip address 128.213.63.5 255.255.255.252
!
interface Serial2/1
 ip address 128.213.63.2 255.255.255.252

router bgp 200
 network 128.213.0.0
 neighbor 128.213.63.1 remote-as 100
 neighbor 128.213.63.1 distribute-list 1 out
 neighbor 128.213.63.6 remote-as 400

ip classless
access-list 1 deny   195.211.0.0 0.0.255.255
access-list 1 permit any
```

Sur RTC, vous agrégez 128.213.0.0/16 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.

```
RTB# show ip bgp regexp ^300$
BGP table version is 14, local router ID is 203.250.15.10
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
*> 192.208.10.0     192.208.10.5           0    300      0 300

RTC#
hostname RTC

ip subnet-zero

interface Loopback0
 ip address 128.213.63.130 255.255.255.192

interface Serial2/0
 ip address 128.213.63.5 255.255.255.252
!
interface Serial2/1
 ip address 128.213.63.2 255.255.255.252

router bgp 200
 network 128.213.0.0
 neighbor 128.213.63.1 remote-as 100
 neighbor 128.213.63.1 distribute-list 1 out
 neighbor 128.213.63.6 remote-as 400

ip classless
access-list 1 deny   195.211.0.0 0.0.255.255
access-list 1 permit any
```

Une démonstration de l'utilisation du filtrage de communauté se trouve sur RTG. Vous ajoutez un **no-export community** aux mises à jour de 195.211.0.0 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.

```
RTB# show ip bgp regexp ^300$
BGP table version is 14, local router ID is 203.250.15.10
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
*> 192.208.10.0     192.208.10.5           0    300      0 300

RTC#
hostname RTC

ip subnet-zero

interface Loopback0
 ip address 128.213.63.130 255.255.255.192

interface Serial2/0
 ip address 128.213.63.5 255.255.255.252
!
interface Serial2/1
 ip address 128.213.63.2 255.255.255.252

router bgp 200
 network 128.213.0.0
```

```
 neighbor 128.213.63.1 remote-as 100
 neighbor 128.213.63.1 distribute-list 1 out
 neighbor 128.213.63.6 remote-as 400

ip classless
access-list 1 deny   195.211.0.0 0.0.255.255
access-list 1 permit any
```

RTE agrège 200.200.0.0/16. Voici le BGP final et les tables de routage pour RTA, RTF et RTB :

```
RTA# show ip bgp
BGP table version is 21, local router ID is 203.250.13.41
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
*> 128.213.0.0      128.213.63.2           0    200      0 200 i
*>i192.208.10.0     192.208.10.5           0    300      0 300 i
*> 200.200.0.0/16   128.213.63.2                200      0 200 400 i
*> 203.250.13.0     0.0.0.0                0         32768 i
*> 203.250.14.0     0.0.0.0                0         32768 i
*>i203.250.15.0     203.250.15.2           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 128.213.63.2 to network 200.200.0.0

     192.208.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    192.208.10.0 255.255.255.0
           [110/1000] via 203.250.14.2, 00:41:25, Ethernet0
O       192.208.10.4 255.255.255.252
           [110/138] via 203.250.14.2, 00:41:25, Ethernet0
C    203.250.13.0 is directly connected, Loopback0
     203.250.15.0 is variably subnetted, 3 subnets, 3 masks
O       203.250.15.10 255.255.255.255
           [110/75] via 203.250.14.2, 00:41:25, Ethernet0
O       203.250.15.0 255.255.255.252
           [110/74] via 203.250.14.2, 00:41:25, Ethernet0
B       203.250.15.0 255.255.255.0 [200/0] via 203.250.15.2, 00:41:25
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
B       128.213.0.0 255.255.0.0 [20/0] via 128.213.63.2, 00:41:26
C       128.213.63.0 255.255.255.252 is directly connected, Serial0
O*E2 0.0.0.0/0 [110/1000] via 203.250.14.2, Ethernet0/0
B*   200.200.0.0 255.255.0.0 [20/0] via 128.213.63.2, 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 203.250.15.2 to network 0.0.0.0

     192.208.10.0 is variably subnetted, 2 subnets, 2 masks
```

```
O E2    192.208.10.0 255.255.255.0
           [110/1000] via 203.250.15.2, 00:48:50, Serial1
O       192.208.10.4 255.255.255.252
           [110/128] via 203.250.15.2, 01:12:09, Serial1
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
           [110/11] via 203.250.14.1, 01:12:09, Ethernet0
O E2    203.250.13.0 255.255.255.0
           [110/2000] via 203.250.14.1, 01:12:09, Ethernet0
     203.250.15.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.15.10 255.255.255.255
           [110/65] via 203.250.15.2, 01:12:09, Serial1
C       203.250.15.0 255.255.255.252 is directly connected, Serial1
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
O E2    128.213.0.0 255.255.0.0
           [110/2000] via 203.250.14.1, 00:45:01, Ethernet0
O       128.213.63.0 255.255.255.252
           [110/74] via 203.250.14.1, 01:12:11, Ethernet0
O E2 200.200.0.0 255.255.0.0 [110/2000] via 203.250.14.1, 00:03:47,
Ethernet0
O*E2 0.0.0.0 0.0.0.0 [110/1000] via 203.250.15.2, 00:03:33, Serial1
```

Remarque: La table de routage RTF indique que pour atteindre des réseaux locaux à AS300, tels que 192.208.10.0, il faut passer par RTB. Pour atteindre d'autres réseaux connus, tels que 200.200.0.0, 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.

```
RTB# show ip bgp
BGP table version is 14, local router ID is 203.250.15.10
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
*>i128.213.0.0      128.213.63.2           0    200      0 200 i
*> 192.208.10.0     192.208.10.5           0    300      0 300 i
*>i200.200.0.0/16   128.213.63.2                200      0 200 400 i
*>i203.250.13.0     203.250.13.41          0    100      0 i
*>i203.250.14.0     203.250.13.41          0    100      0 i
*> 203.250.15.0     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 192.208.10.5 to network 192.208.10.0

 *   192.208.10.0 is variably subnetted, 2 subnets, 2 masks
B*      192.208.10.0 255.255.255.0 [20/0] via 192.208.10.5, 00:50:46
C       192.208.10.4 255.255.255.252 is directly connected, Serial1
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
           [110/75] via 203.250.15.1, 01:20:33, Serial0
O E2    203.250.13.0 255.255.255.0
           [110/2000] via 203.250.15.1, 01:15:40, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 2 subnets
```

```
C       203.250.15.8 is directly connected, Loopback1
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 01:20:33, Serial0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
O E2    128.213.0.0 255.255.0.0 [110/2000] via 203.250.15.1, 00:46:55, Serial0
O       128.213.63.0 255.255.255.252
           [110/138] via 203.250.15.1, 01:20:34, Serial0
O*E2 0.0.0.0/0 [110/2000] via 203.250.15.1, 00:08:33, Serial0
O E2 200.200.0.0 255.255.0.0 [110/2000] via 203.250.15.1, 00:05:42, Serial0
```

Informations connexes

- BGP : Forum 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
- Configuration du mode redondance de routeur unique et de BGP sur Cat6000 MSFC
- Optimisation du routage et réduction de la consommation de mémoire au niveau des routeurs BGP
- Dépannage de BGP
- Dépannage de l'utilisation élevée du CPU provoquée par le scanner BGP ou le processus du routeur BGP
- Partage de charge avec BGP en environnement mono et multihébergé : Exemples de configuration
- Page de support BGP
- Support et documentation techniques - Cisco Systems