Border Gateway Protocol-gebruiksscenario's onderzoeken

Inhoud

Inleiding

In dit document worden vijf gebruiksscenario's van het Border Gateway Protocol (BGP) beschreven.

Voorwaarden

Vereisten

Er zijn geen specifieke vereisten van toepassing op dit document.

Gebruikte componenten

Dit document is niet beperkt tot specifieke software- en hardware-versies.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Conventies

Raadpleeg Cisco Technical Tips Conventions (Conventies voor technische tips van Cisco) voor

meer informatie over documentconventies.

BGP-casestudy’s 1

Met het BGP, dat in RFC 1771 is gedefinieerd, kunt u lusvrije interdomeinrouting tussen autonome systemen (AS) realiseren. Een AS is een set routers die onder één technisch beheersysteem vallen. Routers in een AS kunnen meerdere IGP’s (Interior Gateway Protocol) gebruiken om routinginformatie binnen het AS uit te wisselen. De routers kunnen een extern gatewayprotocol gebruiken om pakketten buiten het AS te routeren.

Hoe werkt BGP?

BGP gebruikt TCP als transportprotocol op poort 179. Twee BGP-routers vormen een TCP-verbinding tussen elkaar. Deze routers zijn peerrouters. De peerrouters wisselen berichten uit om de verbindingsparameters te openen en te bevestigen.

BGP-routers wisselen informatie over netwerkbereikbaarheid uit. Deze informatie is hoofdzakelijk een indicatie van de volledige paden die een route moet nemen om het bestemmingsnetwerk te bereiken. De paden zijn BGP AS-nummers. Deze informatie helpt bij het maken van een grafiek van autonome systemen die lusvrij zijn. De grafiek toont tevens waar routingbeleid kan worden toegepast om beperkingen in te stellen voor het routinggedrag.

Twee routers die een TCP-verbinding hebben voor het uitwisselen van BGP-routinginformatie worden ‘peers’ of ‘neighbors’ genoemd. BGP-peers wisselen eerst de complete BGP-routingtabellen uit. Na deze uitwisseling, verzenden de peers incrementele updates wanneer de routingtabel verandert. BGP houdt een versienummer van de BGP-tabel bij. Het versienummer is hetzelfde voor alle BGP-peers. Het versienummer verandert wanneer BGP de tabel updatet met gewijzigde routinginformatie. Het verzenden van keepalive-pakketten houdt de verbinding tussen BGP-peers actief. Als reactie op fouten of speciale omstandigheden worden pakketten met meldingen verzonden.

eBGP en iBGP

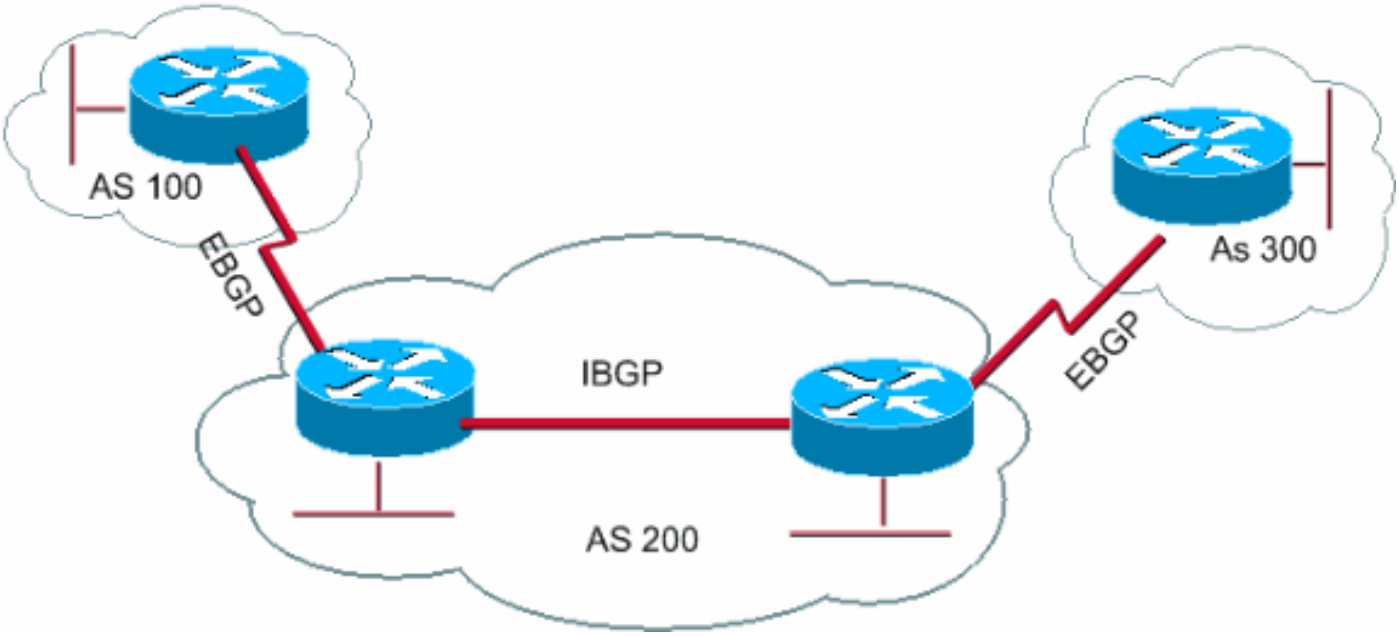
Wanneer een AS meerdere BGP-speakers heeft, kan het AS dienen als transitservice voor andere autonome systemen. Zoals in het volgende diagram in deze paragraaf wordt getoond, is AS200 een transit AS voor AS100 en AS300.

Om de informatie naar externe autonome systemen te kunnen sturen, moet de bereikbaarheid van netwerken worden verzekerd. Om de bereikbaarheid van netwerken te verzekeren, vinden de volgende processen plaats:

- Interne BGP-peering (iBGP) tussen routers binnen een AS
- Herdistributie van BGP-informatie naar IGP’s die in het AS worden uitgevoerd

Wanneer het BGP wordt uitgevoerd tussen routers van twee verschillende autonome systemen, dan wordt dit een extern BGP (eBGP) genoemd. Wanneer het BGP tussen routers in hetzelfde AS

wordt uitgevoerd, wordt dit iBGP genoemd.



BGP draait tussen routers op hetzelfde AS

BGP-routing inschakelen

Voer de volgende stappen uit om BGP in te schakelen en te configureren.

Laten we aannemen dat u twee routers, RTA en RTB, wilt laten communiceren via BGP. In het eerste voorbeeld bevinden RTA en RTB zich in verschillende autonome systemen. In het tweede voorbeeld behoren beide routers tot hetzelfde AS.

1. Definieer het routerproces en het AS-nummer waartoe de routers behoren.

   Voer de volgende opdracht uit om BGP in te schakelen op een router:

   <#root>

   router bgp <autonomous-system>

   ```
   RTA#
   router bgp 100

   RTB#
   router bgp 200
   ```

   Deze verklaringen wijzen erop dat RTA BGP in werking stelt en tot AS100 behoort. RTB beheert BGP en behoort tot AS200.

2. Definieer BGP-neighbors.

   De formatie van BGP-neighbors geeft aan welke routers via BGP proberen te communiceren. In de volgende paragraaf wordt dit proces toegelicht.

BGP-neighbors vormen

Twee BGP-routers worden neighbors nadat de routers een TCP-verbinding met elkaar hebben gemaakt. De TCP-verbinding is essentieel zodat de twee peerrouters kunnen starten met het uitwisselen van routingupdates.

Nadat de TCP-verbinding is gemaakt, versturen de routers open berichten voor het uitwisselen van waarden. De waarden die de routers uitwisselen omvatten het AS-nummer, de BGP-versie die de routers uitvoeren, de BGP-router-ID en de keepalive-wachtstandtijd. Nadat deze waarden zijn bevestigd en geaccepteerd, wordt de verbinding met de neighbor tot stand gebracht. Statussen anders dan Established zijn een indicatie dat de twee routers geen neighbors zijn geworden en dat de routers geen BGP-updates kunnen uitwisselen.

Geef deze neighbor opdracht uit om een TCP-verbinding tot stand te brengen:

<#root>

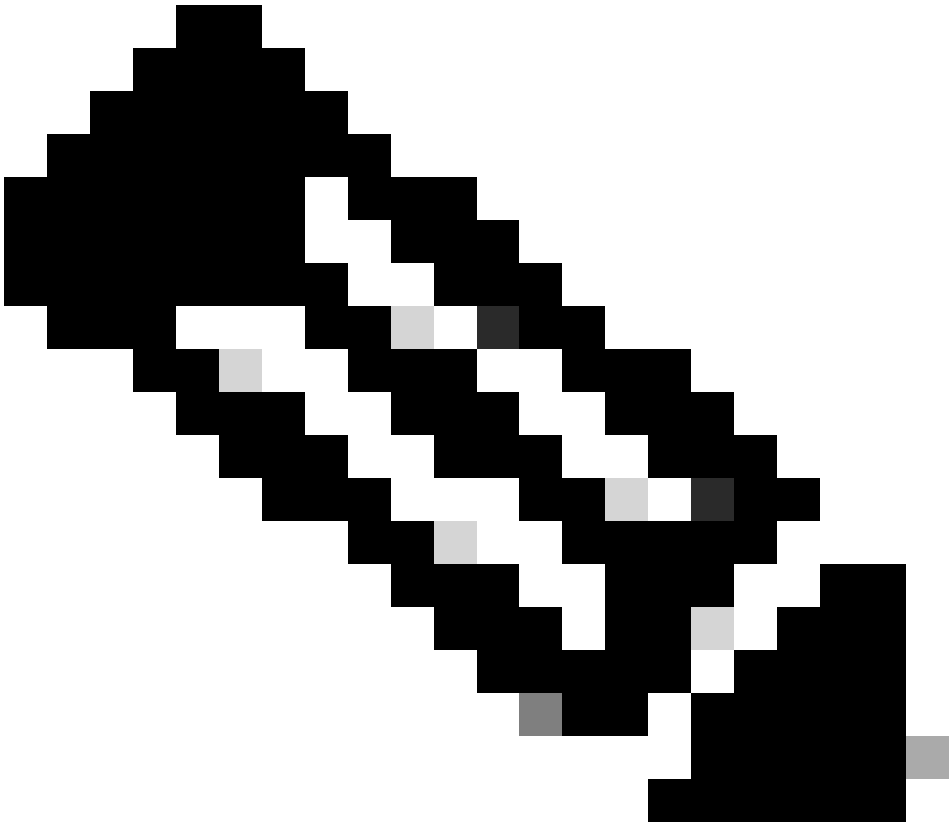
neighbor <ip-address> remote-as <number>

Het nummer in de opdracht is het AS-nummer van de router waarmee u verbinding wilt maken met BGP. Het ip-adres is het adres van de volgende hop met directe verbinding voor eBGP. Voor iBGP is ip-adres elk IP-adres op de andere router.

De twee IP-adressen die u in de opdracht van neighbor de peer routers gebruikt, *moeten* elkaar kunnen bereiken. Eén manier om de bereikbaarheid te controleren is een uitgebreide ping tussen de twee IP-adressen. Uitgebreid pingelt krachten de pingende router om als bron het IP adres te gebruiken dat het neighbor bevel specificeert. De router moet dit adres gebruiken in plaats van het IP-adres van de interface waarvan het pakket afkomstig is.

Als er BGP-configuratiewijzigingen zijn, moet de neighbor-verbinding worden gereset om de nieuwe parameters van kracht te laten worden. .

- clear ip bgp address



Opmerking: het adres is het buuradres

•

clear ip bgp *

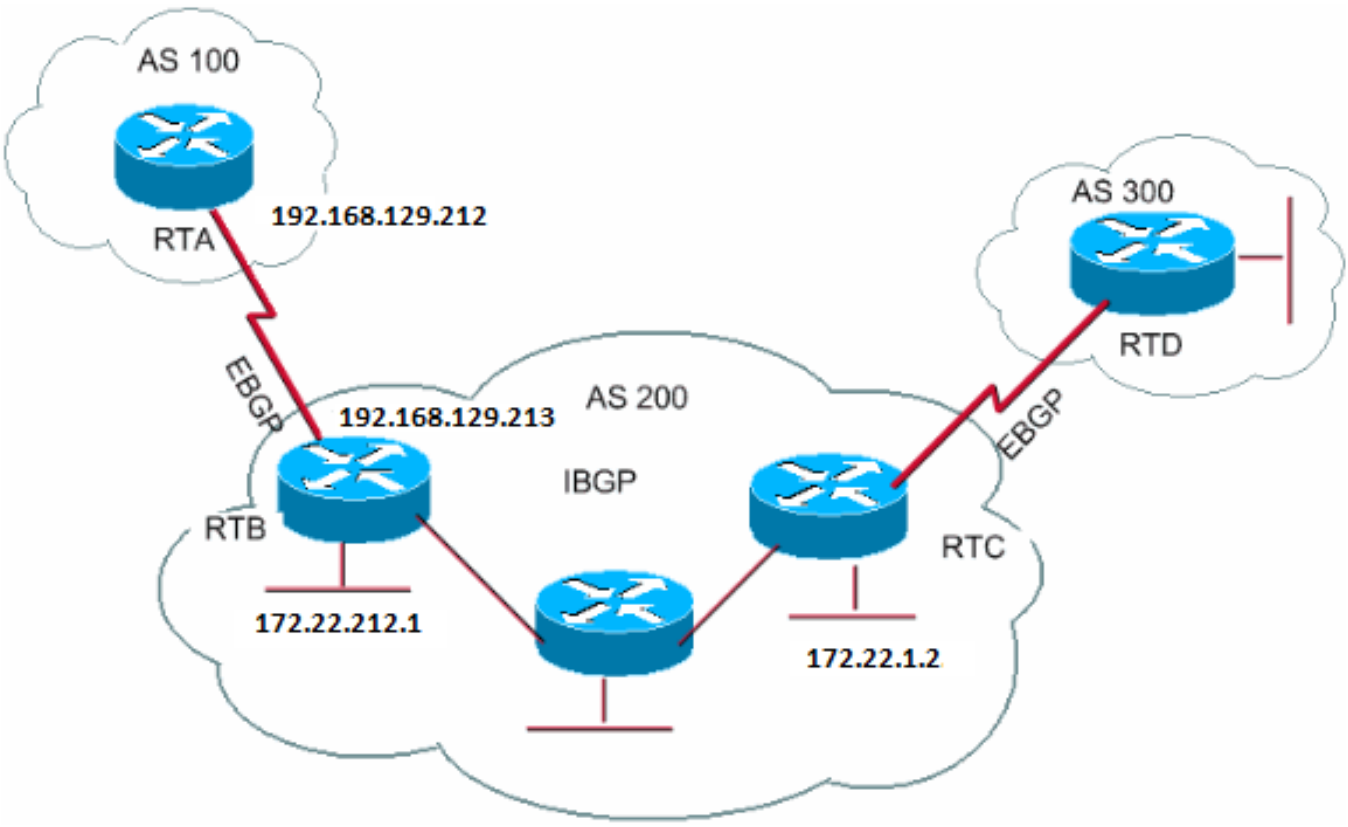
Deze opdracht wist alle neighbor-verbindingen.

Standaard beginnen BGP-sessies met het gebruik van BGP-versie 4, indien nodig wordt vervolgens omlaag onderhandeld naar eerdere versies. U kunt onderhandelingen voorkomen en eisen stellen aan de BGP-versie die de routers gebruiken om met een neighbor te communiceren. Voer deze opdracht uit in de modus voor routerconfiguratie:

<#root>

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

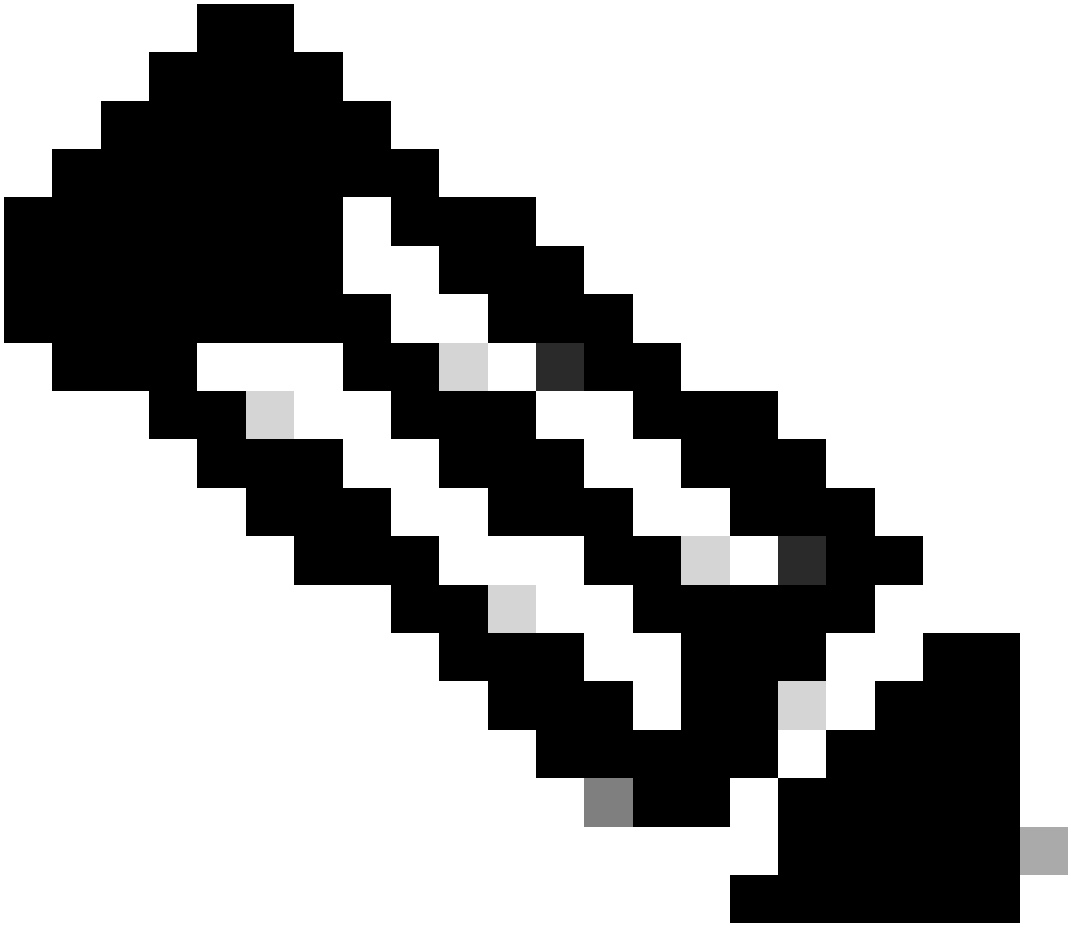
Hier is een voorbeeld van de neighbor opdrachtconfiguratie:



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

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

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

In dit voorbeeld gebruiken RTA en RTB eBGP. RTB en RTC gebruiken iBGP. Het externe AS-nummer wijst naar een extern of een intern AS, en duidt op eBGP of iBGP. Ook hebben de eBGP-peers een directe verbinding, maar de iBGP-peers hebben geen directe verbinding. iBGP-routers hoeven geen directe verbinding te hebben. Maar er moet wel een IGP zijn die draait en de twee buren in staat stelt elkaar te bereiken.

Deze sectie bevat een voorbeeld van de informatie die door de opdracht show ip bgp neighbors wordt weergegeven.



Opmerking: Let vooral op de BGP-status. Elke staat die niet is ingesteld, geeft aan dat de peers niet omhoog zijn. Let ook op deze volgende items:

- De BGP-versie is 4
- De `externe router-ID`

  Dit getal is het hoogste IP-adres op de router of de hoogste loopback-interface, indien aanwezig.
- De `tabelversie`

  De `tabelversie geeft de status van de tabel aan.` Steeds wanneer er nieuwe informatie binnenkomt, wordt de versie van de tabel verhoogd. Een versie die blijft stijgen, duidt op routefluctuatie waardoor het bijwerken van routes voortdurend blijft doorgaan.

<#root>

```
Router#
```

show ip bgp neighbors

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

BGP state = Established

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

BGP- en loopback-interfaces

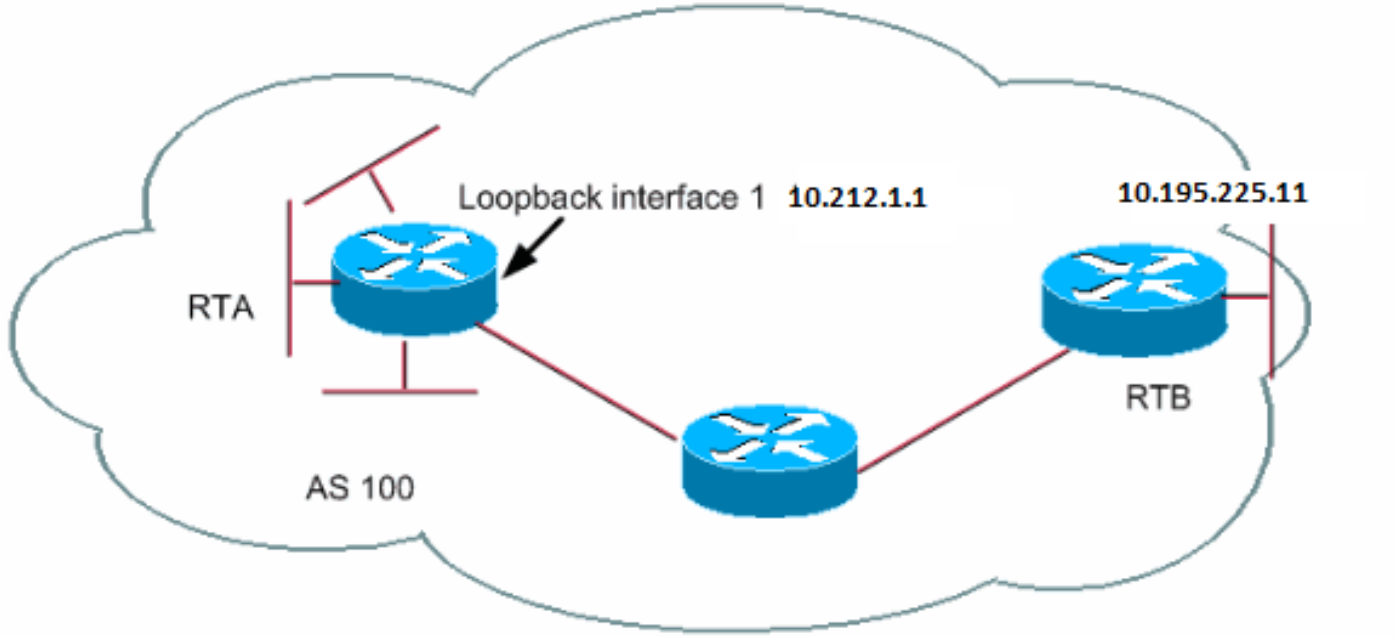
Het gebruik van een loopback interface om buren te definiëren is gebruikelijk bij iBGP, maar niet bij eBGP. Normaal gesproken gebruikt u de loopback-interface om ervoor te zorgen dat het IP-adres van de neighbor actief blijft en onafhankelijk is van hardware die goed functioneert. In het geval van eBGP hebben peerrouters dikwijls een directe verbinding en is loopback niet van toepassing.

Als u het IP-adres van een loopback-interface in de neighbor opdracht gebruikt, hebt u extra configuratie nodig op de buurrouter. De neighbor-router moet BGP informeren over het gebruik van een loopback-interface in plaats van een fysieke interface om de TCP-verbinding naar de BGP-neighbor te maken. Gebruik de volgende opdracht om een loopback-interface aan te duiden:

<#root>

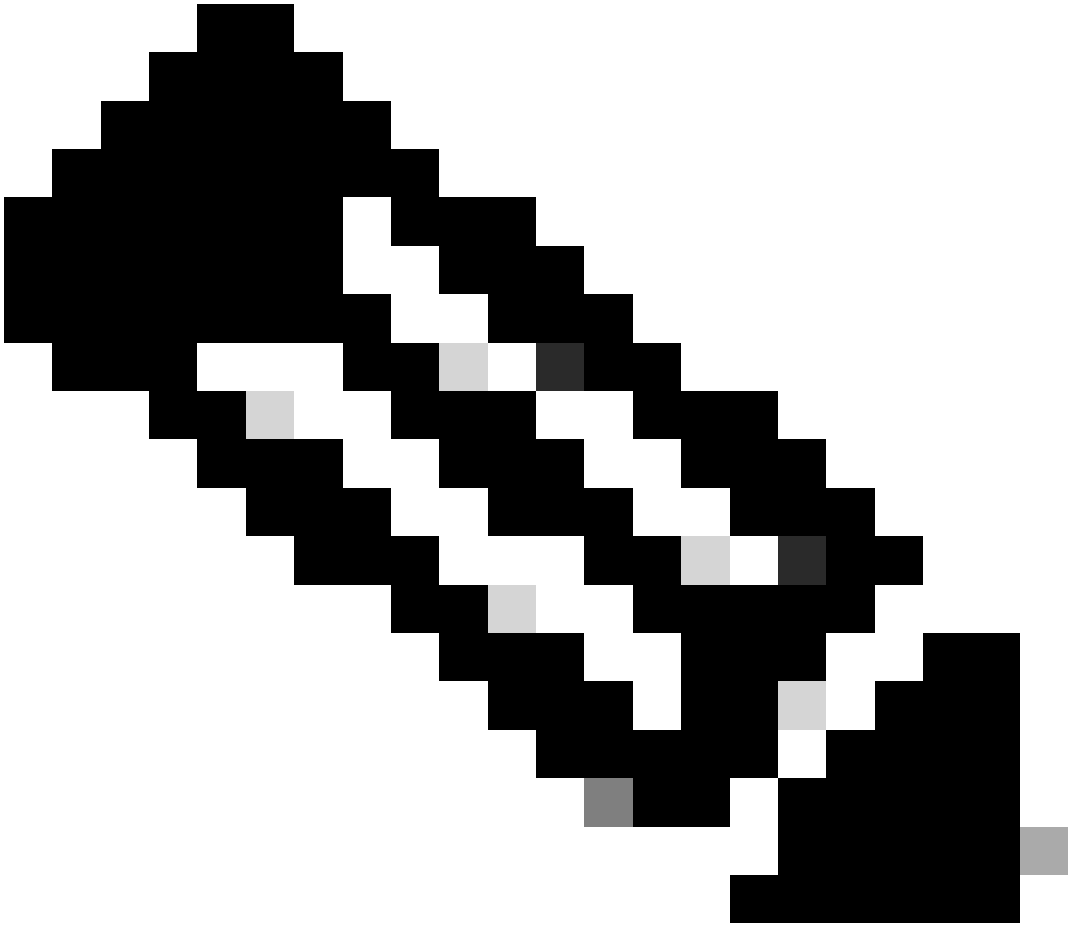
neighbor <ip-address> update-source <interface>

Dit voorbeeld illustreert het gebruik van deze opdracht:




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

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

In dit voorbeeld draaien RTA en RTB iBGP binnen AS100. In de neighbor opdracht gebruikt RTB de loopback-interface van RTA, 10.212.1.1. In dit geval moet RTA BGP dwingen om het loopback IP-adres als bron in de TCP-buurverbinding te gebruiken. Om deze actie af te dwingen, voegt RTA toe **update-source interface-type interface-number** zodat de opdracht is neighbor 10.195.225.11 update-source loopback 1. Deze verklaring dwingt BGP om het IP-adres van de loopback-interface te gebruiken wanneer BGP met buur 10.195.225.11 praat.

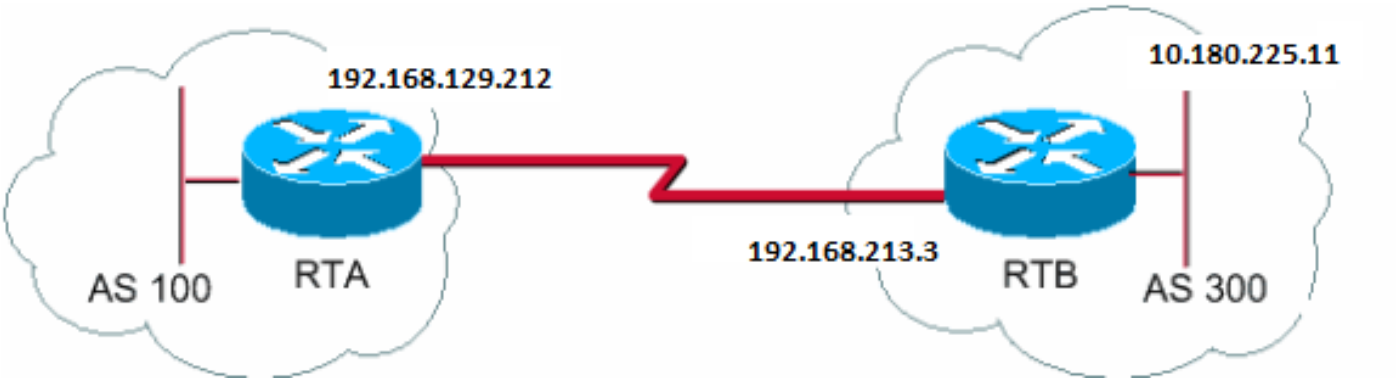


Opmerking: RTA heeft het IP-adres van de fysieke interface van RTB, 10.195.225.11, gebruikt als neighbor. Door dit IP-adres te gebruiken heeft RTB geen speciale configuratie nodig. Raadpleeg Voorbeeldconfiguratie voor iBGP en eBGP met of zonder loopback-adres voor een volledige voorbeeldconfiguratie van een netwerkscenario.

---

eBGP-multihop

In sommige gevallen kan een Cisco-router eBGP uitvoeren met een router van derden die geen directe verbinding van de twee externe peers toestaat. Om de verbinding tot stand te brengen, kunt u eBGP-multihop gebruiken. De eBGP-multihop maakt een neighbor-verbinding tussen twee externe peers mogelijk die geen directe verbinding hebben. De multihop is alleen voor eBGP en niet voor iBGP. Dit voorbeeld illustreert eBGP-multihop:

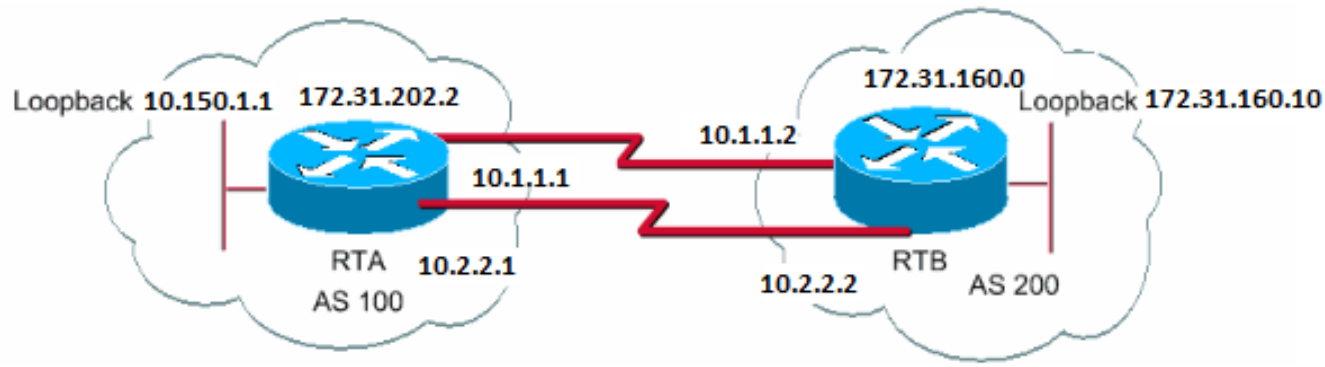



```
RTA#
router bgp 100
 neighbor 10.180.225.11 remote-as 300
 neighbor 10.180.225.11 ebgp-multihop

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

RTA geeft een externe neighbor aan zonder directe verbinding. RTA moet het gebruik van de opdracht neighbor ebgp-multihop aangeven. Aan de andere kant geeft RTB een buur aan die een directe verbinding heeft, namelijk 192.168.129.212. Vanwege deze directe verbinding heeft RTB geen neighbor ebgp-multihop opdracht nodig. U moet ook een IGP of statische routing configureren om de buren zonder verbinding elkaar te laten bereiken.

Het voorbeeld in de sectie BGP multihop (taakverdeling) toont hoe u taakverdeling met BGP kunt bereiken in een geval waarin u eBGP over parallelle lijnen hebt.

eBGP-multihop (taakverdeling)




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

router bgp 100
```

```
 neighbor 172.31.160.10 remote-as 200
 neighbor 172.31.160.10 ebgp-multihop
 neighbor 172.31.160.10 update-source loopback 0
 network 172.31.202.2

ip route 172.31.160.0 255.255.0.0 10.1.1.2
ip route 172.31.160.0 255.255.0.0 10.2.2.2

RTB#
int loopback 0
 ip address 172.31.160.10 255.255.255.0

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

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

Dit voorbeeld illustreert het gebruik van loopback interfaces, update-source, en ebgp-multihop. Het voorbeeld is een tijdelijke oplossing om een taakverdeling mogelijk te maken tussen twee eBGP-speakers via parallelle seriële lijnen. Normaliter kiest BGP een van de lijnen voor het verzenden van pakketten en vindt er geen taakverdeling plaats. Met de introductie van loopback-interfaces is de volgende hop voor eBGP de loopback-interface. Statische routes, of een IGP, worden gebruikt om twee gelijkwaardige paden te kunnen gebruiken om de bestemming te bereiken. RTA heeft twee keuzes om de volgende hop 172.31.160.10 te bereiken: één pad via 10.1.1.2 en het andere pad via 10.2.2.2. RTB heeft dezelfde keuzes.

Routekaarten

BGP maakt veel gebruik van routekaarten. In de BGP-context is de routekaart een methode om routinginformatie te beheren en aan te passen. Het beheren en wijzigen van routinginformatie vindt plaats via het definiëren van voorwaarden voor de herdistributie van routes van het ene routingprotocol naar het andere. Het beheren van routinginformatie kan plaatsvinden bij het invoegen in en verwijderen uit BGP. Zo is het formaat van de routekaart:

<#root>

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

De kaarttag is simpelweg een naam die u aan de routekaart geeft. U kunt meerdere instanties van dezelfde routekaart of dezelfde naamtag definiëren. Het volgnummer geeft de positie aan die een nieuwe routekaart krijgt in de lijst met routekaarten die u al heeft geconfigureerd met dezelfde naam.

In dit voorbeeld zijn twee instanties van de routekaart gedefinieerd met de naam MYMAP. De eerste instantie heeft volgnummer 10 en de tweede heeft volgnummer 20.

- **route-map MYMAP permit 10** (Hier komt de eerste set voorwaarden.)

- **route-map MYMAP permit 20** (Hier komt de tweede set voorwaarden.)

Wanneer u routekaart MYMAP op inkomende of uitgaande routes toepast, wordt de eerste reeks voorwaarden via instantie 10 toegepast. Als niet aan de eerste reeks voorwaarden wordt voldaan, gaat u naar een hoger geval van de routekaart.

Configuratieopdrachten match en set

Elke routekaart bestaat uit een lijst van match en set configuratiebevelen. De overeenkomst specificeert een match criteria en een reeks specificeert een set actie als de criteria die het match bevel afdwingt worden voldaan aan.

Zo kunt u bijvoorbeeld een routekaart definiëren die uitgaande updates controleert. Als er een overeenkomst voor IP-adres 10.1.1.1 is, wordt de metriek voor die update ingesteld op 5. Deze opdrachten illustreren het voorbeeld:

```
<#root>

match ip address 10.1.1.1

set metric 5
```

Nu, als aan de matchcriteria wordt voldaan en u een permit hebt, is er een herverdeling of controle van de routes, zoals de vastgestelde actie specificeert. De lijst wordt onderbroken.

Als aan de matchcriteria wordt voldaan en u een deny hebt, is er geen herverdeling of controle van de route. De lijst wordt onderbroken.

Als niet aan de matchcriteria wordt voldaan en u een permit of een deny hebt, wordt de volgende instantie van de routekaart gecontroleerd. Instantie 20 wordt bijvoorbeeld gecontroleerd. Deze controle van de volgende instantie gaat door tot het proces wordt onderbroken of totdat alle instanties van de routekaart zijn voltooid. Als u de lijst zonder een gelijke beëindigt, is de route not accepted nor forwarded.

In Cisco IOS®-softwarereleases eerder dan Cisco IOS-softwarerelease 11.2 kunt u, wanneer u routekaarten gebruikt om BGP-updates te filteren in plaats van ze tussen protocollen te herverdelen, niet op de inkomende route filteren wanneer u een **match-**opdracht op het IP-adres gebruikt. Een filter voor uitgaand verkeer is toegestaan. Cisco IOS-softwarerelease 11.2 en latere releases hebben deze beperking niet.

De verwante opdrachten voor match zijn:

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

- matchmetric

- match route-type

- match tag

De verwante opdrachten voor set zijn:

- set as-path

- set clns

- set automatic-tag

- set community

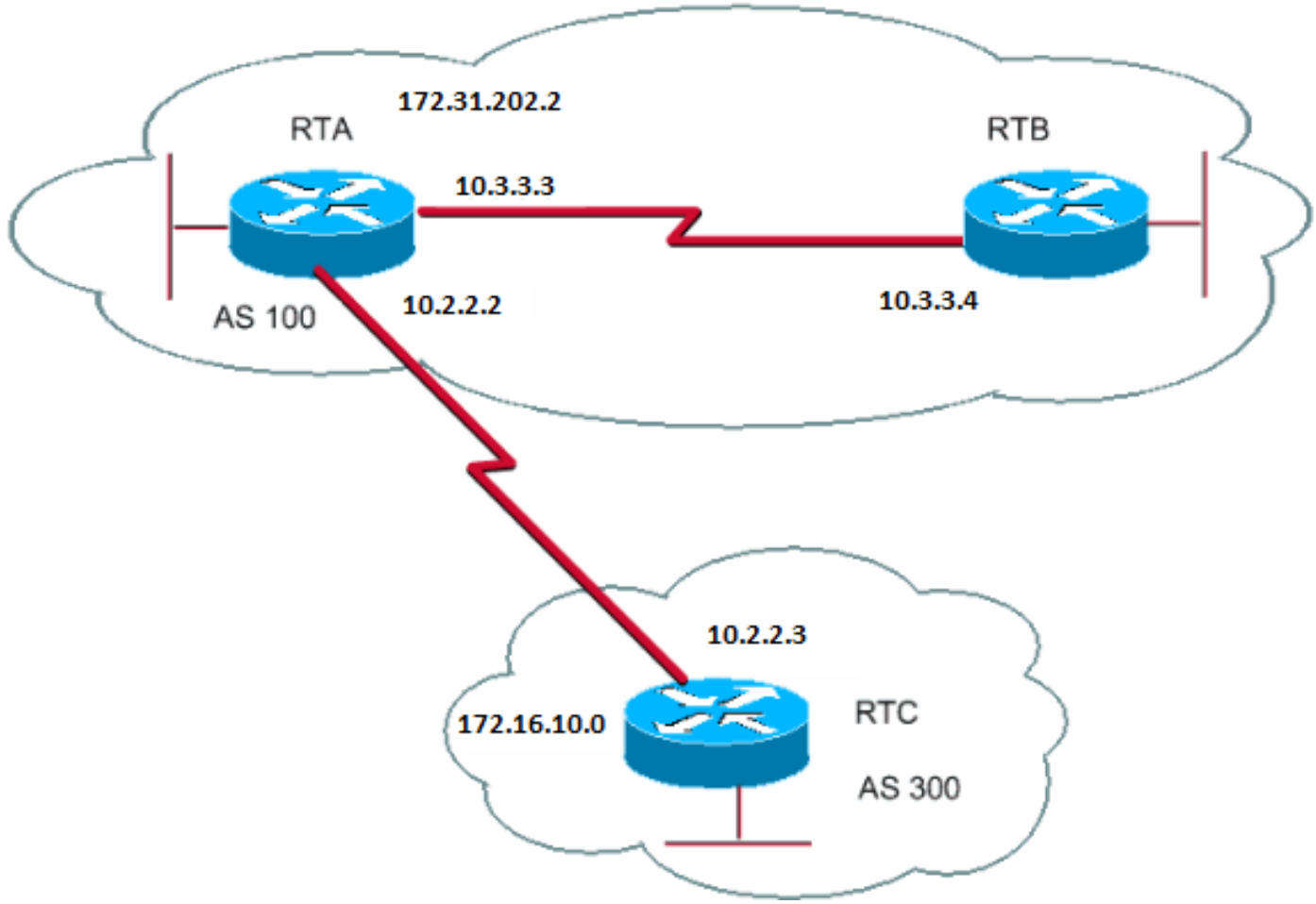
- set interface

- set default interface

- set ip default nexthop

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

Kijk eens naar een aantal voorbeelden van routekaarten:



Voorbeelden routekaart

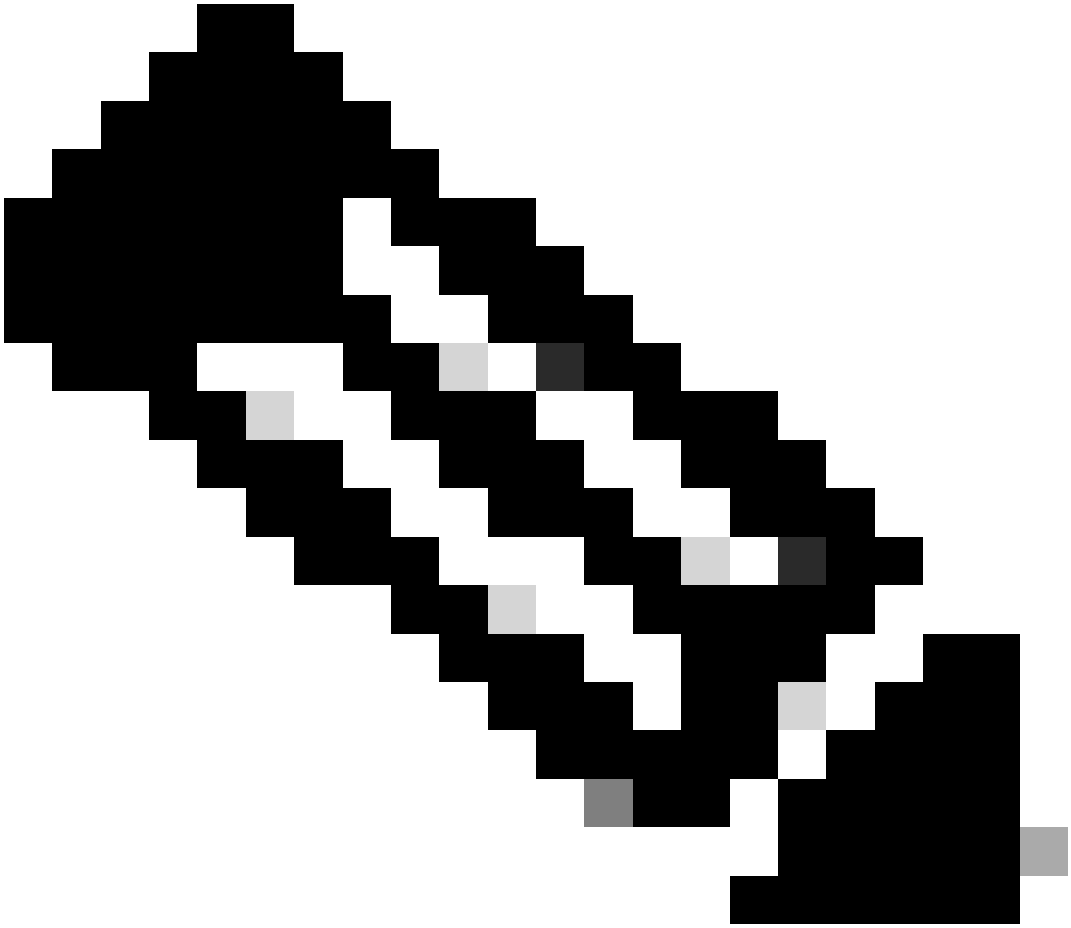
Voorbeeld 1

Stel dat RTA en RTB het Routing Information Protocol (RIP) uitvoeren, en RTA en RTC voeren BGP uit. RTA krijgt updates via BGP en herdistribueert de updates naar RIP. Stel dat RTA wil herverdelen naar RTB routes ongeveer 172.16.10.0 met een metriek van 2 en alle andere routes met een metriek van 5. In dit geval kunt u deze configuratie gebruiken:

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

router bgp 100
 neighbor 10.2.2.3 remote-as 300
 network 172.31.202.2

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

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

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

In dit voorbeeld, als een route het IP adres 172.16.10.0 aanpast, heeft de route een metriek van 2. Dan, breek je uit de routekaartlijst. Als er geen gelijke is, gaat u onderaan de routekaartlijst te werk, die op alles wijst anders wordt geplaatst aan metrische 5.



Opmerking: Stel altijd de vraag "Wat gebeurt er met routes die niet overeenkomen met een van de match-instructies?" Deze routes worden standaard afgewezen.

Voorbeeld 2

Stel dat u in Voorbeeld 1 niet wilt dat AS100 updates over 172.16.10.0 accepteert. U kunt routekaarten niet op de inkomende route toepassen wanneer u met een IP-adres als basis past. Daarom moet er een uitgaande routekaart worden gebruikt op RTC:

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

route-map STOPUPDATES permit 10
 match ip address 1

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

Nu u meer weet over het starten van BGP en het definiëren van een neighbor, kunnen we kijken naar het starten van de uitwisseling van netwerkinformatie.

Er zijn meerdere manieren om netwerkinformatie te verzenden met behulp van BGP. In deze secties worden de methoden één voor één behandeld:

- Opdracht network
- Herdistributie
- Statische routes en herdistributie

Opdracht network

Het formaat van het network bevel is:

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

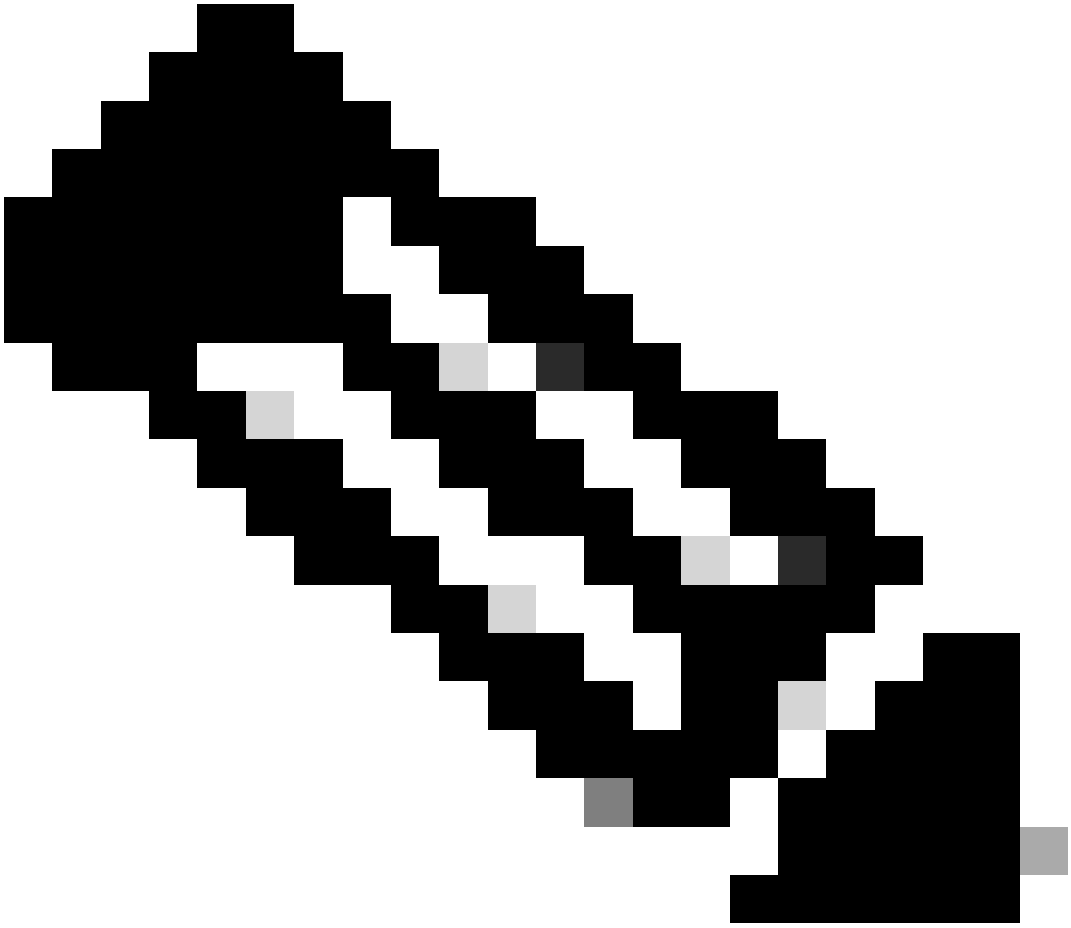
De network opdracht regelt de netwerken die uit dit vak voortkomen. Dit concept verschilt van de bekende configuratie met Interior Gateway Routing Protocol (IGRP) en RIP. Met deze opdracht, probeert u BGP niet op een specifieke interface te starten. In plaats daarvan probeert u aan BGP aan te geven welke netwerken BGP uit dit vak moet voortkomen. De opdracht gebruikt een mask-gedeelte omdat BGP-versie 4 (BGP4) subnetten en supernetten ondersteunt. Een maximum van 200 ingangen van het network bevel zijn aanvaardbaar.

Het network bevel werkt als de router het netwerk kent dat u probeert te adverteren, hetzij verbonden, statisch, of dynamisch geleerd.

Een voorbeeld van de opdracht network is:

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

ip route 192.168.213.0 255.255.0.0 null 0
```

Dit voorbeeld geeft aan dat router A een netwerkingang voor 192.168.213.0/16 genereert. De /16 geeft aan dat u een supernet van het klasse C-adres gebruikt en de eerste twee octetten of de eerste 16 bits adverteert.

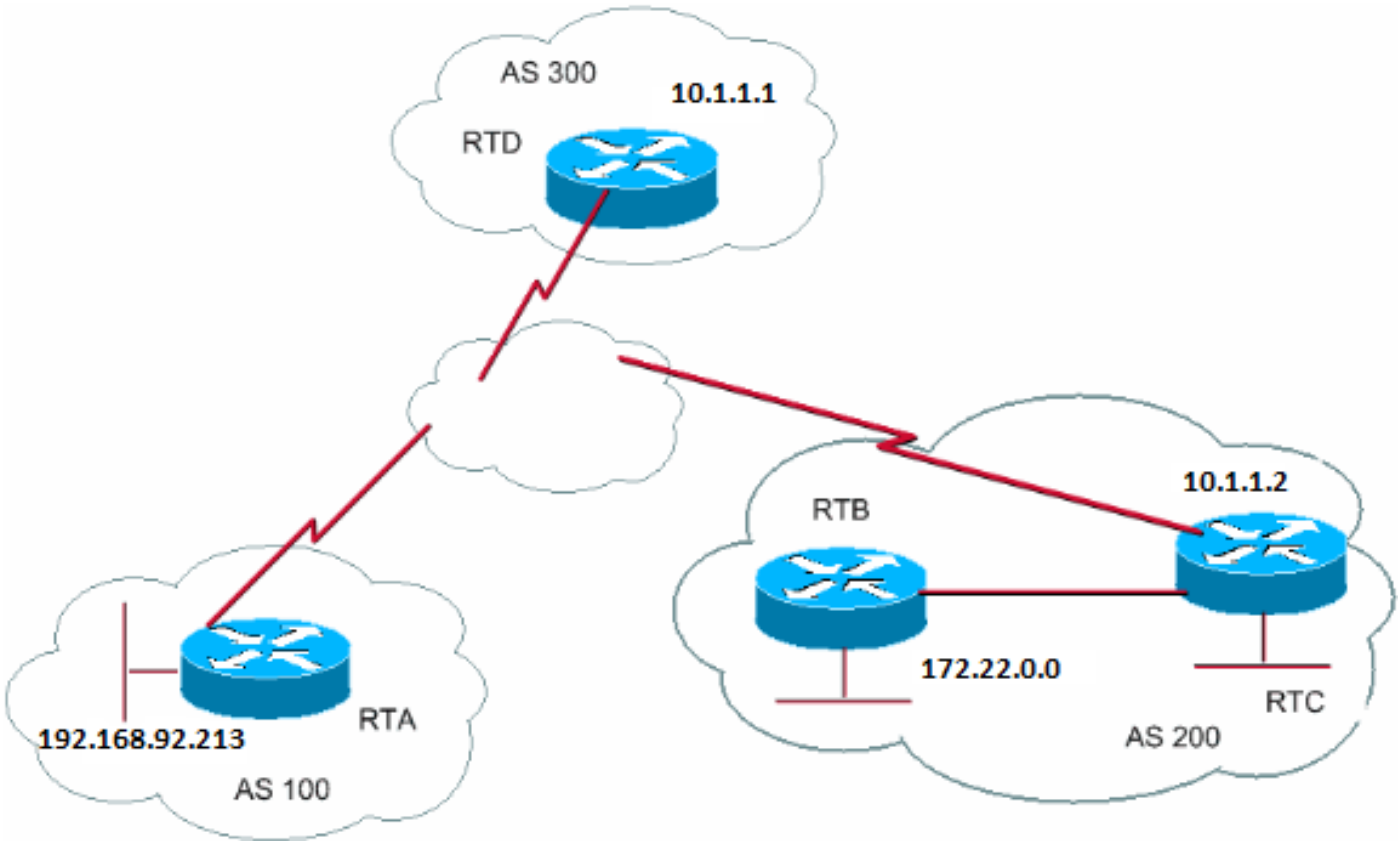


Opmerking: U heeft de statische route nodig om de router 192.168.213.0 te laten genereren, omdat de statische route een overeenkomende vermelding in de routingtabel plaatst.

Herdistributie

De network opdracht is een manier om uw netwerken te adverteren via BGP. Een andere manier is door uw IGP te herdistribueren in BGP. Uw IGP kan een IGRP, OSPF-protocol (Open Shortest Path First), RIP, Enhanced Interior Gateway Routing Protocol (EIGRP) of een ander protocol zijn. Deze herverdeling kan eng lijken omdat u nu al uw interne routes in BGP dumpen; sommige van deze routes kunnen via BGP geleerd zijn en u hoeft ze niet opnieuw uit te sturen. Wees voorzichtig als je filtert om ervoor te zorgen dat je naar de internet-alleen routes die je wilt adverteren en niet naar alle routes die je hebt. Hierna volgt een voorbeeld.

RTA kondigt 192.168.92.213 aan en RTC kondigt 172.22.0.0 aan. Bekijk de RTC-configuratie:



Als u de networkopdracht geeft, hebt u:

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

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

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

Als u in plaats daarvan herdistributie gebruikt, heeft u:

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

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

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

Deze herdistributie veroorzaakt de oorsprong van 192.168.92.213 door uw AS. U bent niet de bron van 192.168.92.213; AS100 is de bron. Je moet dus filters gebruiken om te voorkomen dat de bron van dat netwerk via je AS terechtkomt. De juiste configuratie is:

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

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

access-list 1 permit 172.22.0.0 0.0.255.255
```

U gebruikt de access-listopdracht om de netwerken te besturen die afkomstig zijn van AS200.

Herdistributie van OSPF in BGP verschilt van herdistributie voor andere IGP's. Het simpele probleem van redistribute ospf 1ondervoeding werkt router bgpniet. Specifieke trefwoorden zoals internal, external, en **nssa-external** zijn nodig om de respectievelijke routes opnieuw te verdelen. Zie de herverdeling van OSPF-routers in BGP begrijpen voor meer informatie.

Statische routes en herdistributie

U kunt altijd statische routes gebruiken om een netwerk of subnet te maken. Het enige verschil is dat deze routes volgens BGP een oorsprong hebben die onvolledig of onbekend is. U kunt hetzelfde resultaat bereiken dat het voorbeeld in de herverdelingssectie met dit heeft bereikt:

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

router bgp 200
 neighbor 10.1.1.1 remote-as 300
 redistribute static

ip route 172.22.0.0 255.255.255.0 null0
```

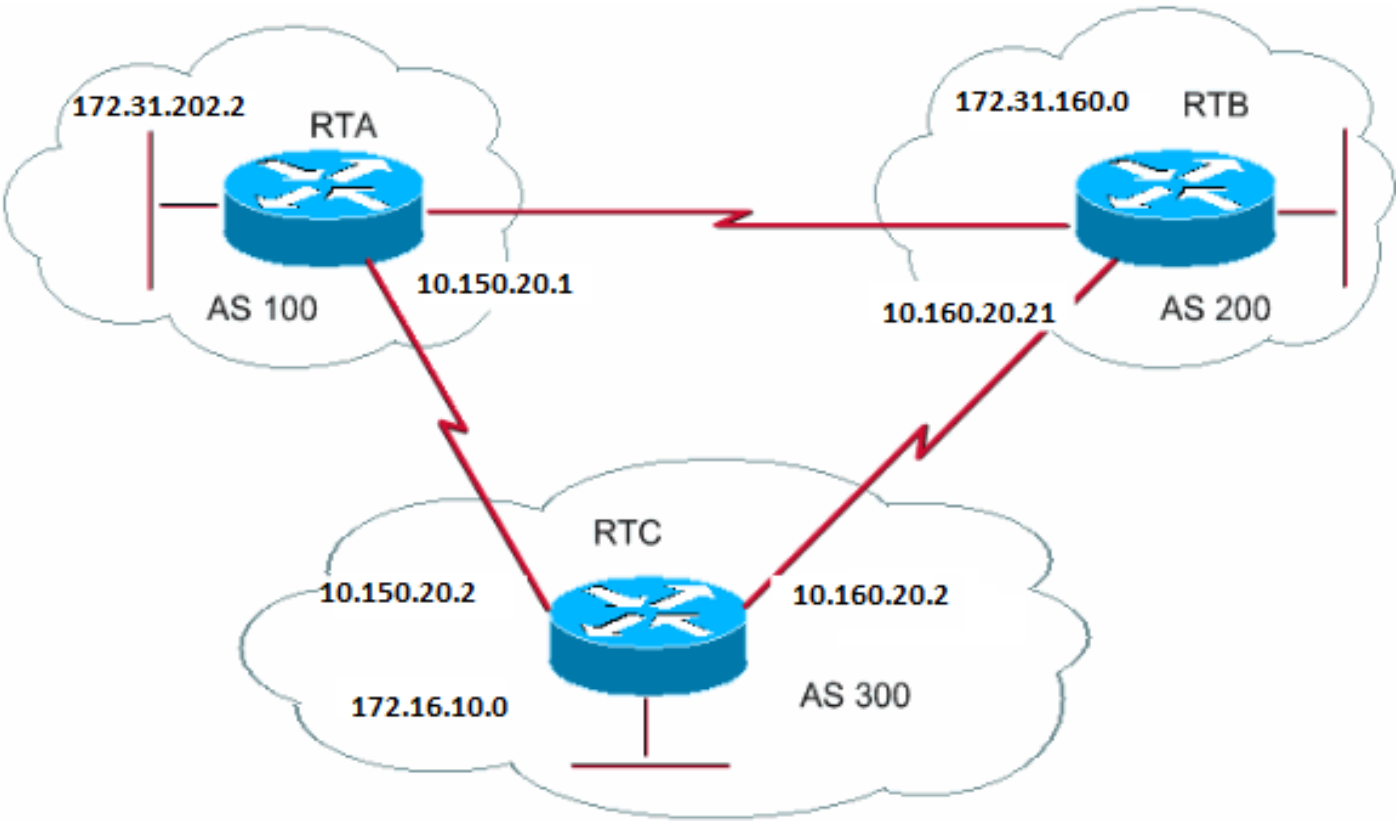
De null0 interface betekent het pakket negeren. Zo, als u het pakket krijgt en er een specifiekere gelijke dan 172.22.0.0 is, die bestaat, verzendt

de router het pakket naar de specifieke gelijke. Anders negeert de router het pakket. Deze methode is een goede manier om een supernet aan te kondigen.

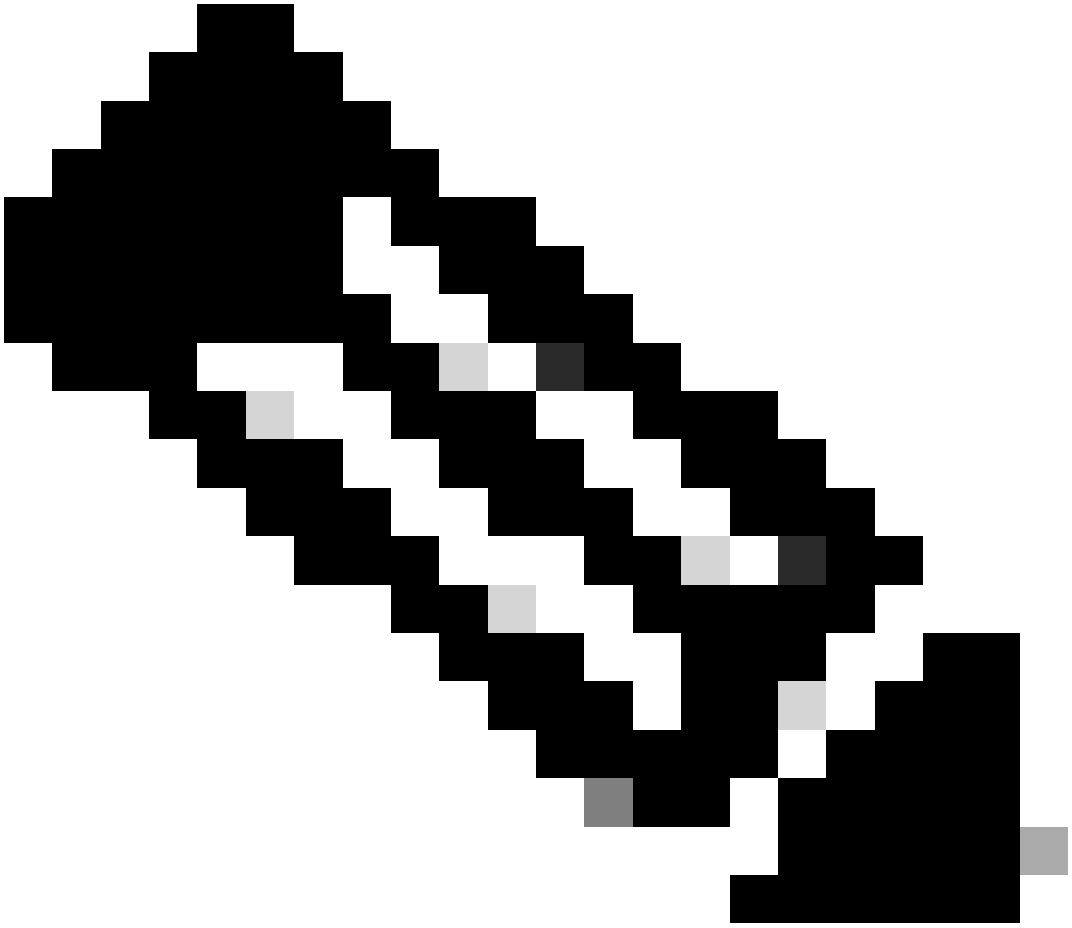
In dit document vindt u informatie over hoe u verschillende methoden kunt gebruiken om routes te verkrijgen van uw AS. Onthoud dat deze routes worden gegenereerd naast andere BGP-routes die BGP via neighbors heeft overgenomen, zowel intern als extern. BGP geeft informatie door die BGP van de ene peer leert en doorgeeft aan andere peers. Het verschil is dat de routes die van het network bevel genereren, herverdelen, of statisch aangeven dat uw AS de oorsprong van deze netwerken is.

Herdistributie is altijd de methode om BGP in IGP te injecteren.

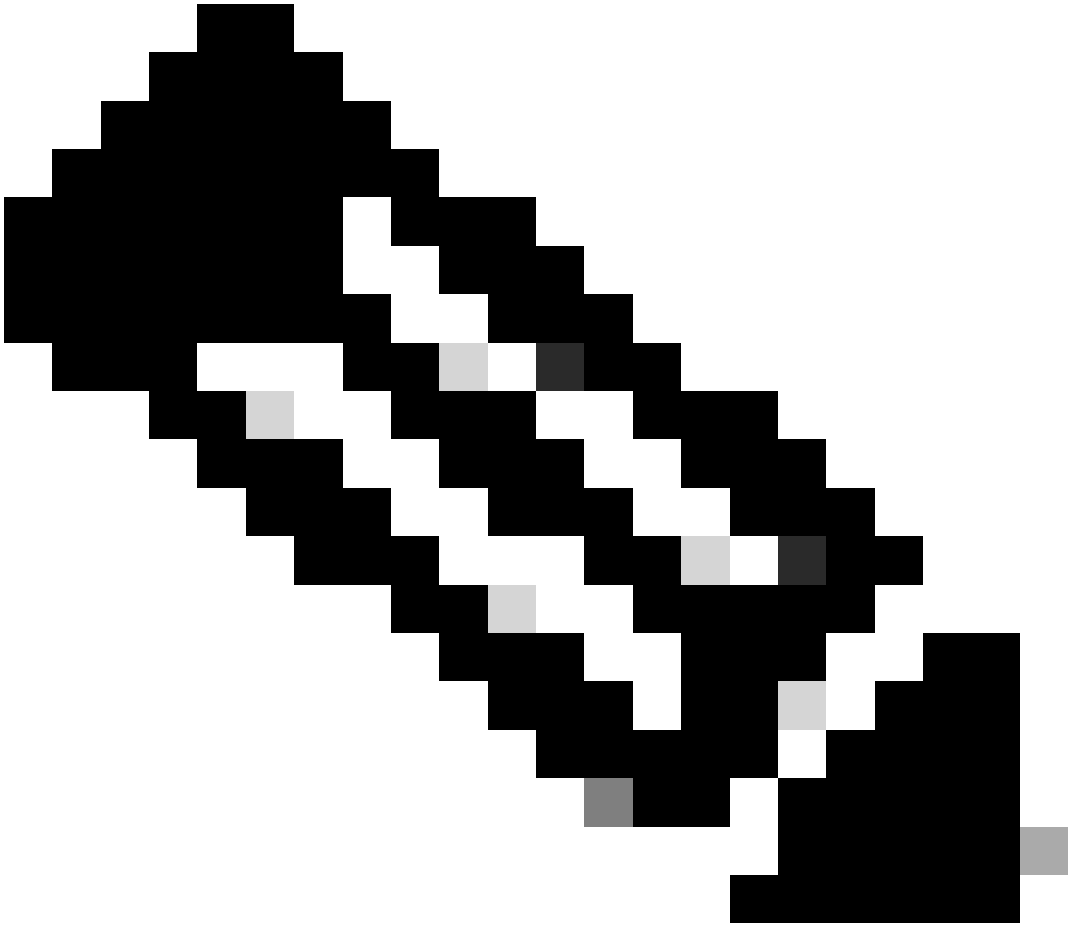
Hierna volgt een voorbeeld:




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

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

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



Opmerking: u hebt geen netwerk 172.31.202.2 of netwerk 172.31.160.0 in RTC nodig, tenzij u wilt dat RTC deze netwerken genereert en deze netwerken doorgeeft als ze binnenkomen uit AS100 en AS200. Opnieuw, het verschil is dat het netwerkbevel een extra reclame voor deze zelfde netwerken toevoegt, die erop wijst dat AS300 ook een oorsprong voor deze routes is.



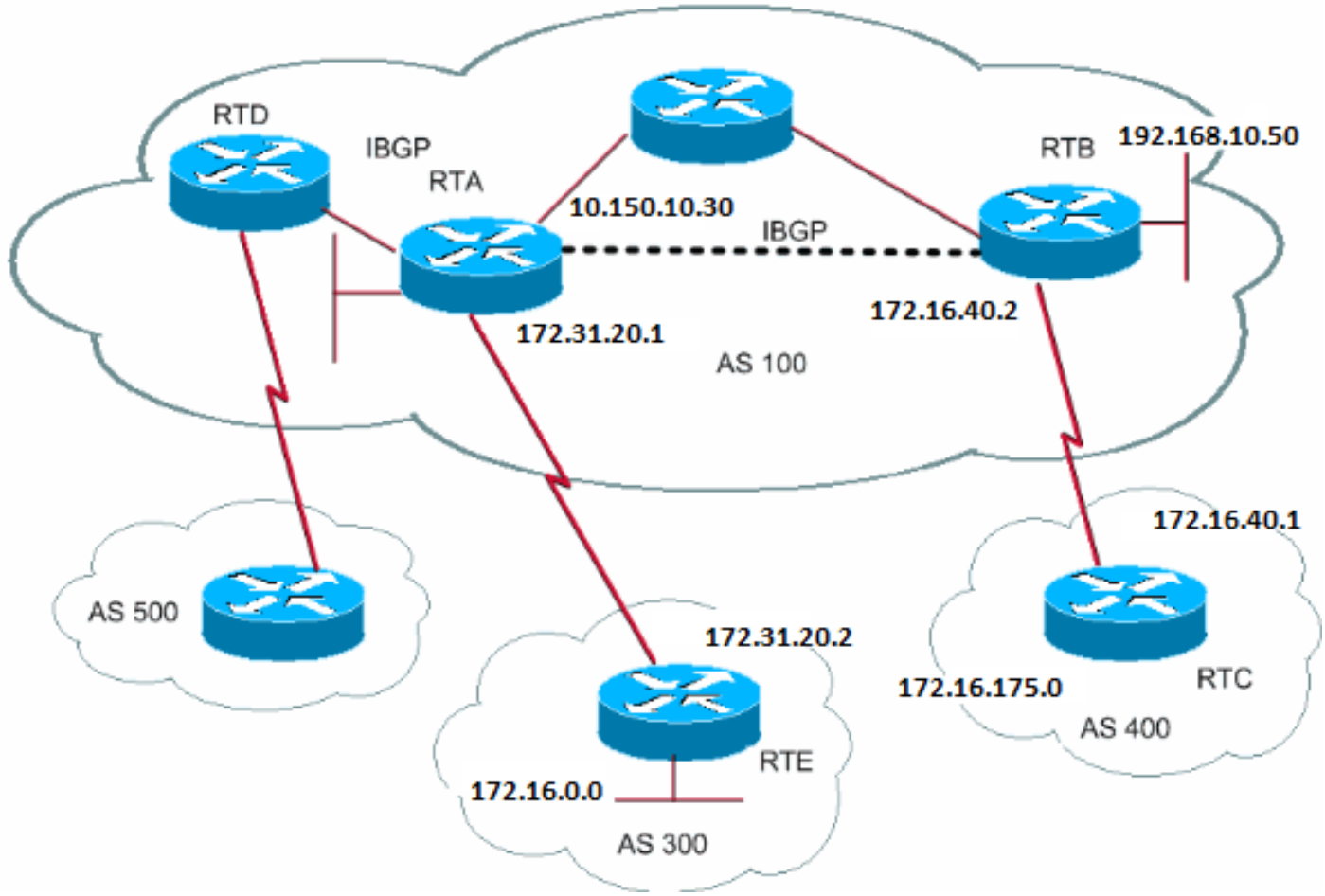
Opmerking: Onthoud dat BGP geen updates accepteert die afkomstig zijn van het eigen AS. Deze weigering garandeert een lusvrije topologie tussen domeinen.

Ga er bijvoorbeeld van uit dat AS200 uit het voorbeeld in deze paragraaf een directe BGP-verbinding naar AS100 heeft. RTA genereert een route 172.31.202.2 en verstuurt de route naar AS300. Dan, RTC gaat deze route tot AS200 over en houdt de oorsprong als AS100. RTB gaat 172.31.202.2 over op AS100 met de oorsprong nog AS100. RTA merkt op dat de update is voortgekomen uit zijn eigen AS en negeert de update.

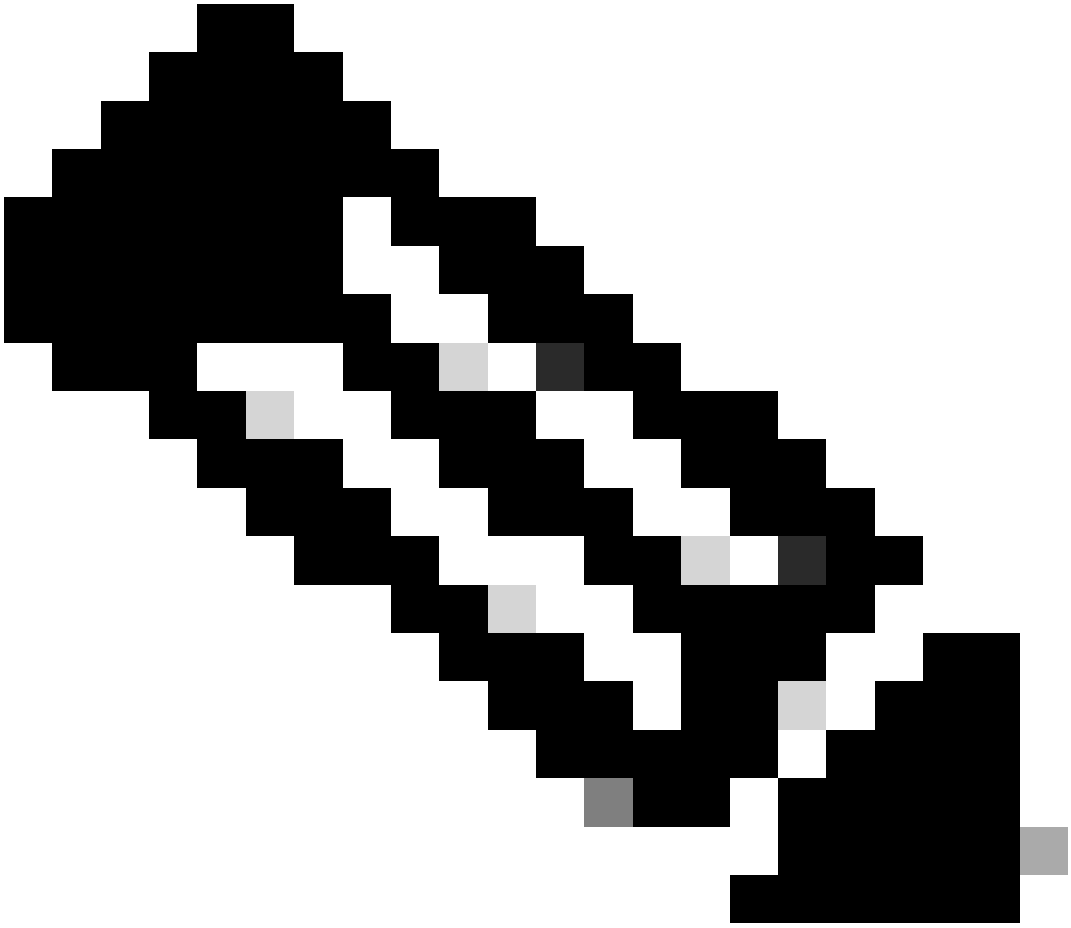
iBGP

U gebruikt iBGP als een AS wil optreden als doorvoersysteem naar een ander AS. U kunt hetzelfde doen als u via eBGP leert, herverdelen in IGP en vervolgens opnieuw verdelen in een ander AS. Maar iBGP biedt meer flexibiliteit en efficiëntere manieren om informatie uit te wisselen

binnen een AS. iBGP biedt bijvoorbeeld manieren om het beste uitgangspunt uit het AS te bepalen met behulp van het kenmerk local-preference. De sectieLocal Preference Attribute geeft meer informatie over de lokale voorkeur.




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

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

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



Opmerking: Onthoud dat wanneer een BGP-speaker een update ontvangt van andere BGP-speakers in het eigen AS (iBGP), de BGP-speaker die de update ontvangt de betreffende informatie niet herdistribueert naar andere BGP-speakers in het eigen AS. De BGP-speaker die de update ontvangt, herdistribueert de informatie naar andere BGP-speakers buiten het AS. Daarom moet een full mesh worden aangehouden tussen de iBGP-speakers binnen een AS.

De RTA en RTB werken met iBGP. RTA en RTD voeren ook iBGP uit. De BGP-updates die afkomstig zijn van RTB naar RTA verzenden naar RTE, buiten het AS. De updates verzenden niet naar RTD, die zich binnen het AS bevindt. Maak daarom een iBGP-peering tussen RTB en RTD om de stroom updates niet te onderbreken.

Het BGP-beslissingsalgoritme

Nadat BGP van verschillende autonome systemen updates ontvangt over verschillende bestemmingen, moet het protocol paden kiezen om een

specifieke bestemming te bereiken. BGP kiest slechts één pad om een specifieke bestemming te bereiken.

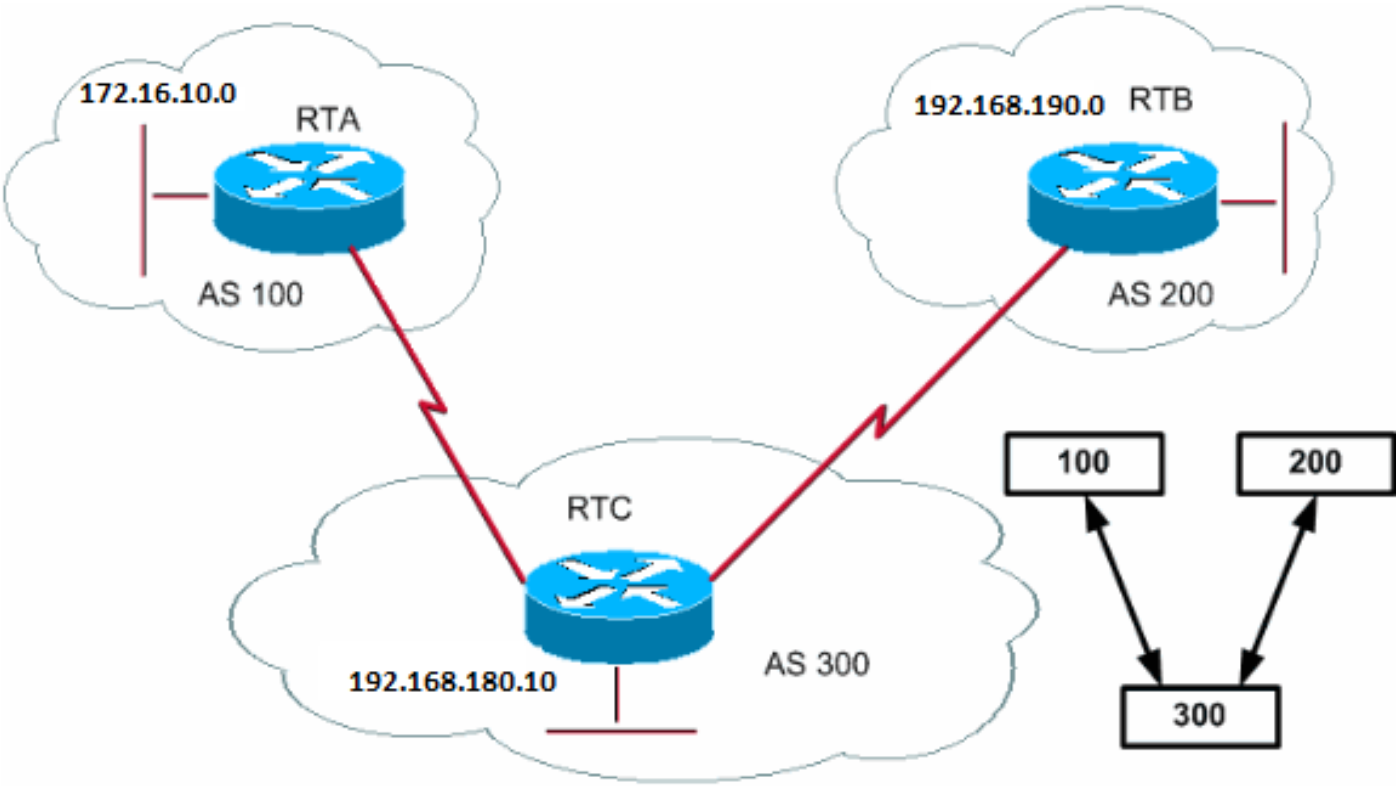
BGP baseert het besluit op verschillende attributes, zoals volgende hop, administratieve gewichten, lokale voorkeur, routeoorsprong, weglengte, oorsprongscode, metriek, en andere attributen.

BGP geeft altijd het beste pad door naar de neighbors. Raadpleeg het BGP-algoritme voor beste padselectie voor meer informatie.

De volgende sectie verklaart deze eigenschappen en hun gebruik.

BGP-casestudy's 2

Kenmerk AS_PATH



Wanneer een route-update via een AS loopt, dan wordt het AS-nummer aan de update toegevoegd. Het kenmerk AS_PATH is in feite de lijst met AS-nummers die een route heeft doorlopen om een bestemming te bereiken. Een AS_SET is een geordende wiskundige set {} van alle autonome systemen die zijn doorlopen. De sectie CIDR Voorbeeld 2 (as-set) van dit document geeft een voorbeeld van AS_SET.

In het voorbeeld in deze sectie, adverteert RTB netwerk 192.168.190.0 in AS200. Wanneer die route AS300 passeert, voegt RTC haar eigen AS-nummer toe aan het netwerk. Wanneer 192.168.190.0 RTA bereikt, heeft het netwerk twee AS-nummers toegevoegd: eerst 200, dan 300. Voor RTA is het pad naar 192.168.190.0 (300, 200).

Hetzelfde proces is van toepassing op 172.16.10.0 en 192.168.180.10. RTB moet het pad nemen (300, 100); RTB vervolgt AS300 en dan AS100 om 172.16.10.0 te bereiken. RTC moet pad (200) doorlopen om 192.168.190.0 en pad (100) te bereiken om 172.16.10.0 te bereiken.

Kenmerk origin

De oorsprong ('origin') is een verplicht kenmerk dat de oorsprong van de padinformatie definieert. Het kenmerk origin kan drie waarden hebben:

-

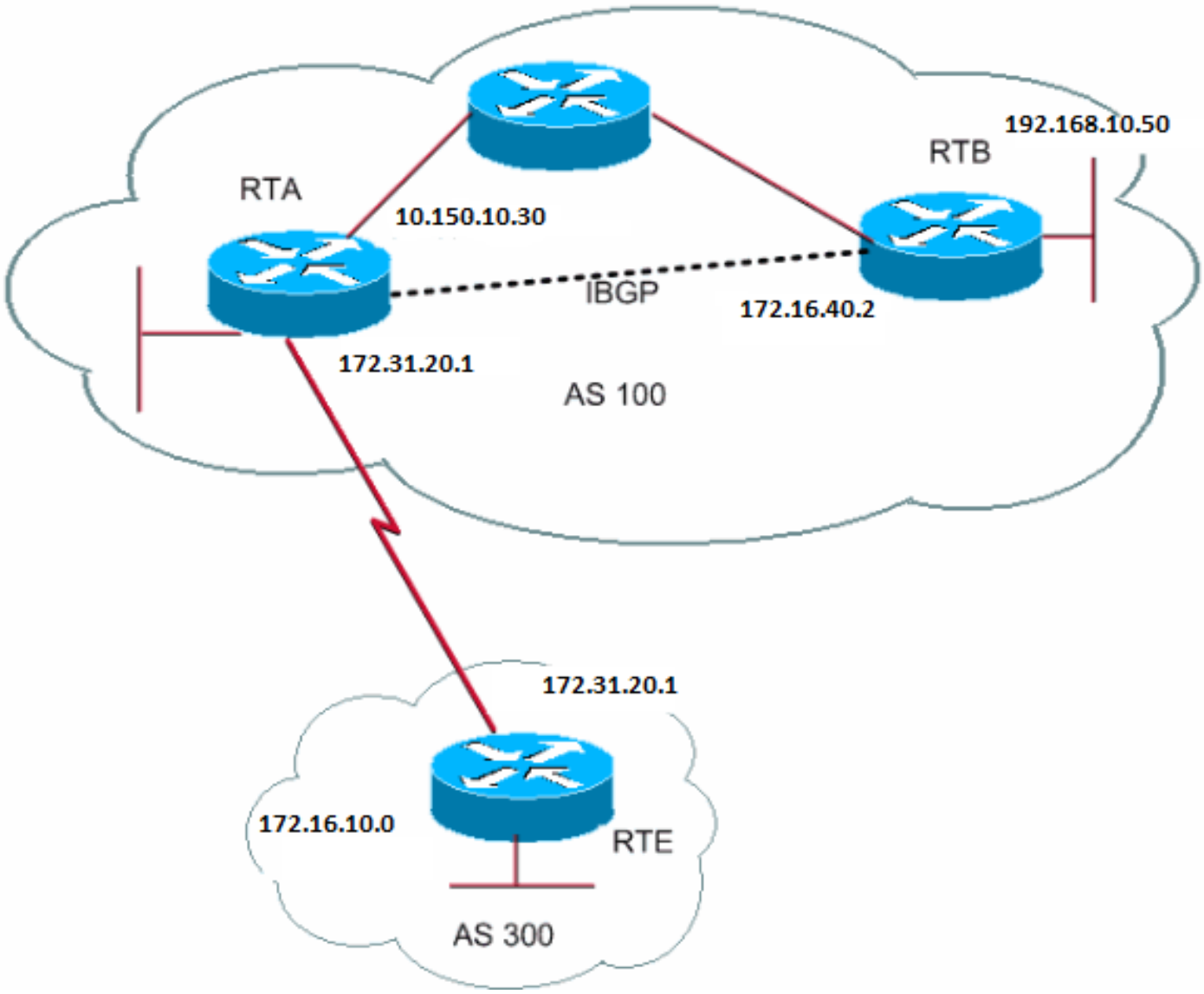
IGP – Network Layer Reachability Information (NLRI) hoort bij het AS van oorsprong. Dit gebeurt normaal wanneer u het **bgp network**bevel uitgeeft. De animatie van de BGP-tabel geeft IGP aan.

-

EGP – NLRI wordt geleerd via Exterior Gateway Protocol (EGP). Anein de BGP-tabel geeft EGP aan.

-

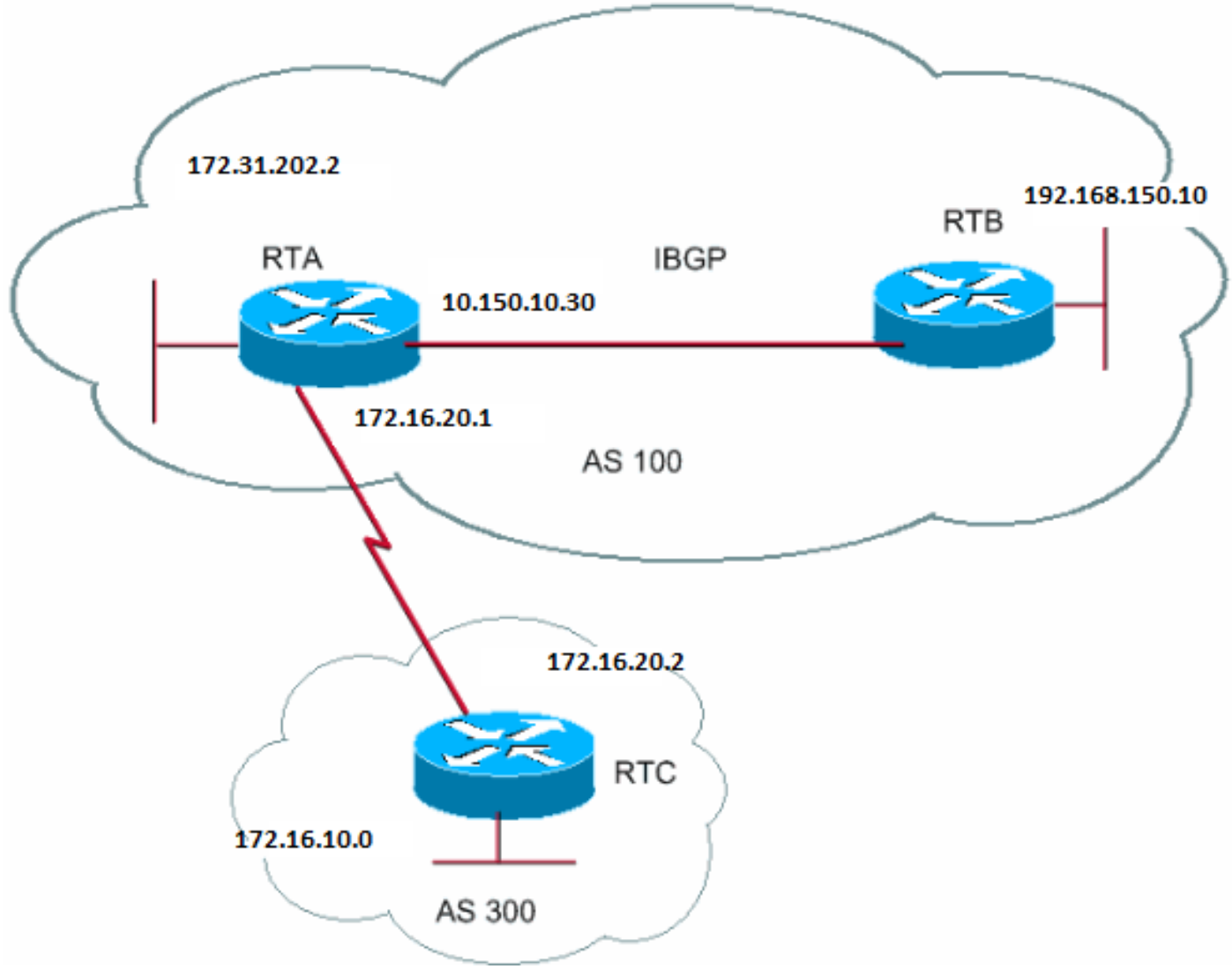
INCOMPLETE – NLRI is onbekend of op een andere manier geleerd. INCOMPLETE treedt doorgaans op wanneer u routes van andere routingprotocollen herdistribueert naar BGP waarbij de oorsprong van de route onvolledig is. En?in de BGP-tabel duidt op INCOMPLEET.



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

ip route 192.168.190.0 255.255.0.0 null0

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

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

RTA bereikt 172.16.10.0 via 300 i. De ‘300 i’ betekent dat het volgende AS-pad 300 is en dat de oorsprong van de route IGP is. RTA bereikt ook 192.168.10.150 via i. Deze ‘i’ betekent dat de vermelding zich in hetzelfde AS bevindt en dat de oorsprong IGP is. RTE bereikt 172.31.202.2 via 100 i. De ‘100 i’ betekent dat het volgende AS 100 is en dat de oorsprong IGP is. RTE bereikt ook 192.168.190.0 via 100 ?. De "100 ?" betekent dat het volgende AS 100 is en dat de oorsprong onvolledig is en afkomstig is van een statische route.

BGP-kenmerk next-hop



BGP-kenmerk next-hop

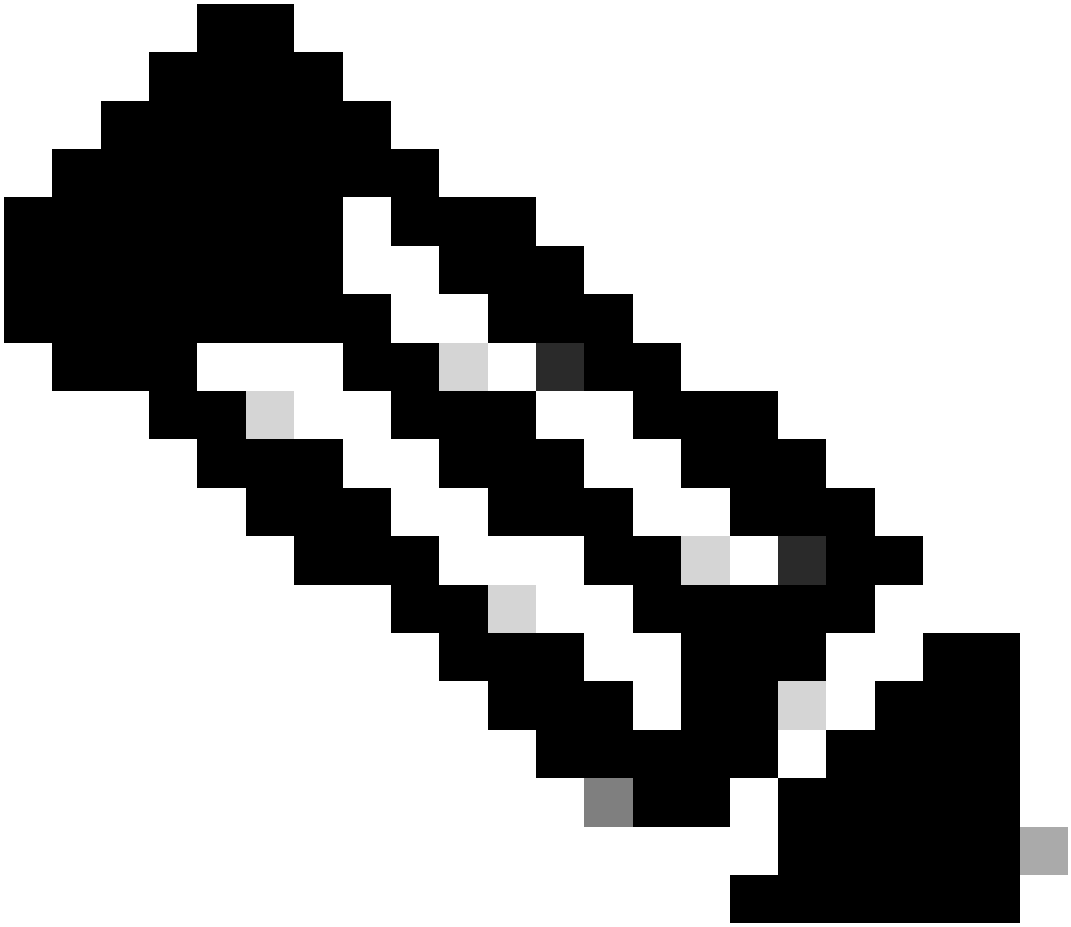
Het BGP-kenmerk next-hop is het IP-adres van de volgende hop dat moet worden gebruikt om een bepaalde bestemming te bereiken.

Voor eBGP is de volgende hop altijd het IP-adres van de buur die door de neighbor opdracht wordt gespecificeerd. In het voorbeeld in deze sectie, adverteert RTC 172.16.10.0 aan RTA met een volgende hop van 172.31.20.2. RTA adverteert 172.31.202.2 aan RTC met een volgende hop van 172.31.20.1. Voor iBGP staat in het protocol dat de volgende hop die eBGP adverteert, in iBGP moet worden uitgevoerd. Vanwege deze regel adverteert RTA 172.16.10.0 naar zijn iBGP peer RTB met een volgende hop van 172.31.20.2. Gebaseerd op RTB, is de volgende hop om 172.16.10.0 te bereiken 172.31.20.2 *en niet* 10.150.10.30.

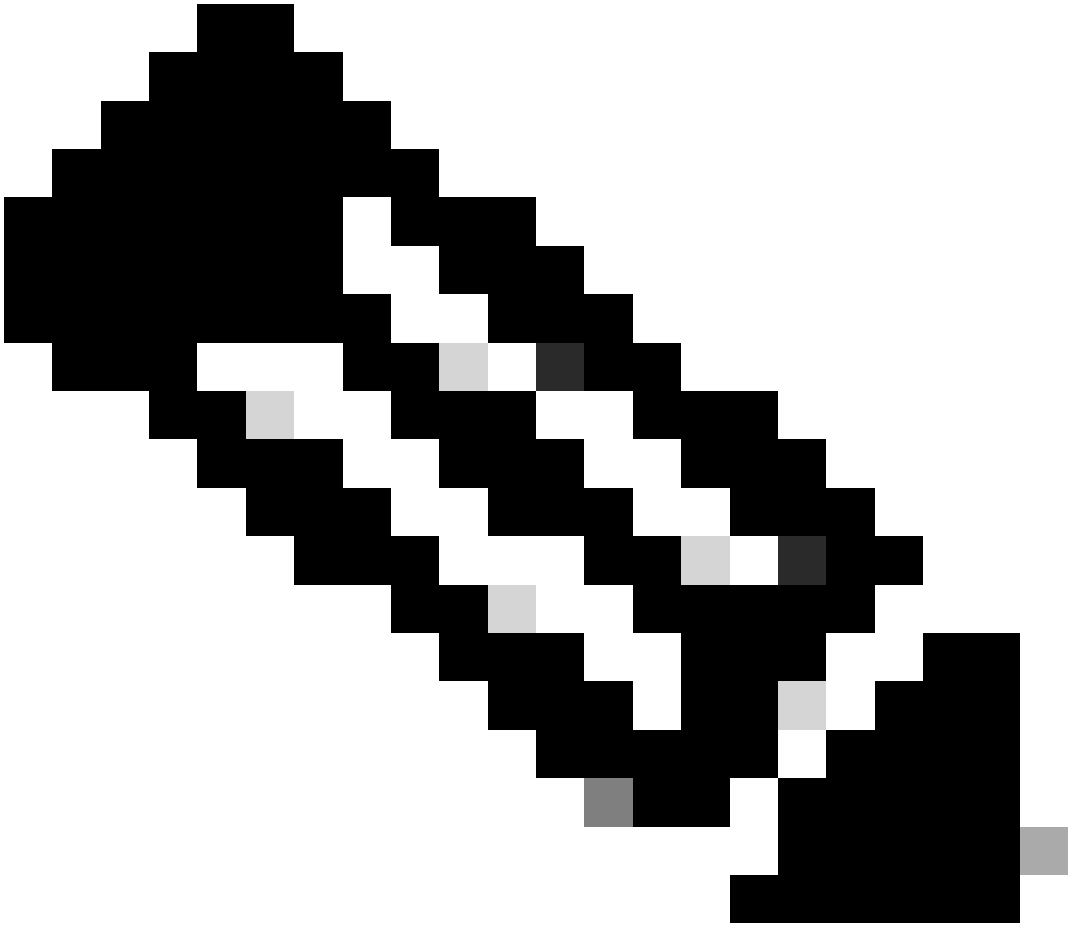
Zorg ervoor dat RTB 172.31.20.2 kan bereiken via IGP. Anders wijst RTB pakketten met de bestemming 172.16.10.0 af omdat het volgende hopadres ontoegankelijk is. Als RTB bijvoorbeeld iGRP uitvoert, kunt u iGRP ook uitvoeren op RTA-netwerk 172.16.10.0. U wilt iGRP passief maken op de koppeling naar RTC zodat BGP alleen wordt uitgewisseld.

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

RTB#
```

```
router bgp 100
 neighbor 10.150.10.30 remote-as 100

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



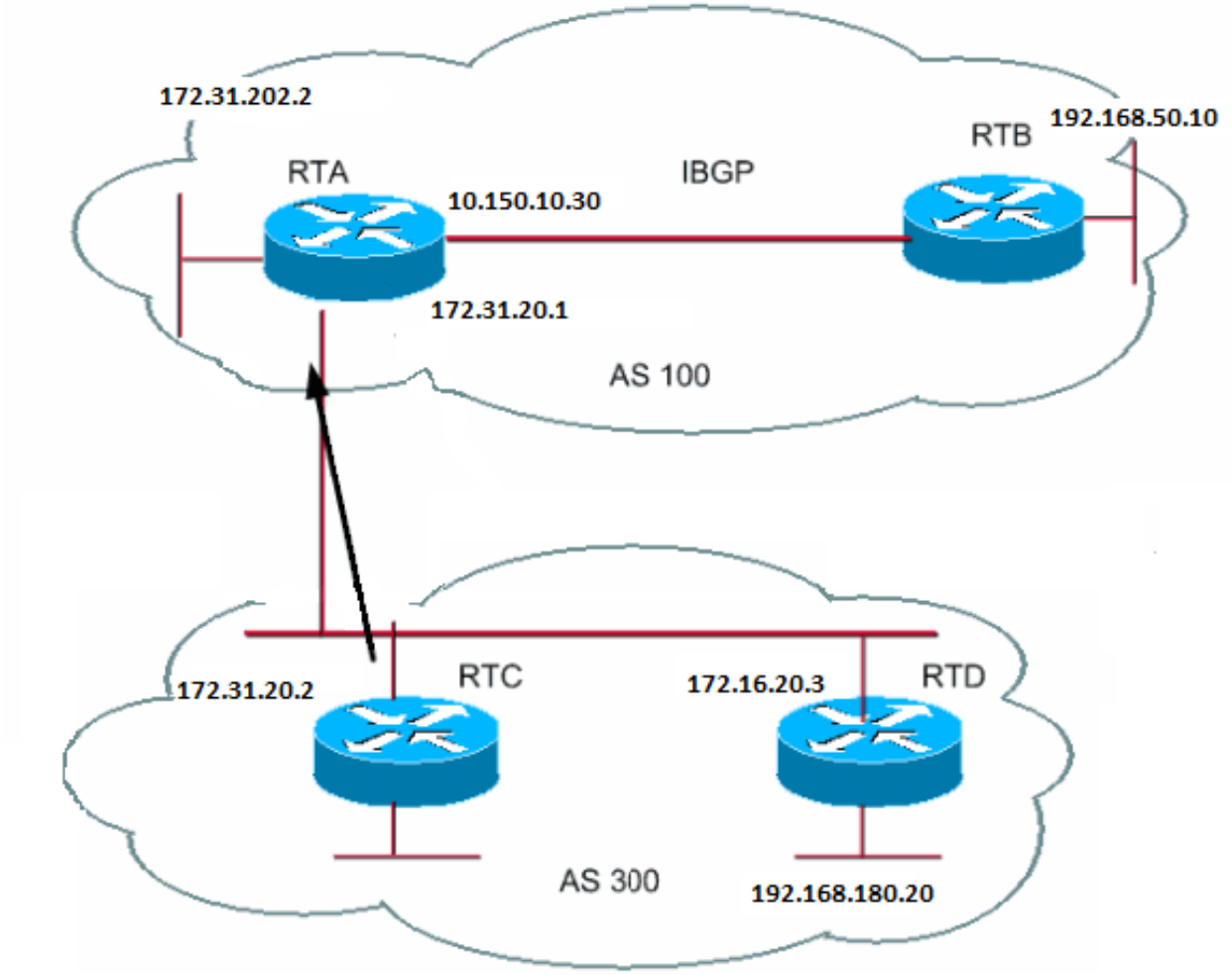
Opmerking: RTC kondigt 172.16.10.0 aan voor RTA met als volgende hop 172.31.20.2.



Opmerking: RTA adverteert 172.16.10.0 naar RTB met een volgende hop gelijk aan 172.31.20.2. De eBGP volgende hop wordt gedragen in iBGP.

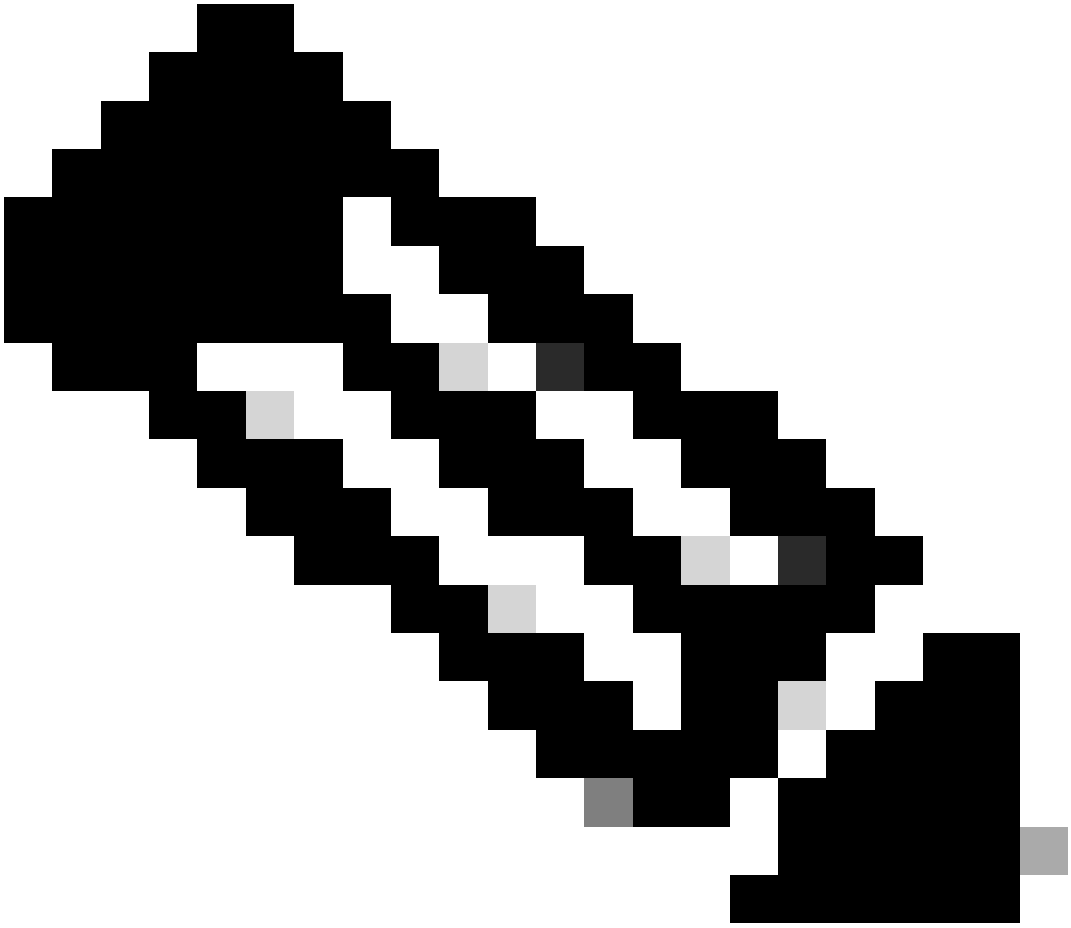
Wees extra voorzichtig wanneer u te maken hebt met multi-access en non-broadcast multiaccess (NBMA) netwerken. De secties BGP Next Hop (Multiaccess Networks) en BGP Next Hop (NBMA) bieden meer details.

BGP-kenmerk next-hop (multi-access netwerken)



Dit voorbeeld toont hoe de volgende hop zich gedraagt op een multi-access netwerk zoals Ethernet.

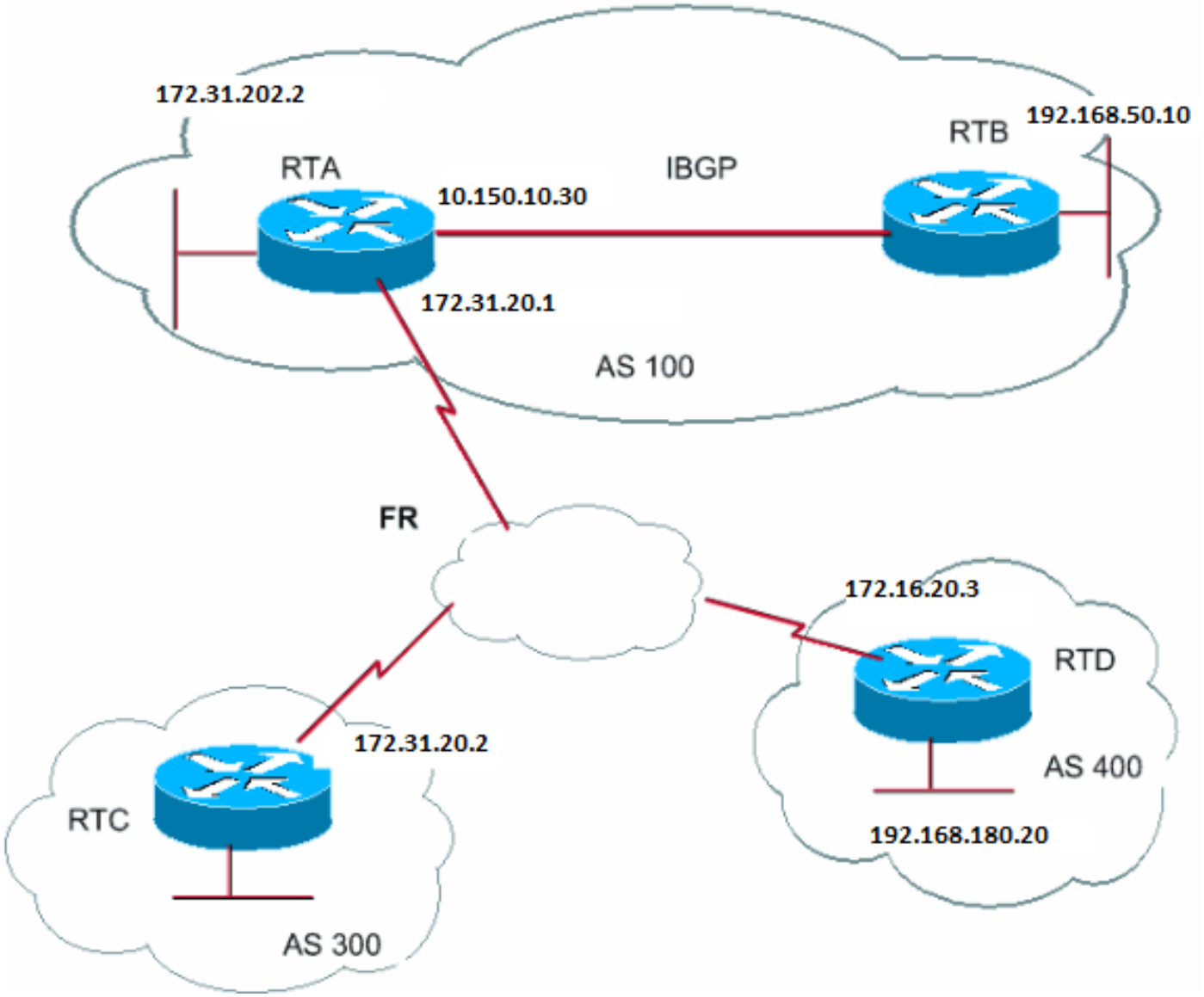
Stel dat RTC en RTD OSPF uitvoeren in AS300. RTC voert BGP uit met RTA. RTC kan netwerk 192.168.180.20 bereiken via 172.16.20.3. Wanneer RTC een BGP-update naar RTA stuurt met betrekking tot 192.168.180.20, gebruikt RTC als volgende hop 172.16.20.3. RTC gebruikt geen eigen IP-adres, 172.31.20.2. RTC gebruikt dit adres omdat het netwerk tussen RTA, RTC en RTD een multiaccess netwerk is. Dat RTA RTD gebruikt als volgende hop om 192.168.180.20 te bereiken is logischer dan de extra hop via RTC.



Opmerking: RTC kondigt 192.168.180.20 aan voor RTA met als volgende hop 172.16.20.3.

Als het gemeenschappelijke medium naar RTA, RTC en RTD niet multi-access is, maar NBMA, dan zullen er verdere complicaties optreden.

BGP-kenmerk next-hop (NBMA)



Het gemeenschappelijke medium heeft de vorm van een wolk in het diagram. Als het gemeenschappelijke medium een Frame Relay- of NBMA-cloud is, dan is het gedrag exact hetzelfde als wanneer u een verbinding via Ethernet heeft. RTC kondigt 192.168.180.20 aan voor RTA met als volgende hop 172.16.20.3.

Het probleem is dat RTA geen direct Permanent Virtual Circuits (PVC) heeft met RTD, en de volgende hop niet kan bereiken. In dit geval mislukt de routing.

Het next-hop-selfcommando lost deze situatie op.

Opdracht next-hop-self

Voor situaties met de volgende hop, zoals in het voorbeeld BGP Next Hop (NBMA), kunt u de next-hop-self opdracht gebruiken. De syntaxis is:

<#root>

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

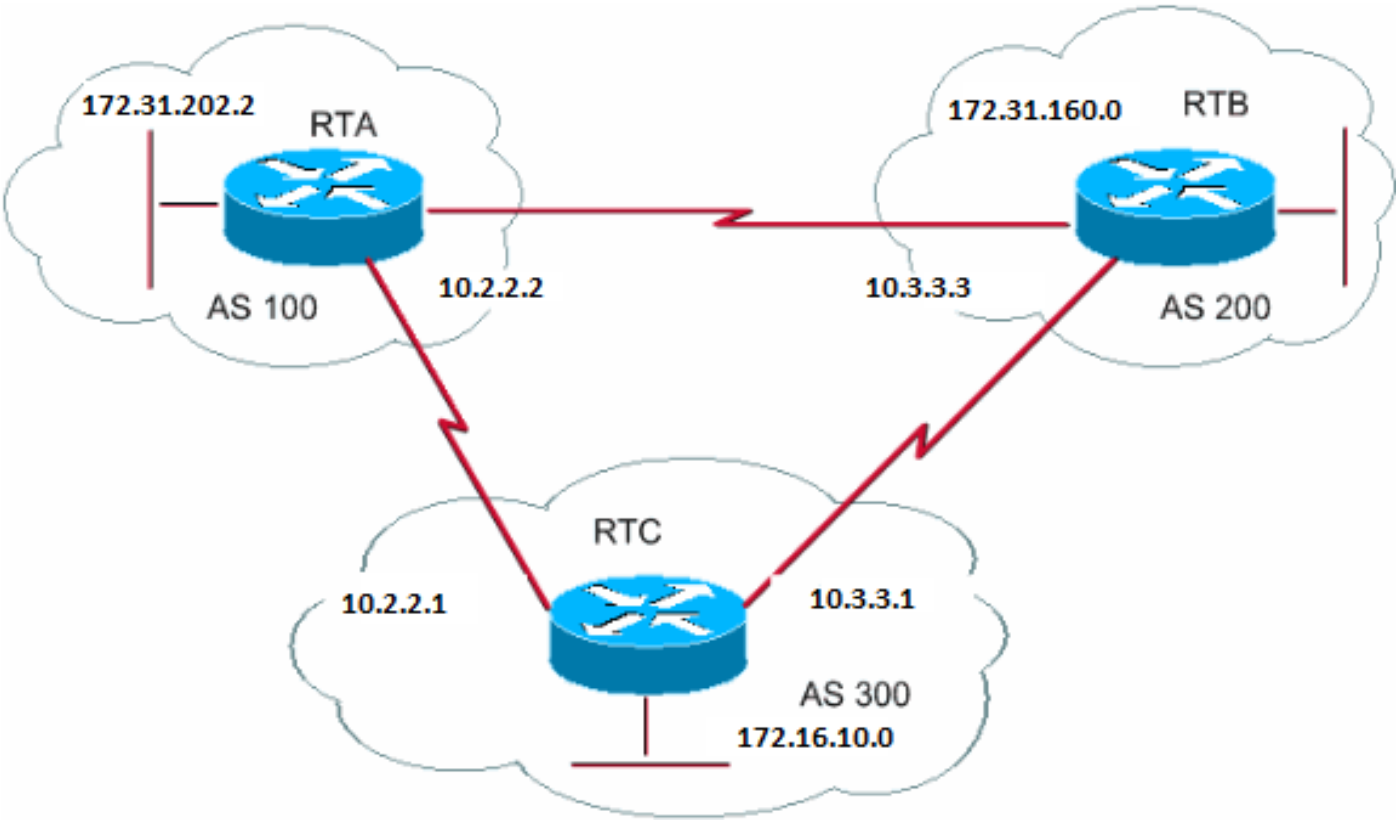
Met deze next-hop-selfopdracht kunt u BGP dwingen een specifiek IP-adres als volgende hop te gebruiken.

Voor het voorbeeld BGP-kenmerk next-hop (NBMA) lost deze configuratie het probleem op:

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

RTC kondigt 192.168.180.20 aan voor RTA met als volgende hop 172.31.20.2.

BGP-backdoor



In het vorige diagram worden eBGP met RTA en RTC uitgevoerd. RTB en RTC voeren eBGP uit. RTA en RTB voeren een vorm van IGP uit:

RIP, IGRP of een ander protocol. Per definitie hebben eBGP-updates een afstand van 20, wat minder is dan de IGP-afstanden. De standaardafstanden zijn:

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

RTA ontvangt updates over 172.31.160.0 via twee routingprotocollen:

- eBGP met een afstand van 20
- IGP met een afstand die groter is dan 20

Standaard heeft BGP de volgende afstanden:

- Externe afstand: 20
- Interne afstand: 200

-

  Lokale afstand: 200

Maar u kunt de distance opdracht gebruiken om de standaardafstanden te wijzigen:

<#root>

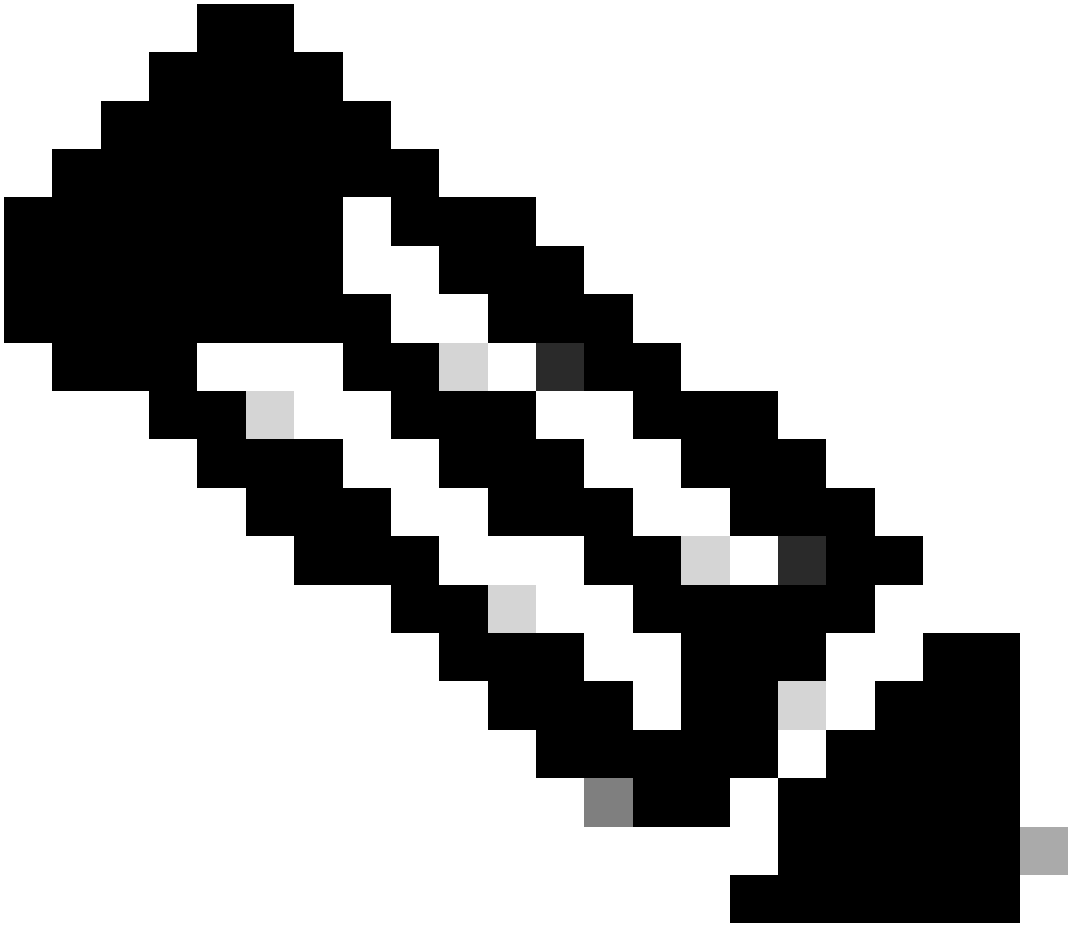
distance bgp <external-distance> <internal-distance> <local-distance>

RTA kiest eBGP via RTC vanwege de kortere afstand.

Als u wilt dat RTA via RTB (IGP) over 172.31.160.0 leert, dan heeft u twee opties:

-

  De externe afstand van eBGP of de IGP-afstand wijzigen.



Opmerking: Deze wijziging wordt niet aanbevolen.

- BGP-backdoor gebruiken.

BGP-backdoor maakt de IGP-route de voorkeursroute.

Geef de opdracht **netwerk addressbackdoor uit**.

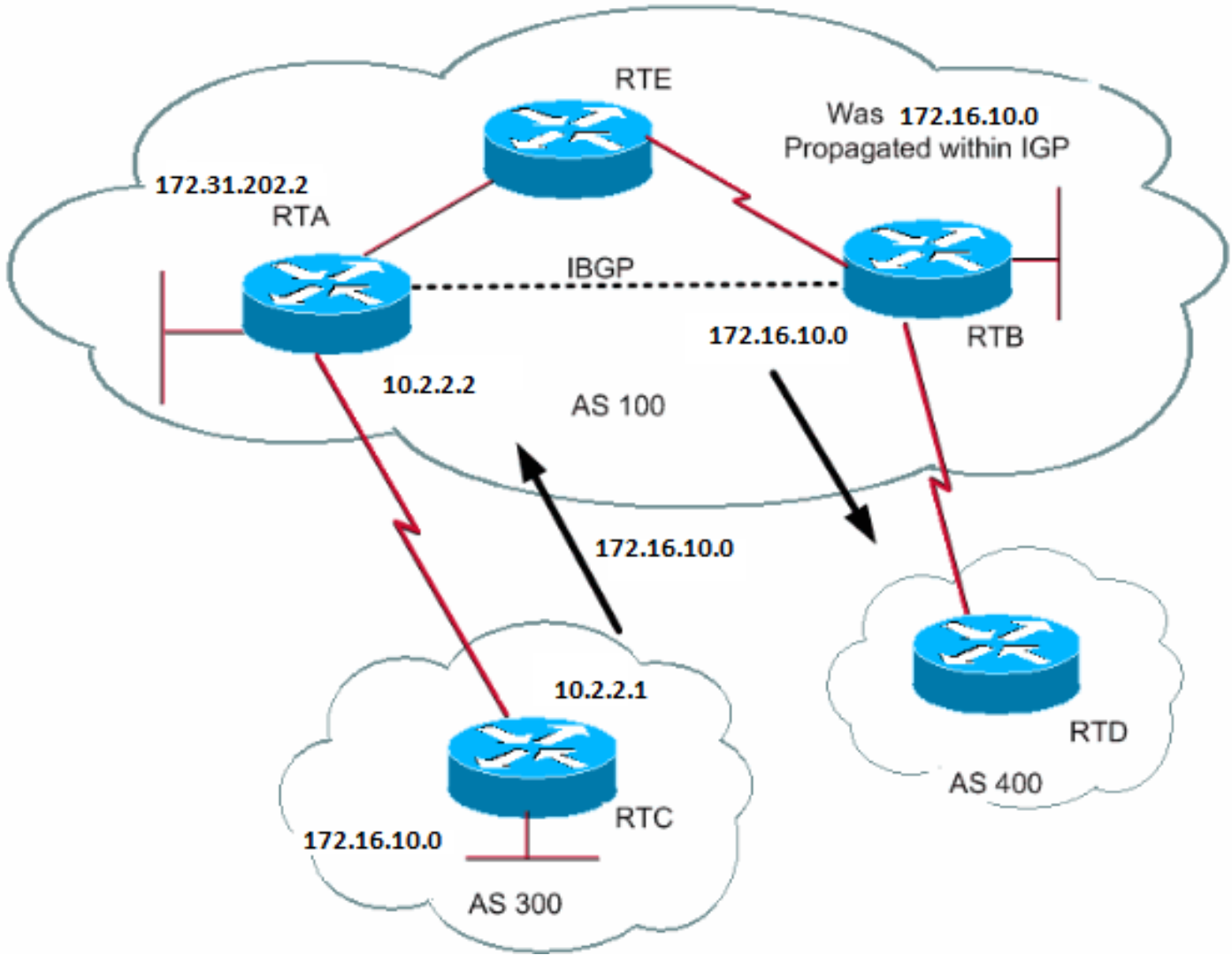
Het geconfigureerde netwerk is het netwerk dat u via IGP wilt bereiken. Voor BGP krijgt dit netwerk dezelfde behandeling als een lokaal toegewezen netwerk, alleen kondigen de BGP-updates dit netwerk niet aan.

```
RTA#
router eigrp 10
 network 172.31.202.2

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

Network 172.31.160.0 wordt behandeld als een lokale ingang, maar wordt niet geadverteerd als een normale netwerkingang.

RTA leert 172.31.160.0 van RTB via EIGRP met afstand 90. RTA leert ook het adres van RTC via eBGP met afstand 20. Normaal is eBGP de voorkeur, maar vanwege de **netwerk backdoor** opdracht, EIGRP is de voorkeur.

Synchronisatie



Voordat we ingaan op synchronisatie bekijken we eerst het volgende scenario. RTC in AS300 verstuurt updates over 172.16.10.0. RTA en RTB

draaien iBGP, dus RTB krijgt de update en kan 172.16.10.0 bereiken via volgende hop 10.2.2.1. Vergeet niet dat de volgende hop via iBGP wordt vervoerd. Om de volgende hop te bereiken, moet RTB het verkeer naar RTE sturen.

Stel dat RTA netwerk 172.16.10.0 niet heeft geherdistribueerd in IGP. RTE heeft nu geen idee dat 172.16.10.0 bestaat.

Als RTB begint te adverteren naar AS400 dat RTB 172.16.10.0 kan bereiken, komt het verkeer dat van RTD naar RTB komt met bestemming 172.16.10.0 in en daalt bij RTE.

Synchronisatie geeft aan dat, als uw AS verkeer van een andere AS naar een derde AS doorgeeft, BGP geen route mag adverteren voordat alle routers in uw AS via IGP over de route hebben geleerd. BGP wacht tot IGP de route binnen het AS heeft doorgegeven. Vervolgens kondigt BGP de route aan bij externe peers.

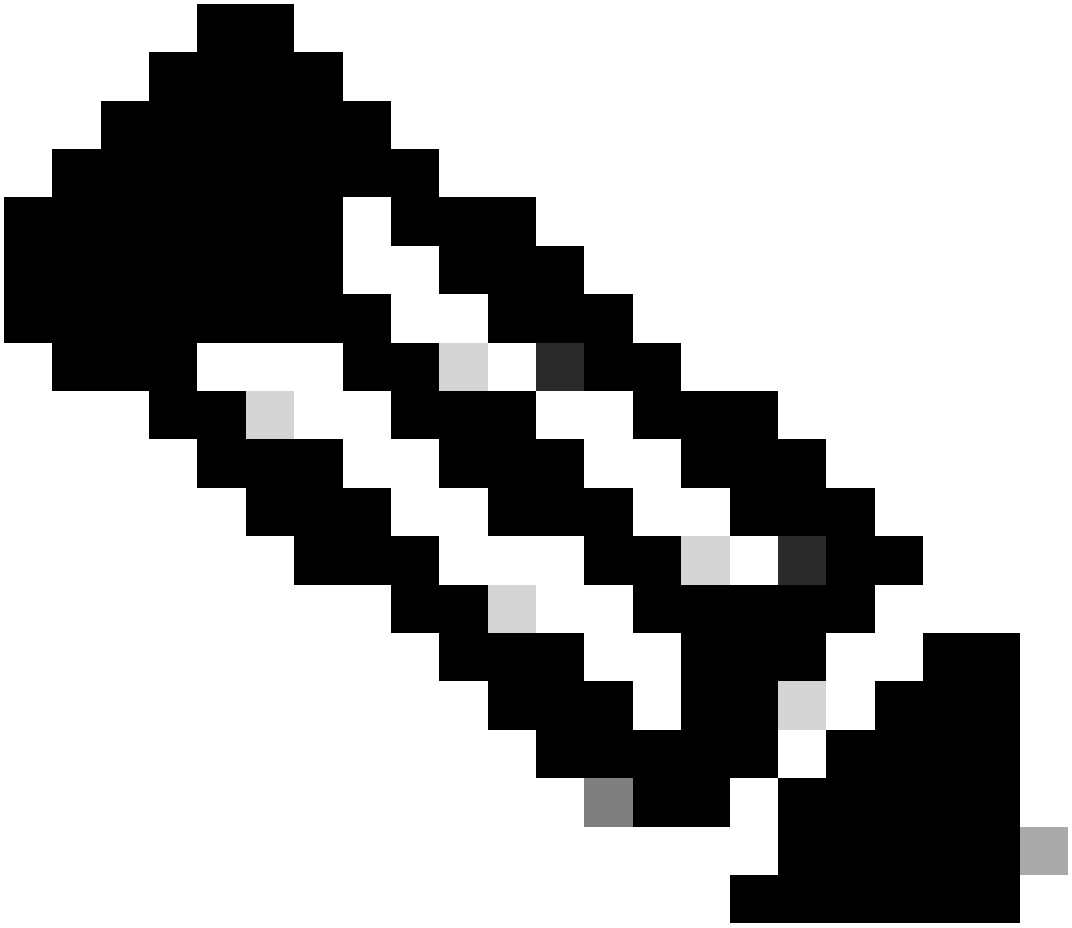
In het voorbeeld in deze sectie wacht RTB op informatie over 172.16.10.0 via IGP. Vervolgens begint RTB met het verzenden van de update naar RTD. U kunt RTB laten denken dat IGP de informatie heeft verspreid als u een statische route in RTB toevoegt die naar 172.16.10.0 verwijst. Zorg ervoor dat andere routers 172.16.10.0 kunnen bereiken.

Synchronisatie uitschakelen

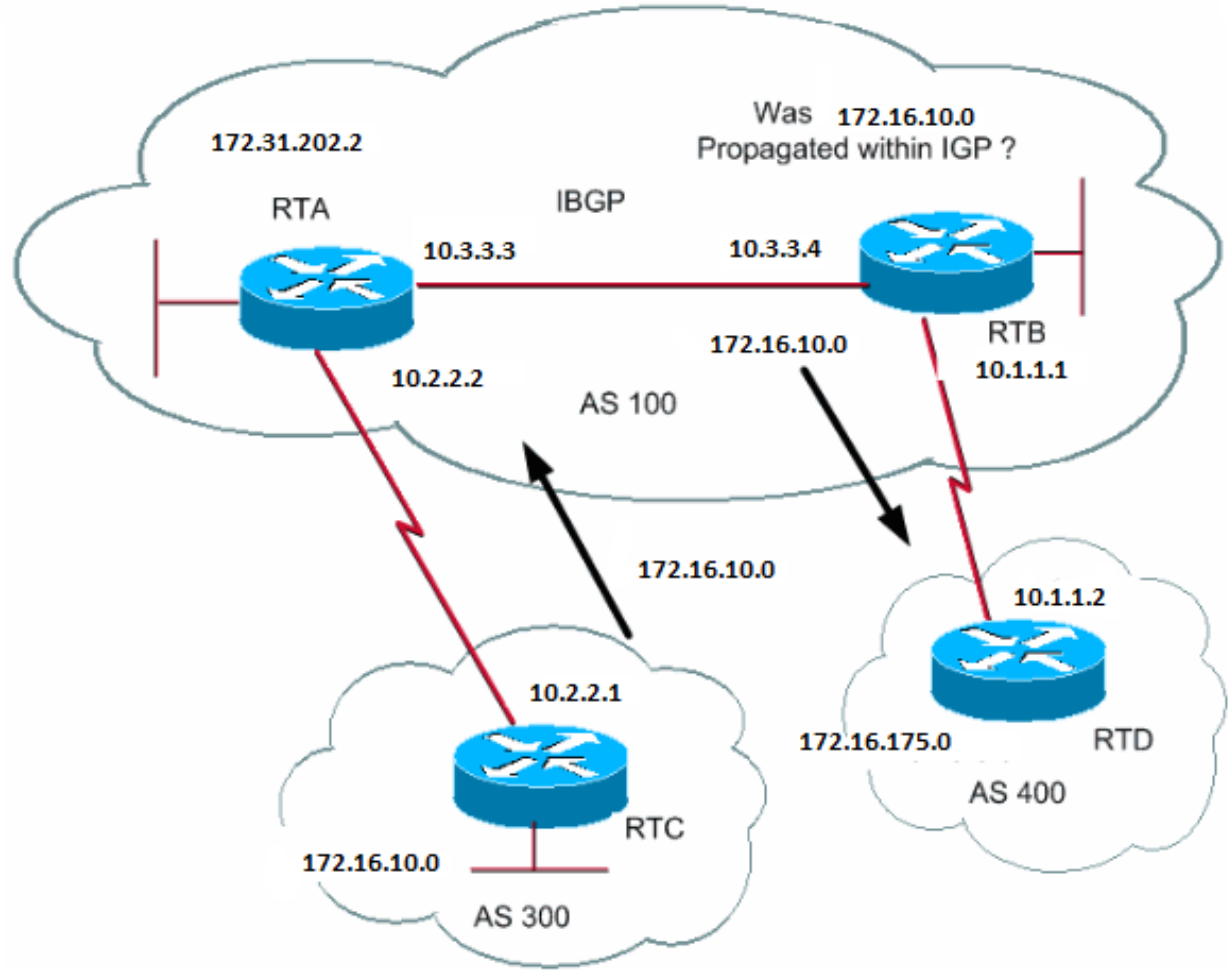
In sommige gevallen heeft u geen synchronisatie nodig. Als u geen verkeer van een ander AS doorgeeft via uw AS, dan kunt u synchronisatie uitschakelen. U kunt synchronisatie ook uitschakelen als alle routers in uw AS BGP uitvoeren. Door deze functie uit te schakelen kunt u minder routes in uw IGP overdragen en kan BGP sneller convergeren.

Het uitschakelen van de synchronisatie gaat niet automatisch. Wanneer al uw routers in het AS BGP uitvoeren en u helemaal geen IGP uitvoert, dan kan de router dit niet weten. Uw router wacht oneindig op een IGP-update over een bepaalde route voordat de router de route naar externe peers verstuurt. U moet synchronisatie in dit geval handmatig uitschakelen zodat de routing correct functioneert:

```
router bgp 100
 no synchronization
```

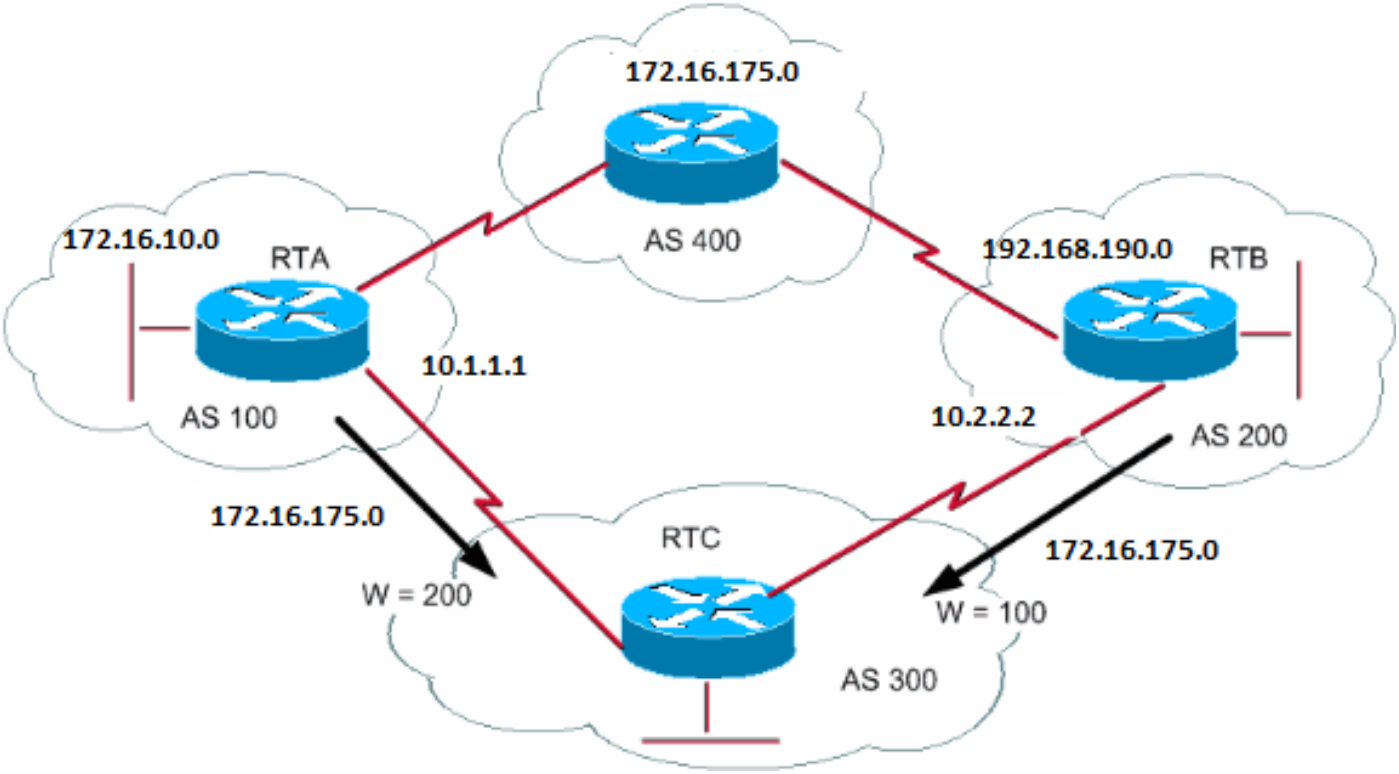


Opmerking: Zorg ervoor dat u de opdracht IP bgp-adres wissen om de sessie opnieuw in te stellen.




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

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

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

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

Kenmerk weight



Het kenmerk weight is een door Cisco gedefinieerd kenmerk. Dit kenmerk gebruikt gewicht om het beste pad te selecteren. Het gewicht wordt lokaal aan de router toegewezen. De waarde heeft alleen betekenis voor de specifieke router. De waarde wordt niet doorgegeven of verspreid via een van de route-updates. Een gewicht kan een getal van 0 tot 65.535 zijn. Paden die de router voortkomt hebben standaard een gewicht van 32.768 en andere paden hebben een gewicht van 0.

Routes met een hogere gewichtswaarde hebben de voorkeur wanneer er meerdere routes naar dezelfde bestemming zijn. Bekijk het voorbeeld in deze sectie. RTA heeft over netwerk 172.16.0.0 van AS4 geleerd. RTA brengt de update naar RTC. RTB heeft ook geleerd over netwerk 172.16.0.0 van AS4. RTB brengt de update naar RTC. RTC heeft nu twee manieren om 172.16.0.0 te bereiken en moet beslissen welke route er wordt genomen. Als u het gewicht van de updates op RTC die uit RTA komen zo instelt dat het gewicht groter is dan het gewicht van updates die van RTB komen, dwingt u RTC om RTA als volgende hop te gebruiken om 172.16.0.0 te bereiken. Meerdere methoden bereiken dit gewicht ingesteld:

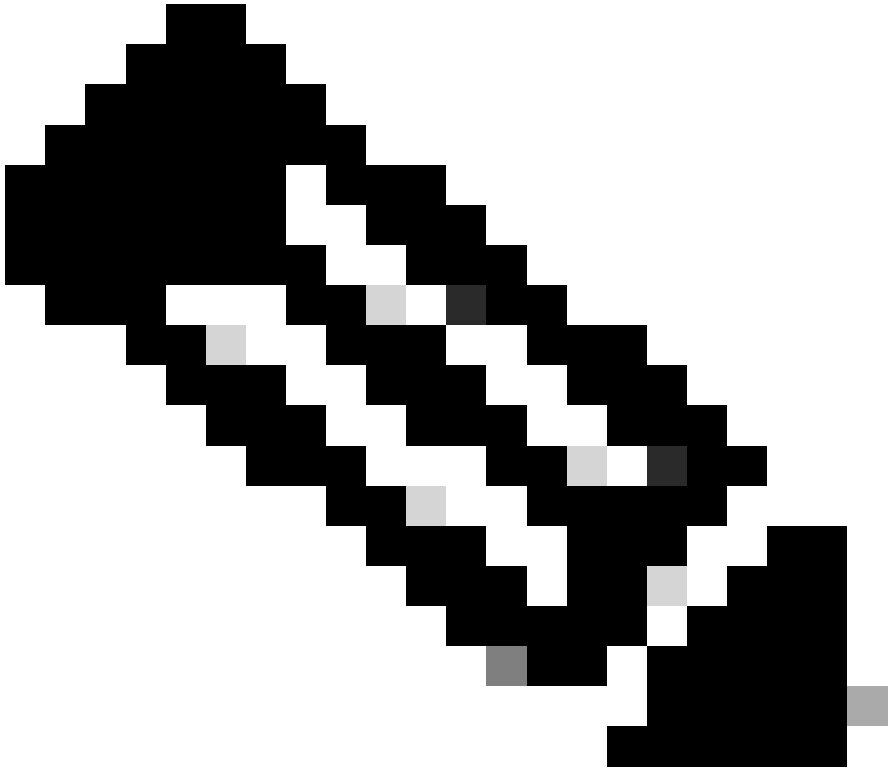
- De opdracht neighbor gebruiken.

  buur {ip-adres|peer-group} gewicht <weight>

- AS_PATH-toegangslijsten gebruiken.

- **IP as-path access-list <access-list-number>{vergunning | deny} <as-regular-expressie>**

- **buurman <ip-adres>filterlijst <access-list-number>weight <weight>**



Opmerking: in sommige scenario's kunnen er zeer weinig opdrachten zijn die niet beschikbaar zijn in sommige softwareversies.

-

Routekaarten gebruiken.

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

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

 neighbor 10.2.2.2 remote-as 200
 neighbor 10.2.2.2 weight 100

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

RTA heeft een hogere gewichtswaarde en heeft de voorkeur als volgende hop.

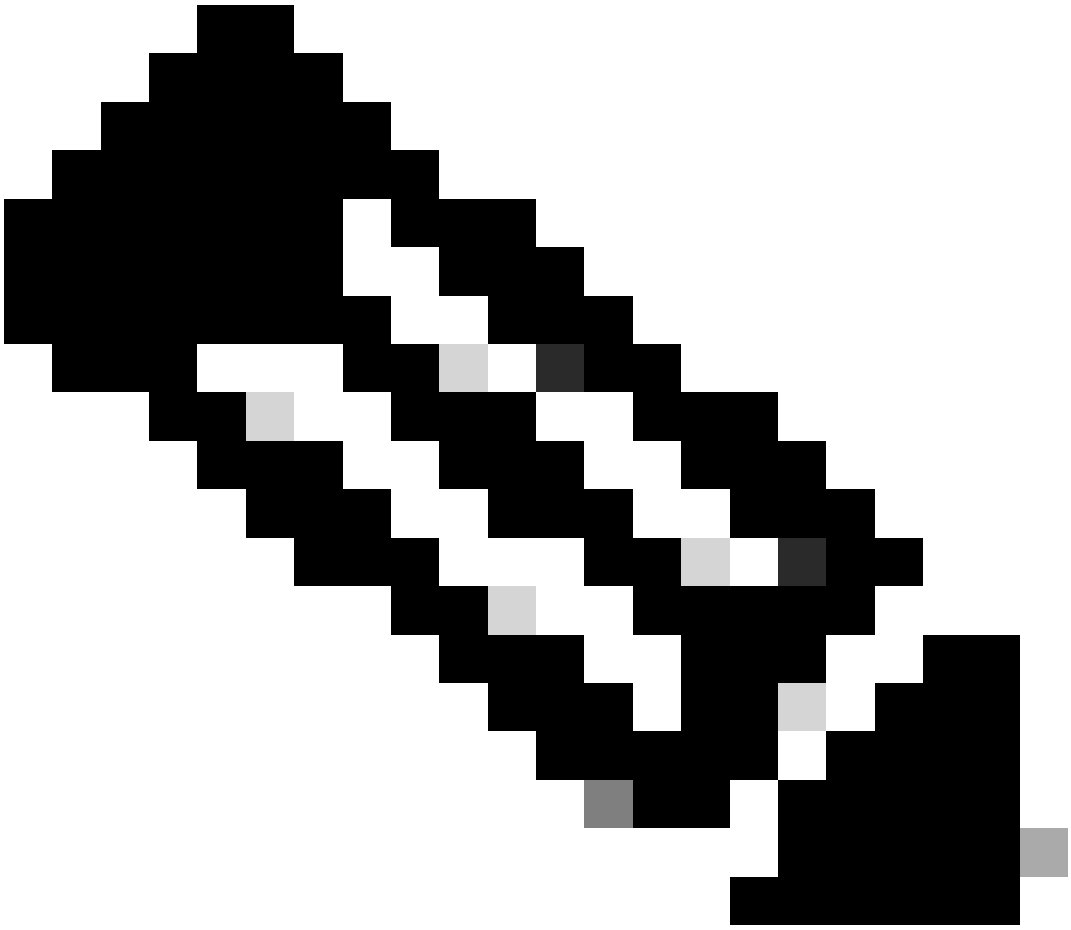
U kunt hetzelfde resultaat bereiken met IP AS_PATH en filterlijsten.

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

!--- This only permits path 100.

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

En hetzelfde resultaat kan ook worden bereikt door routekaarten te gebruiken.

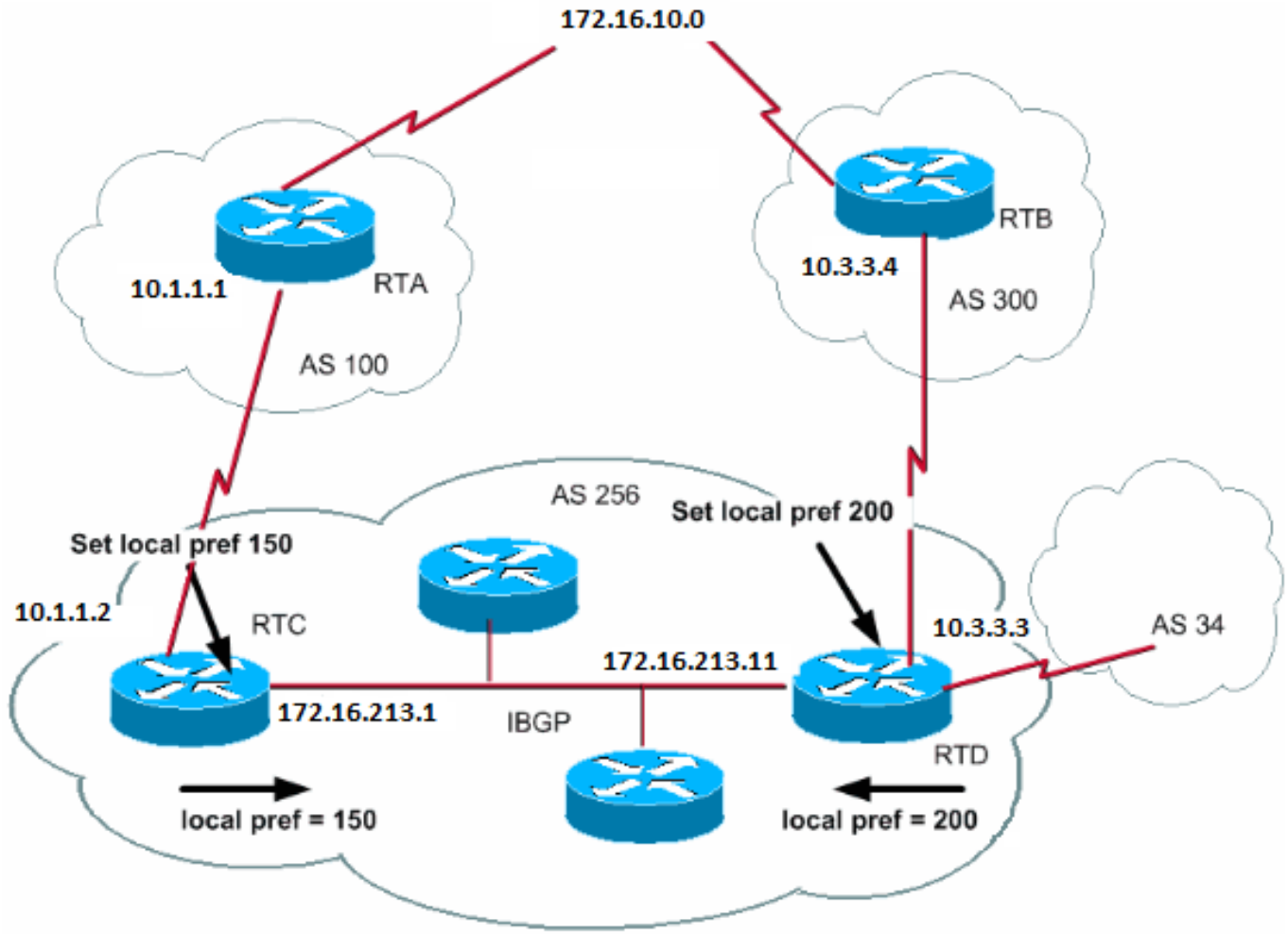
```
RTC#
router bgp 300
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 route-map setweightin in
 neighbor 10.2.2.2 remote-as 200
 neighbor 10.2.2.2 route-map setweightin in
...
```

```
ip as-path access-list 5 permit ^100$
...

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

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

route-map setweightin permit 20
 set weight 100

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



Opmerking: U kunt het gewicht aanpassen zodat het MPLS VPN BGP-pad de voorkeur heeft met het IGP-pad als back-up.

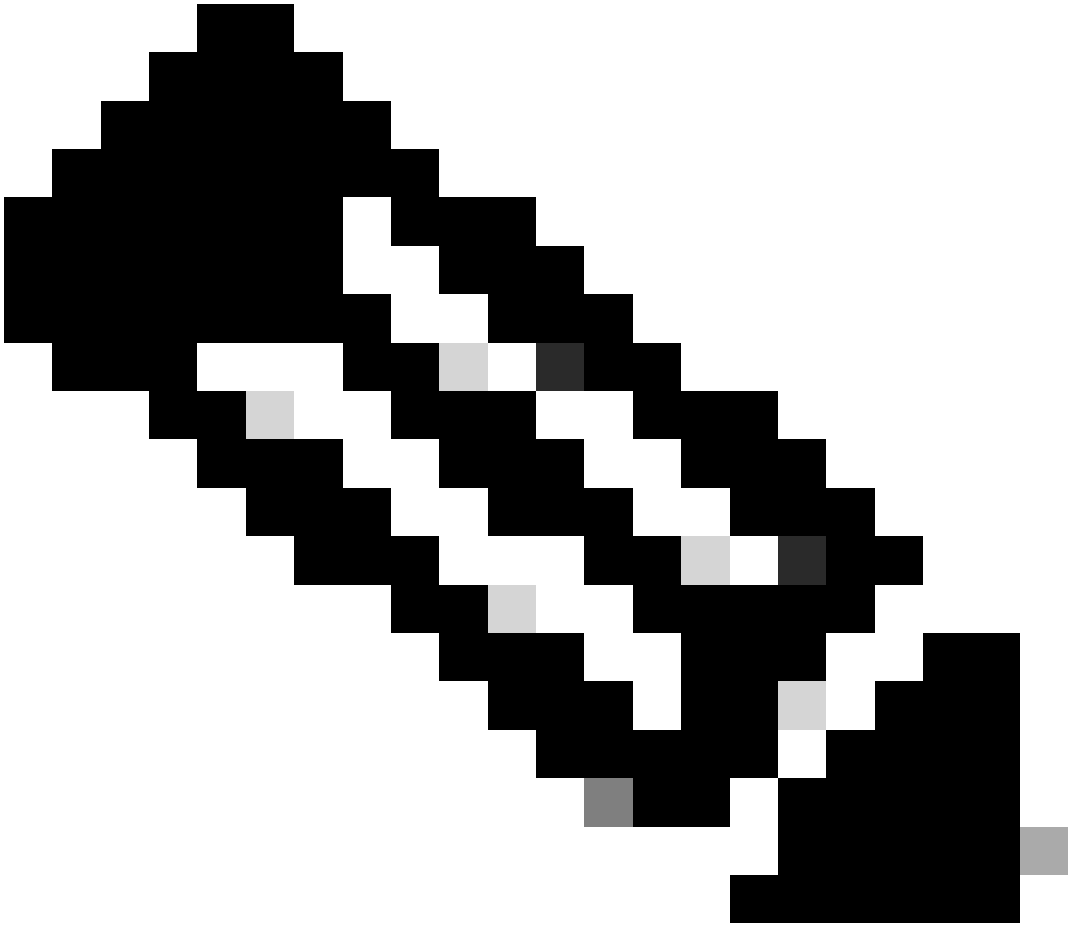
Kenmerk local-preference



Lokale voorkeur (‘local-preference’) is een aanwijzing voor het AS over welk pad de voorkeur heeft om het AS te verlaten voor het bereiken van een bepaald netwerk. Een pad met een hogere local-preference heeft meer de voorkeur. De standaardwaarde voor local-preference is 100.

In tegenstelling tot het kenmerk weight, dat alleen relevant is voor de lokale router, is local-preference een kenmerk dat alle routers in hetzelfde AS uitwisselen.

U stelt de lokale voorkeur in door de opdracht bgp default local-preference waarde uit te voeren. U kunt de lokale voorkeur ook instellen met routekaarten, zoals te zien is in het voorbeeld in deze sectie:



Opmerking: het is noodzakelijk om een zachte reset uit te voeren (dat wil zeggen, het bgp-proces op de router te wissen), zodat de wijzigingen in aanmerking kunnen worden genomen. Om het bgp proces te wissen, gebruik het clear ip bgp [soft][in/out] commando waar soft een zachte reset aangeeft en de sessie niet scheurt, en [in/out] specificeert inkomende of uitgaande configuratie. Als in/out niet is opgegeven, worden zowel inkomende als uitgaande sessies gereset.

De opdracht bgp default local-preference bepaalt de lokale voorkeur voor updates vanaf de router die naar peers gaan in hetzelfde AS. In het diagram in deze sectie ontvangt AS256 updates over 172.16.10.0 van twee verschillende kanten van de organisatie. Lokale voorkeur helpt u te bepalen hoe AS256 moet worden verlaten om dat netwerk te bereiken. Stel dat RTD de voorkeur heeft als uitgangspunt. Deze configuratie stelt de lokale voorkeur in voor updates afkomstig van AS300 tot 200 en voor updates afkomstig van AS100 tot 150:

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

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

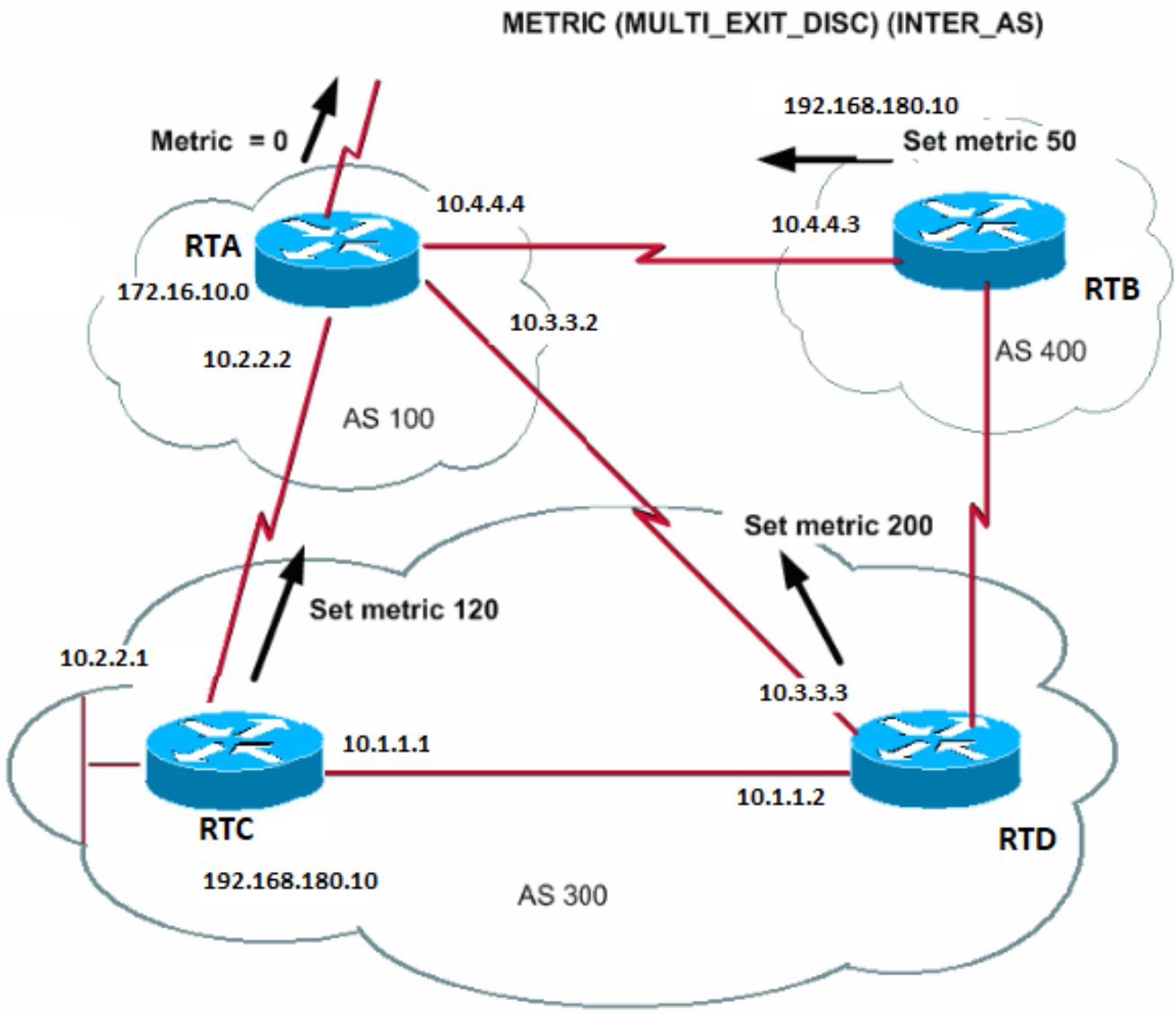
In deze configuratie stelt RTC de lokale voorkeur van alle updates in op 150. Dezelfde OTO stelt de lokale voorkeur van alle updates vast op 200. Binnen AS256 wordt een lokale voorkeur uitgewisseld. Daarom realiseren zowel OTO als OTO zich dat netwerk 172.16.10.0 een hogere lokale voorkeur heeft wanneer de updates van AS300 in plaats van AS100 komen. Al het verkeer in AS256 dat dat netwerk als bestemming heeft brengt met RTD als uitgangspunt over.

Het gebruik van routekaarten biedt meer flexibiliteit. In het voorbeeld in deze sectie krijgen alle updates die RTD ontvangt een lokale voorkeur van 200 wanneer de updates RTD bereiken. Updates die afkomstig zijn van AS34 zijn ook getagd met de lokale voorkeur van 200. Dit label kan overbodig zijn. Daarom kunt u routekaarten gebruiken om specifieke updates te specificeren die met een specifieke lokale voorkeur moeten worden getagd. Hierna volgt een voorbeeld:

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

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

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

Bij deze configuratie heeft elke update die afkomstig is van AS300 een lokale voorkeur van 200. Alle andere updates, zoals updates die afkomstig zijn van AS34, hebben een waarde van 150.

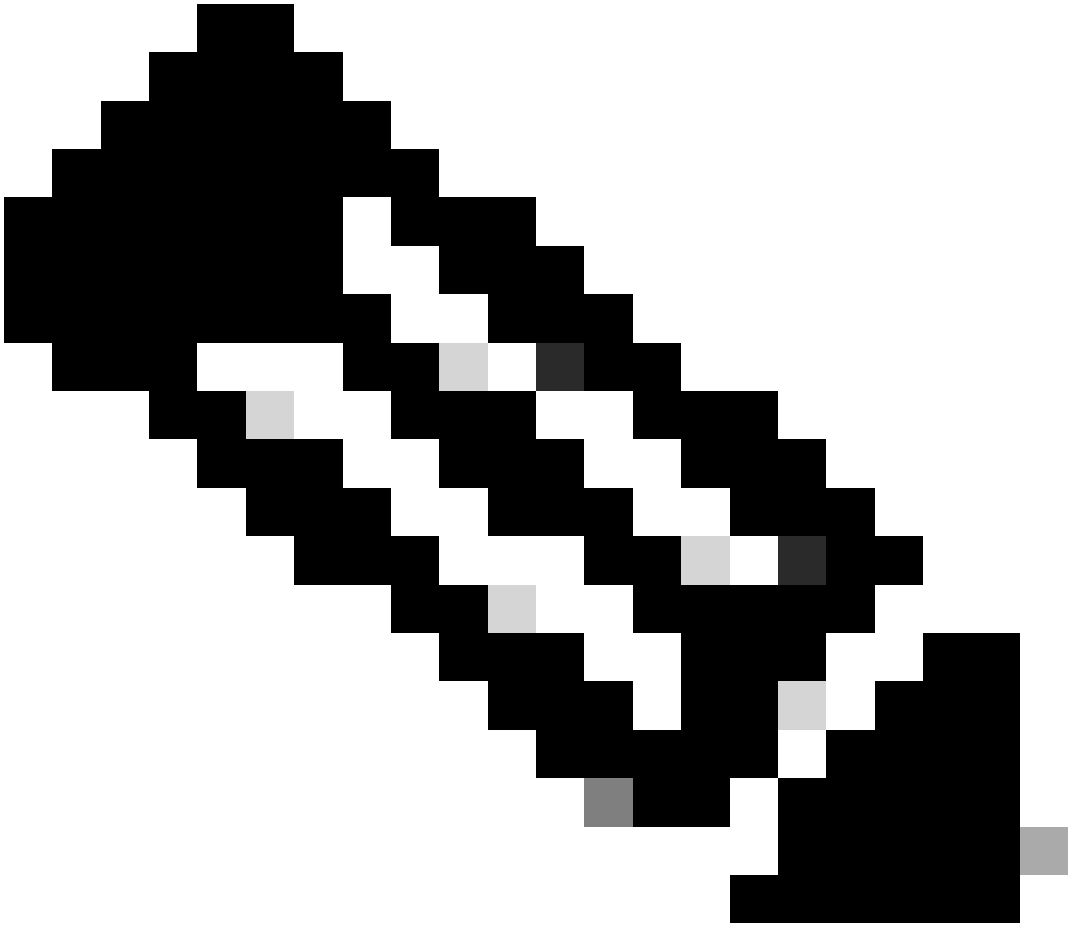
Kenmerk metric



Het kenmerk metric wordt ook MULTI_EXIT_DISCRIMINATOR, MED (BGP4) of INTER_AS (BGP3) genoemd. Het kenmerk is een aanwijzing voor externe neighbors over de padvoorkeur naar een AS. Het kenmerk is een dynamische manier om een ander AS te beïnvloeden wat betreft de manier waarop een bepaalde route wordt bereikt wanneer er meerdere ingangspunten voor het betreffende AS zijn. Een kenmerk metric met een lagere waarde krijgt de voorkeur.

Anders dan het kenmerk local-preference wordt metric uitgewisseld tussen autonome systemen. Er wordt een kenmerk metric doorgegeven naar een AS, maar deze verlaat het AS niet. Wanneer een update het AS binnenkomt met een bepaald kenmerk metric, dan wordt dit kenmerk metric gebruikt om beslissingen te nemen binnen het AS. Wanneer dezelfde update wordt doorgegeven aan een derde AS, keert die metriek terug naar 0. Het diagram in deze sectie toont de reeks metriek. De standaardwaarde voor metric is 0.

Tenzij een router andere richtingen ontvangt, vergelijkt de router de metric-kenmerken voor paden van neighbors in hetzelfde AS. Om de router metric-kenmerken van neighbors uit andere autonome systemen te laten vergelijken, moet de speciale configuratieopdracht bgp always-compare-med op de router worden uitgevoerd.



Opmerking: Er zijn twee BGP-configuratieopdrachten die de MED-gebaseerde (Multi-Exit Discriminator) padselectie kunnen beïnvloeden. Deze opdrachten zijn de opdracht bgp deterministic-med en de opdracht bgp always-compare-med. Het uitvoeren van de opdracht bgp deterministic-med zorgt ervoor dat de MED-variabele wordt vergeleken bij de routekeuze wanneer verschillende peers aankondigingen doen in hetzelfde AS. Het uitvoeren van de opdracht bgp always-compare-med zorgt ervoor dat de MED wordt vergeleken voor paden van neighbors in verschillende autonome systemen. De opdracht bgp always-compare-med komt van pas wanneer meerdere serviceproviders of ondernemingen het eens zijn over een uniform beleid voor het instellen van MED. Raadpleeg Verschillen tussen de opdrachten bgp deterministic-med en bgp always-compare-med om te begrijpen hoe deze opdrachten de BGP-padselectie beïnvloeden.

In het diagram in deze sectie krijgt AS100 informatie over netwerk 192.168.180.10 via drie verschillende routers: RTC, RTD en RTB. RTC en RTD bevinden zich in AS300, en RTB in AS400.

In dit voorbeeld wordt de AS-padvergelijking op RTA genegeerd door de opdracht bgp bestpath as-path ignore . De configuratie is bedoeld om BGP te dwingen door te gaan naar het volgende kenmerk om routes te vergelijken (in dit geval metric of MED). Als de opdracht is weggelaten,

kan de BGP route 192.168.180.10 van router RTC installeren, omdat dat de kortste AS-Path heeft.

Stel dat je de maatstaf die van RTC komt hebt ingesteld op 120, de maatstaf die van RTD naar 200 komt, en de maatstaf die van RTB naar 50 komt. Standaard vergelijkt een router metriek die afkomstig is van buren in hetzelfde AS. Daarom kan RTA alleen het metric-kenmerk afkomstig van RTC vergelijken met het metric-kenmerk afkomstig van RTD. RTA kiest RTC als de beste volgende hop omdat 120 minder dan 200 is. Wanneer RTA een update krijgt van RTB met metrische 50, kan RTA de metriek niet vergelijken met 120 omdat RTC en RTB in verschillende ASs zijn. RTA moet kiezen op basis van andere kenmerken.

Om RTA te dwingen de metric-kenmerken te vergelijken, moet de opdracht bgp always-compare-med worden uitgevoerd op RTA. Deze configuraties illustreren dit proces:

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

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

route-map setmetricout permit 10
   set metric 120

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

route-map setmetricout permit 10
 set metric 200

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

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

Met deze configuraties kiest RTA RTC als volgende hop, met inachtneming van het feit dat alle andere kenmerken hetzelfde zijn. Om RTB op te nemen in de metric-vergelijking, moet RTA als volgt worden geconfigureerd:

```
RTA#
router bgp 100
 neighbor 2.2.21 remote-as 300
 neighbor 10.3.3.3 remote-as 300
```

```
 neighbor 10.4.4.3 remote-as 400
 bgp always-compare-med
```

In dit geval kiest RTA RTB als beste volgende hop om netwerk 192.168.180.10 te bereiken.

U kunt ook metriek instellen tijdens de herverdeling van routes in BGP als u de opdracht **default-metriek nummer** geeft.

Veronderstel dat, in het voorbeeld in deze sectie, RTB een netwerk via statisch in AS100 injecteert. Hier is de configuratie:

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

ip route 192.168.180.10 255.255.0.0 null 0

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

Kenmerk community

Het attribuut community is een transitief, optioneel attribuut in het bereik van 0 tot 4.294.967.200. Het communautaire attribuut is een manier om bestemmingen in een bepaalde gemeenschap te groeperen en routeringsbesluiten toe te passen die die gemeenschappen aanpassen. De routingbeslissingen zijn onder andere accepteren, voorkeur geven en herdistribueren.

U kunt routekaarten gebruiken om de community-kenmerken in te stellen. De opdracht set **voor routekaarten heeft de volgende syntaxis:**

<#root>

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

Een aantal vooraf gedefinieerde, bekende community's die u in deze opdracht kunt gebruiken zijn:

- **no-export**– Kondig niets aan bij eBGP-peers. Houd deze route binnen een AS.

- **no-advertise**– Kondig deze route niet aan bij peers, zowel intern als extern.

- **internet**– Kondig deze route niet aan bij de internetcommunity. Elke router maakt deel uit van deze community.

- **local-as**– Gebruik in confederatiescenario's om te voorkomen dat pakketten buiten het lokale AS worden verzonden.

Hier ziet u twee voorbeelden van routekaarten die de community instellen:

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

of

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

Als u het trefwoord additive niet instelt, vervangt 200 elke oude community die reeds bestaat. Als u het trefwoord additive gebruikt, wordt 200 aan de community toegevoegd. Zelfs als u het community-kenmerk instelt, wordt dit kenmerk niet standaard naar neighbors verzonden. Om het kenmerk naar een neighbor te verzenden, moet deze opdracht worden gebruikt:

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

Hierna volgt een voorbeeld:

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

In Cisco IOS-softwarerelease 12.0 en hoger kunt u gemeenschappen in drie verschillende formaten configureren: decimaal, hexadecimaal en AA:NN. Standaard gebruikt Cisco IOS-software de oudere decimale indeling. Om in AA:NN te configureren en weer te geven, geeft u de opdracht **ip bgp-community new-**global configuration formatteren uit. Het eerste deel van AA:NN vertegenwoordigt het AS-nummer, en het tweede deel vertegenwoordigt een nummer van 2 bytes.

Hierna volgt een voorbeeld:

Zonder de Global Configuration-opdracht **ip bgp-community new-format** zal door het uitvoeren van de opdracht **show ip bgp 10.6.0.0 de kenmerkwaarde community in decimale indeling worden weergegeven.** In dit voorbeeld wordt de waarde van het attribuut community weergegeven als `6553620`.

<#root>

```
Router#
```

show ip bgp 10.6.0.0

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

Community: 6553620

Voer nu de Global Configuration-opdracht ip bgp-community new-format uit op deze router.

```
<#root>

Router#
```

configure terminal

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

ip bgp-community new-format

```
Router(config)#
```

exit

Met **hun bgp-community new**-formatglobal configuratie commando wordt de community waarde weergegeven in AA:NN formaat. De waarde wordt weergegeven als `100:20` in de uitvoer van **de** opdracht **ip bgp 10.6.0.0** in dit voorbeeld:

<#root>

```
Router#
```

show ip bgp 10.6.0.0

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

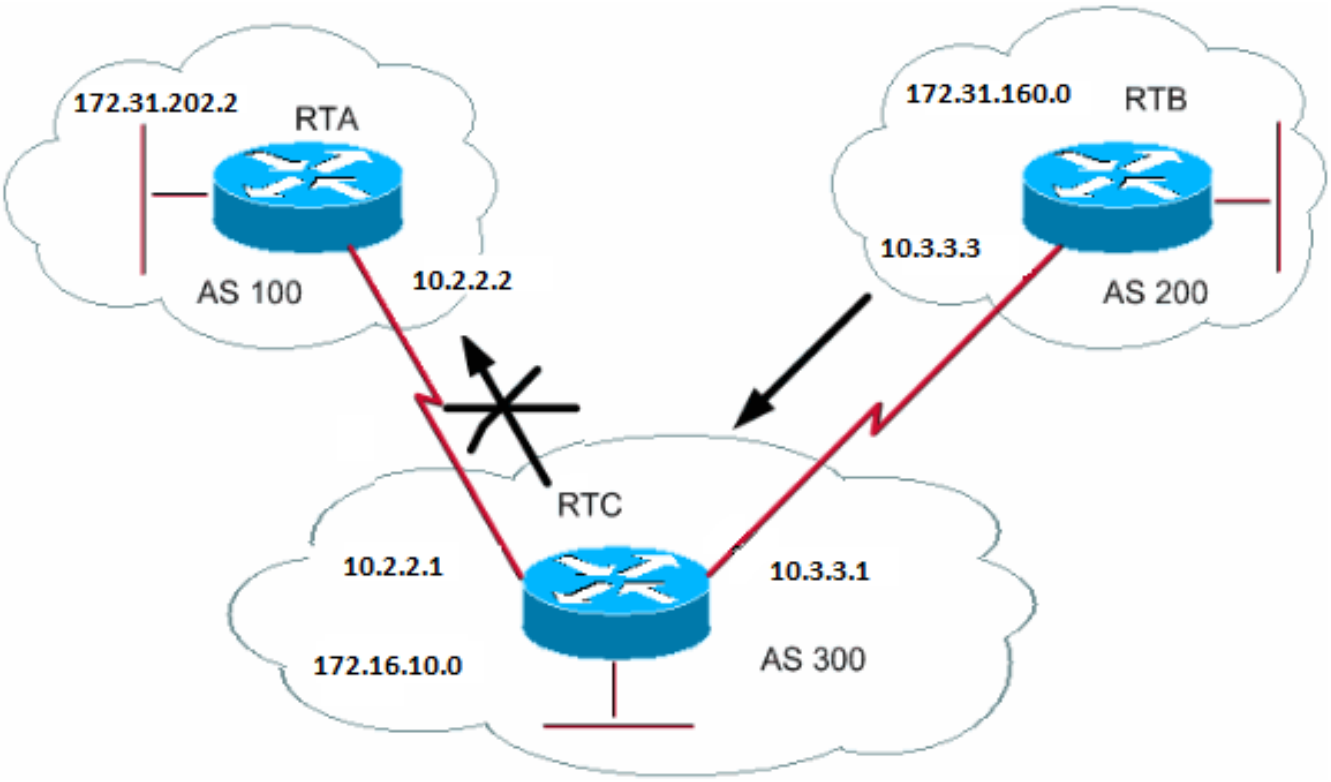
Community: 100:20

BGP-casestudy’s 3

BGP-filter

Met verschillende filtermethoden kunt u het verzenden en ontvangen van BGP-updates controleren. U kunt BGP-updates filteren op basis van route-informatie, of op basis van padinformatie of community’s. Met alle methoden worden dezelfde resultaten bereikt. De keuze voor een bepaalde methode is afhankelijk van de specifieke netwerkconfiguratie.

Routefilter



Om de routinginformatie die de router leert of aankondigt te beperken, kunt u BGP filteren met behulp van routingupdates naar of van een bepaalde neighbor. U definieert een toegangslijst en past de toegangslijst toe op de updates voor of van een neighbor. Voer deze opdracht uit in de modus voor routerconfiguratie:

<#root>

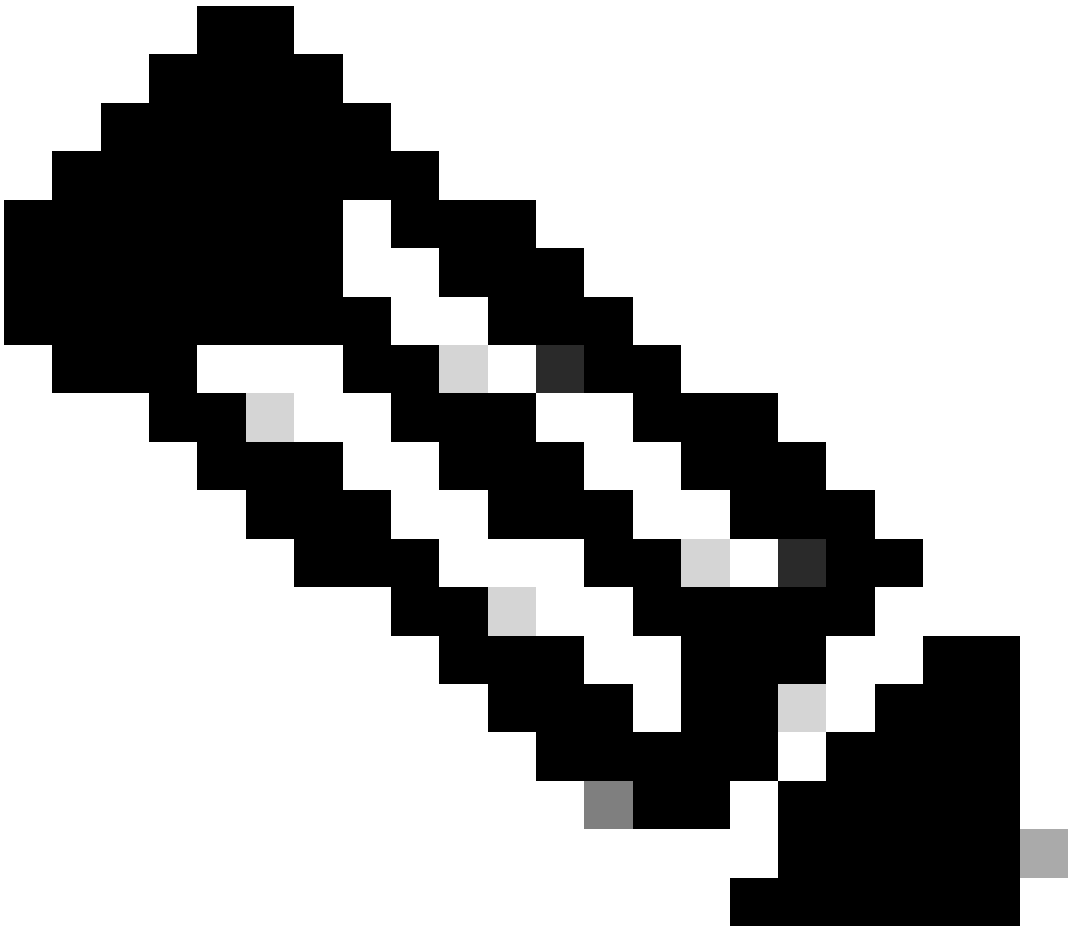
neighbor {ip-address | peer-group-name} distribute-list access-list-number {in | out}

In dit voorbeeld is RTB de oorsprong van netwerk 172.31.160.0 en stuurt de update naar RTC. Als RTC het doorgeven van de updates aan AS100 wil stoppen, moet u een toegangslijst definiëren om de betreffende updates te filteren en de toegangslijst toepassen tijdens de communicatie met RTA:

```
RTC#
router bgp 300
 network 172.16.10.0
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 distribute-list 1 out

access-list 1 deny 172.31.160.0 0.0.255.255

access-list 1 permit 0.0.0.0 255.255.255.255

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

Het gebruik van toegangslijsten kan lastig zijn wanneer u te maken heeft met supernets die conflicten kunnen veroorzaken.

Stel dat RTB in het voorbeeld in deze sectie verschillende subnetten heeft van 160.10.x.x. Uw doel is updates te filteren en alleen 192.168.160.0/8 aan te kondigen.



Opmerking: De notatie /8 betekent dat u 8 bits van het subnetmasker gebruikt, beginnend vanaf de linkerkant van het IP-adres. Dit adres is gelijk aan 192.168.160.0 255.0.0.0.

Het commando access-list 1 permit 192.168.160.0 0.255.255.25 5maakt 192.168.160.0/8, 192.168.160.0/9 enzovoort mogelijk. Om de update te beperken tot enkel 192.168.160.0/8, moet een uitgebreide toegangslijst met de volgende indeling worden gebruikt:

<#root>

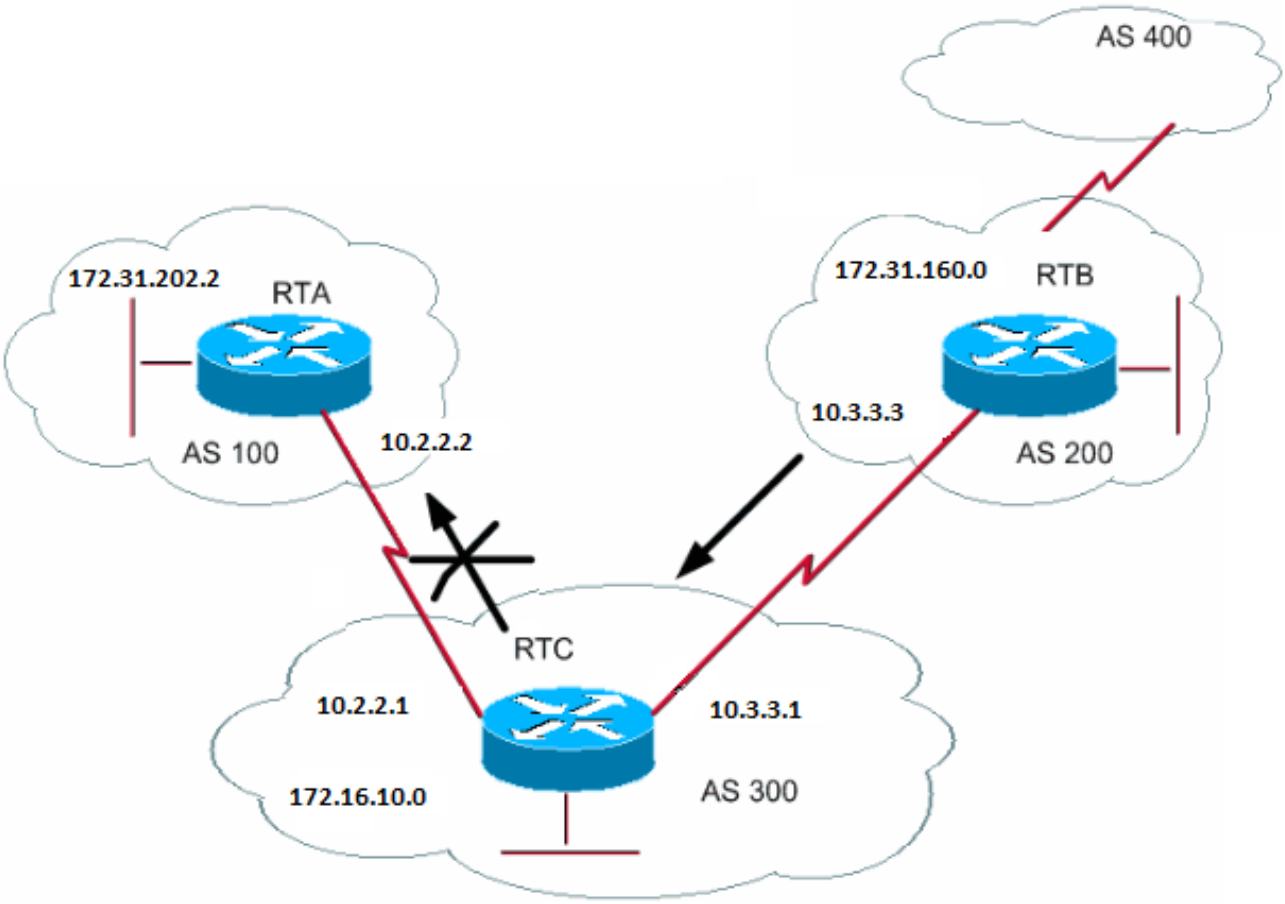
access-list 101 permit ip 192.168.160.0 0.255.255.255 255.0.0.0 0.0.0.0.

Deze lijst staat alleen 192.168.160.0/8 toe.

Raadpleeg Blok een of meer netwerken van een BGP-peer voor voorbeeldconfiguraties over het filteren van netwerken van BGP-peers. De methode gebruikt de opdracht **distribueren-lijst** met standaard- en uitgebreide toegangscontrolelijsten (ACL's), evenals de mogelijkheid om de prefixlijst te filteren.

Padfilter

U kunt ook paden filteren.



U kunt een toegangslijst opgeven voor zowel inkomende als uitgaande updates met behulp van de informatie over BGP AS-paden. In het diagram in deze sectie, kunt u updates over 172.31.160.0 blokkeren zodat zij niet naar AS100 gaan. Om de updates te blokkeren, definieert u een toegangslijst op RTC die het verzenden naar AS100 van updates die afkomstig zijn van AS200 verhindert. Voer de volgende opdrachten uit:

<#root>

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

<#root>

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

Dit voorbeeld zorgt ervoor dat RTC geen updates over 172.31.160.0 verzendt naar RTA:

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

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

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

De access-list 1opdracht in dit voorbeeld dwingt de ontkenning van updates met padinformatie die begint met 200 en eindigt met 200. De`^200$`in de opdracht is een "reguliere expressie", waarin`^`"begint met" en`$`betekent "eindigt met". Aangezien RTB updates verstuurt over 172.31.160.0 met padinformatie die begint met 200 en eindigt met 200, komen de updates overeen met de toegangslijst. De toegangslijst weigert deze updates.

De`.*` is een andere reguliere expressie waarin de`.`betekent "elk teken" en de`*` betekent "de herhaling van dat teken". Dus .* vertegenwoordigt willekeurige padinformatie die nodig is om de overdracht van alle andere updates toe te staan.

Wat gebeurt er als je in plaats van`^200$`,`^200` gebruikt? Met een AS400, zoals in het diagram in deze sectie, hebben updates afkomstig van AS400 padinformatie in de vorm (200, 400). Bij deze padinformatie komt 200 eerst en 400 als laatste. Deze updates komen overeen met de toegangslijst`^200`, omdat de padinformatie begint met 200. De toegangslijst verhindert de transmissie van deze updates naar RTA, wat niet het vereiste is.

Om te controleren of u de juiste reguliere expressie hebt geïmplementeerd, geeft u de opdracht **ip bgp regexpregular-expressie uit**. Deze opdracht geeft alle paden weer die overeenkomen met de configuratie van de reguliere expressie.

Reguliere expressie voor AS

In deze sectie wordt uitgelegd hoe een reguliere expressie wordt gemaakt.

Een reguliere expressie is een patroon dat wordt vergeleken met een invoertekenreeks. Wanneer u een reguliere expressie maakt, specificeert u een tekenreeks die moet overeenkomen met de invoer. In het geval van BGP, specificeert u een tekenreeks die bestaat uit padinformatie die moet overeenkomen met de invoer.

In het voorbeeld in het gedeelte **Path Filter**, gaf u de string`^200$`. Je wilde pad informatie die binnenin updates komt overeen met de string om te beslissen.

Een reguliere expressie bestaat uit:

- **Bereik**

  Een bereik is een reeks tekens omsloten door vierkante haakjes. Een voorbeeld is [abcd].

- **Atoom**

  Een atoom is één teken. Hier volgen enkele voorbeelden:

`.`

- The`.` komt overeen met elk teken.

`^`

- Het teken ^ komt overeen met het begin van invoertekenreeks.

`$`

- Het teken $ komt overeen met het einde van de invoertekenreeks.

\

- De\komt overeen met het teken.

_

- The_match een komma (`1`), linkerkant (`{`), rechterkant (`}`), het begin van de invoerstring, het einde van de invoerstring, of een spatie.

•

Kwantor

Een stuk is een van deze symbolen, die na een atoom komt:

*

- Het teken * komt overeen met 0 of meer sequenties van het atoom.

+

- Het teken + komt overeen met 1 of meer sequenties van het atoom.

?

- Het? komt overeen met het atoom of de null-string.

- **Groep**

Een groep bestaat uit 0 of meer aaneengeschakelde kwantoren.

Hier volgen enkele voorbeelden van reguliere expressies:

a*

- Deze expressie geeft aan hoe vaak de letter ‘a’ voorkomt, inclusief geen enkele keer.

```
a+
```

- Deze expressie geeft aan dat de letter ‘a’ ten minste één keer moet voorkomen.

```
ab?a
```

- Deze expressie komt overeen met ‘aa’ of ‘aba’.

```
_100_
```

- Deze expressie betekent via AS100.

```
_100$
```

- Deze expressie geeft aan dat de oorsprong AS100 is.

```
^100 .*
```

- Deze expressie geeft overdracht van AS100 aan.

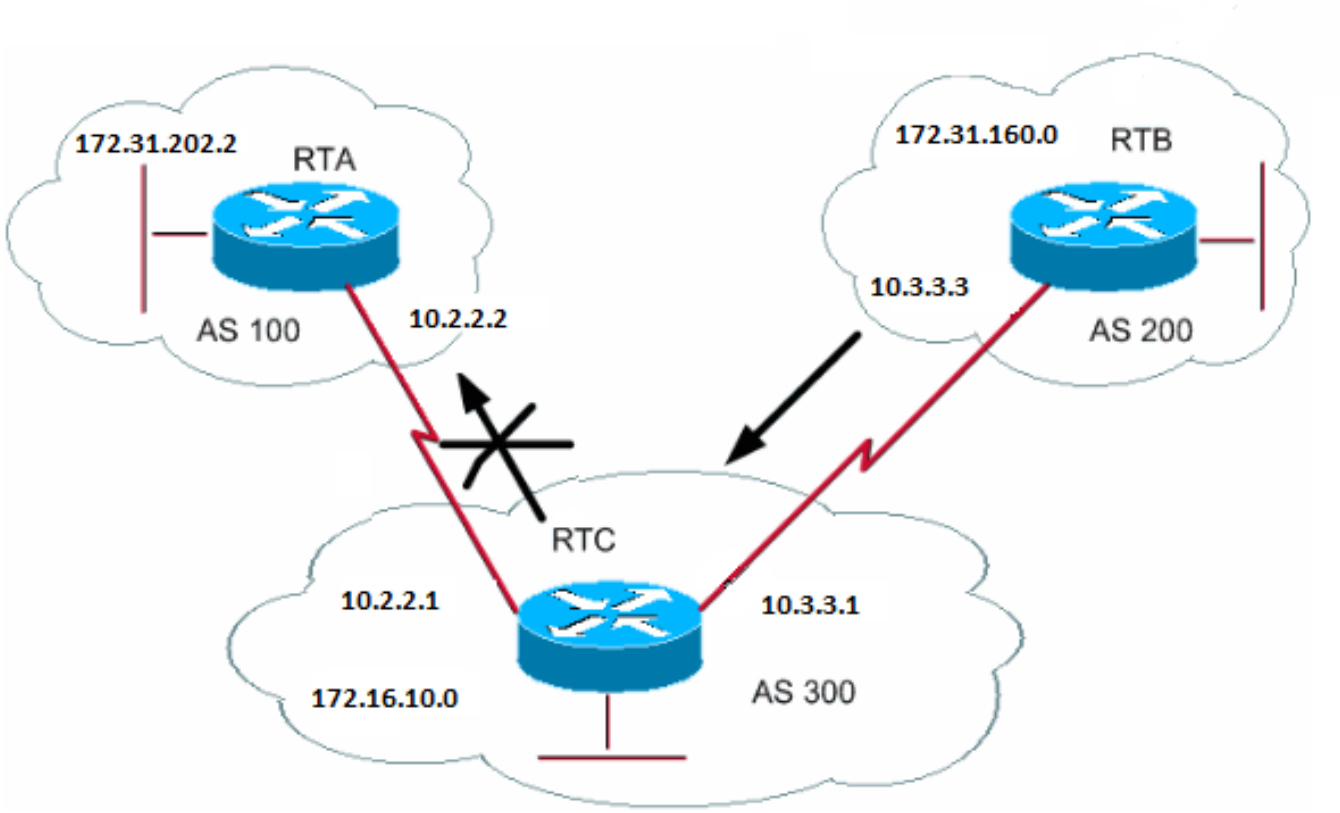
```
^$
```

- Deze expressie geeft dit AS als oorsprong aan.

Raadpleeg Reguliere expressies in BGP gebruiken voor voorbeeldconfiguraties van reguliere expressies en filteren.

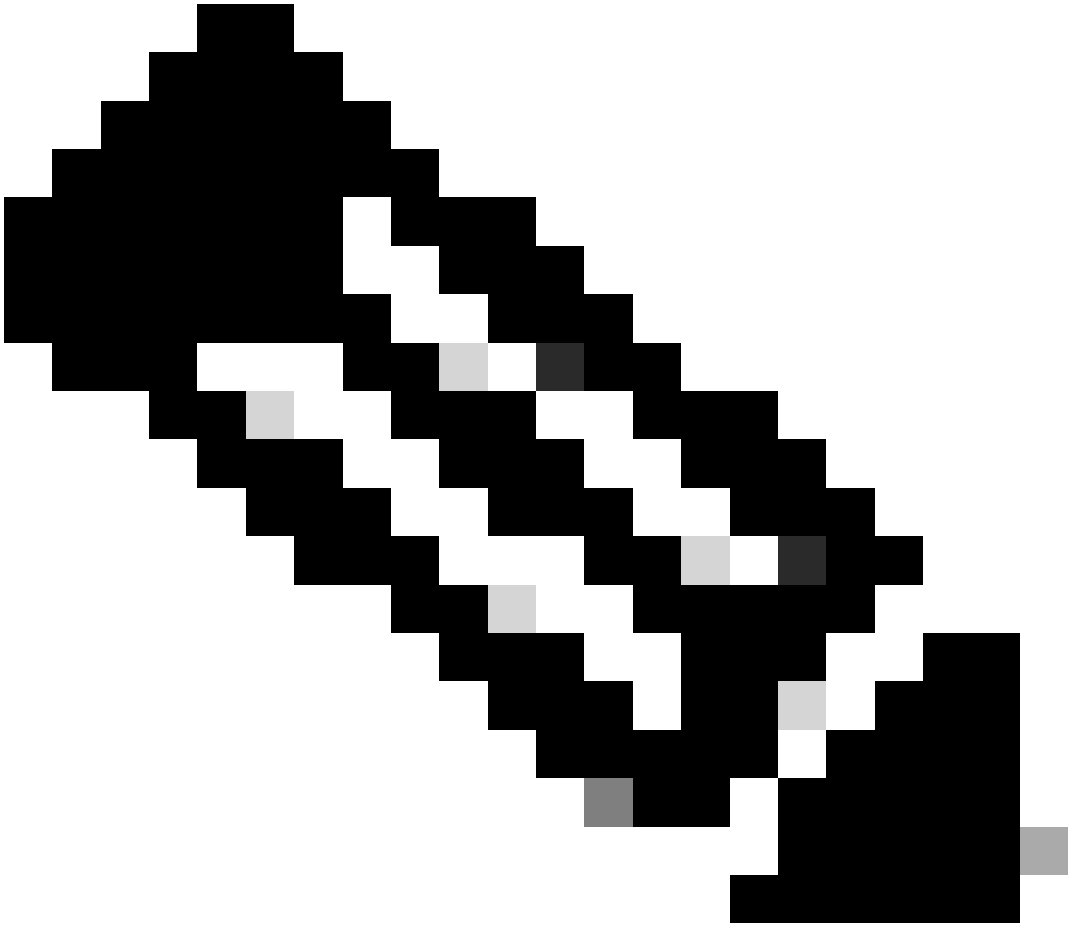
BGP-communityfilter

In dit document zijn het filteren van routes en filteren op AS-pad besproken. Een andere methode is communityfiltering. De sectieCommunity Attribute bespreekt de community en deze sectie geeft een aantal voorbeelden van hoe u de community kunt gebruiken.

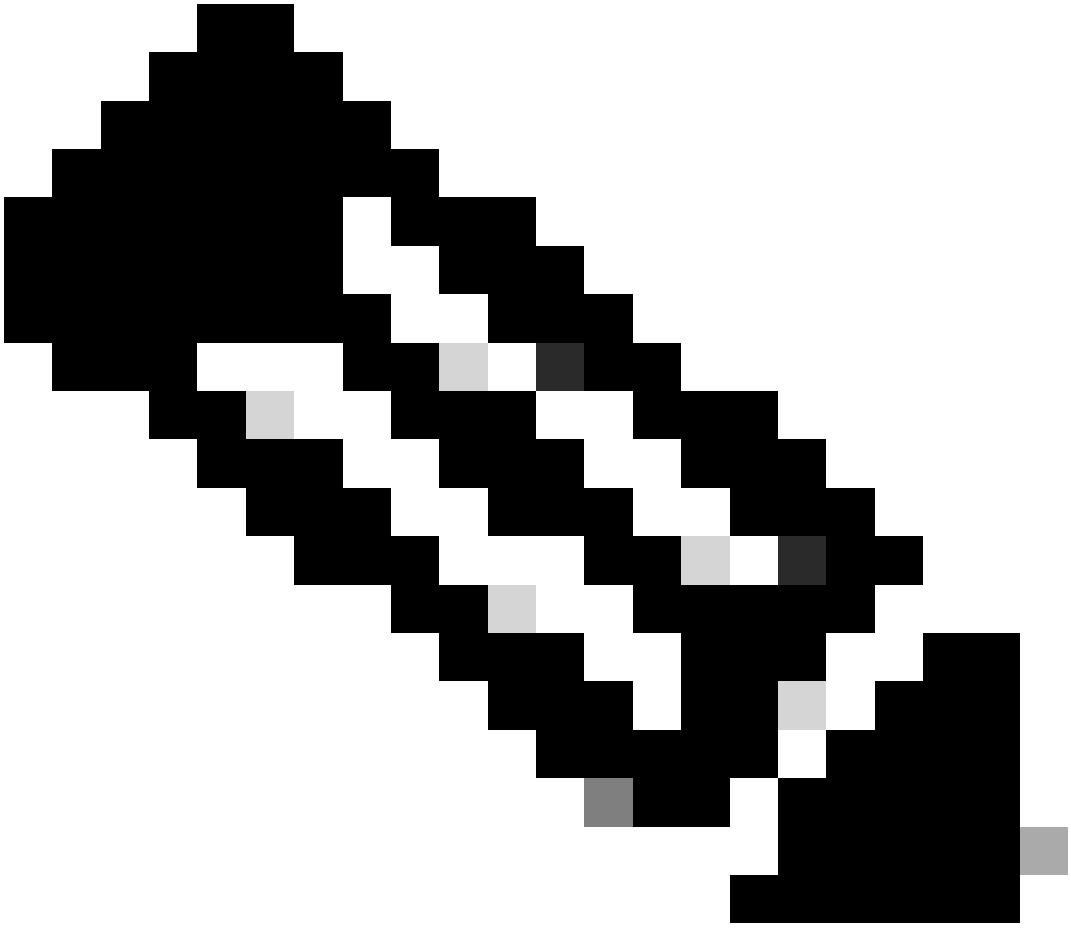


In dit voorbeeld, wilt u dat RTB het kenmerk community zodanig instelt voor de BGP-routes die RTB aankondigt, dat RTC deze routes niet doorgeeft naar de externe peers. Gebruik het no-exportcommunity-kenmerk.

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

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

access-list 1 permit 0.0.0.0 255.255.255.255
```



Opmerking: in dit voorbeeld wordt de opdrachtroute-map setcommunity gebruikt om de community in te stellen op no-export.



Opmerking: De **neighbor send-community** opdracht is nodig om deze eigenschap naar RTC te verzenden.

Wanneer RTC de updates krijgt met het kenmerk NO_EXPORT, dan verspreidt RTC de updates niet naar externe peer RTA.

In dit voorbeeld heeft RTB het attribuut community ingesteld op**100 200 additive** . Met deze actie wordt de waarde 100 200 toegevoegd aan de huidige communautaire waarde vóór transmissie naar RTC.

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

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

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

access-list 2 permit 0.0.0.0 255.255.255.255
```

Een communitylijst is een groep community's die wordt gebruikt in een match-clausule van een routekaart. Met de communitylijst kunt filteren op kenmerken of kenmerken instellen op basis van verschillende lijsten met communitynummers.

<#root>

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

U kunt deze routekaart bijvoorbeeld definiëren, match-on-community:

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

!--- The community list number is 10.

 set weight 20
 ip community-list 10 permit 200 300

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

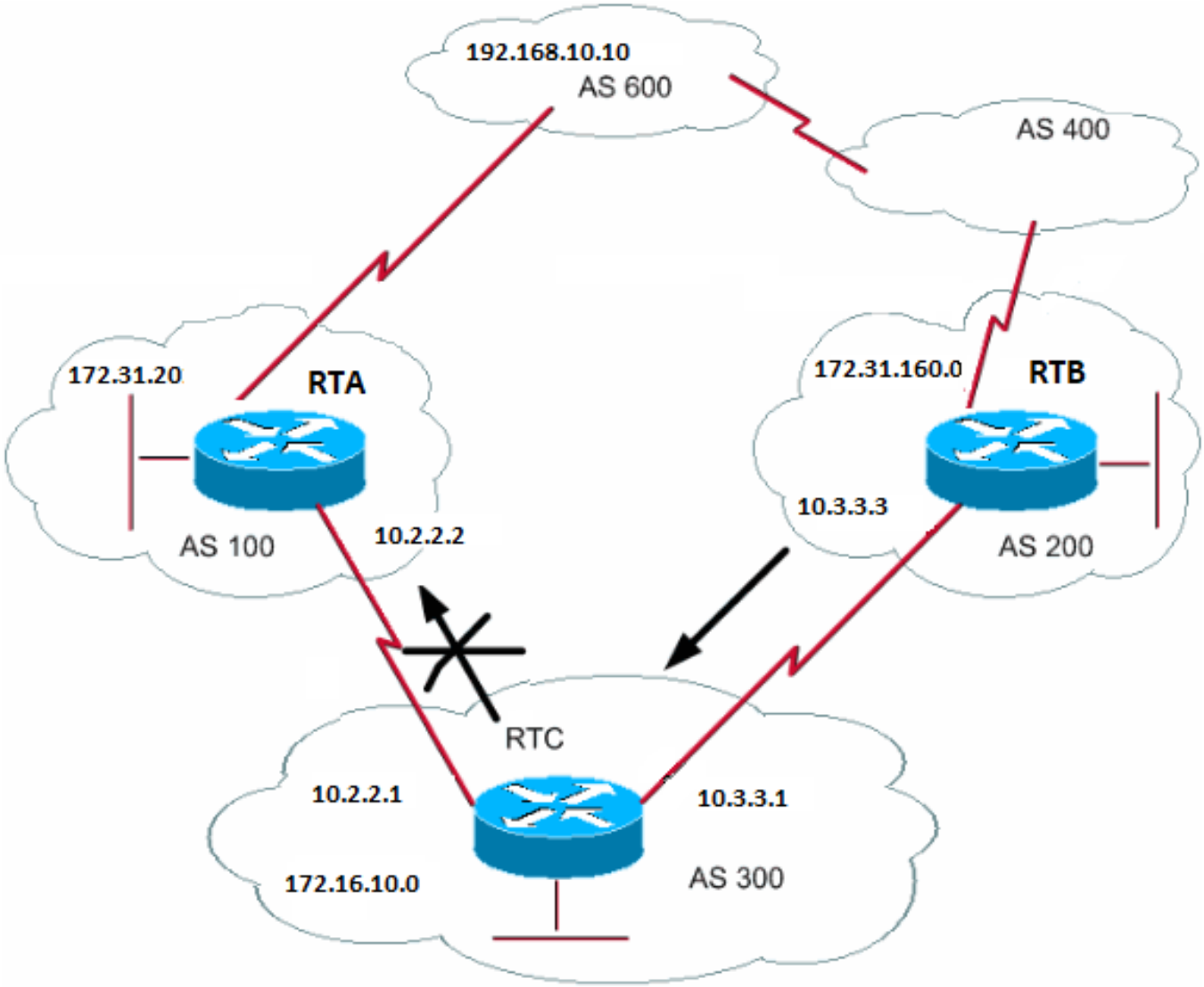
U kunt de communitylijst gebruiken voor het filteren of instellen van bepaalde parameters (zoals weight en metric) in bepaalde updates, gebaseerd op de communitywaarde. In het tweede voorbeeld in deze sectie zond RTB updates naar RTC met een gemeenschap van 100 200. Als RTC het gewicht met die waarden als basis wil instellen, kunt u dit doen:

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

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

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

route-map check-community permit 30
 match community 3

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

In dit voorbeeld komt elke route die 100 heeft in het attribuut community overeen met lijst 1. Het gewicht van deze route is ingesteld op 20. Elke route die slechts 200 als gemeenschap heeft past lijst 2 en heeft een gewicht van 20. Het sleutelwoord zegt **precies** dat de gemeenschap alleen uit 200 bestaat en verder niets. De laatste communitylijst zorgt ervoor dat andere updates niet worden afgewezen. Onthoud dat alles wat niet overeenkomt, standaard wordt afgewezen. Het trefwoord internet geeft alle routes aan omdat alle routes lid zijn van de internetcommunity.

Zie Een upstream-providernetwerk met BGP-communitywaarden configureren en beheren voor meer informatie.

BGP-neighbors en routekaarten



U kunt de opdracht neighbor in combinatie met routekaarten gebruiken voor het filteren of instellen van parameters voor inkomende en uitgaande updates.

Routekaarten gekoppeld aan de instructie neighbor hebben geen invloed op inkomende updates bij een overeenkomst op basis van het IP-adres:

<#root>

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

Stel dat u in het diagram in deze sectie RTC van AS200 wilt laten leren over netwerken die lokaal zijn voor AS200 en niets anders. Ook, wilt u het gewicht op de geaccepteerde routes instellen op 20. Gebruik een combinatie van **buurman-** en **as-path-**toegangslijsten:

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

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

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

Alle updates die afkomstig zijn van AS200 hebben padinformatie die begint met 200 en eindigt met 200. Deze updates zijn toegestaan. Alle andere updates worden afgewezen.

Ga ervan uit dat u het volgende wilt:

- Acceptatie van updates die afkomstig zijn van AS200 en een gewicht hebben van 20
- Afwijzing van updates die afkomstig zijn van AS400
- Een gewicht van 10 voor andere updates

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

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

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

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

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

Deze verklaring plaatst een gewicht van 20 voor updates die aan AS200 lokaal zijn. De verklaring plaatst ook een gewicht van 10 voor updates die achter AS400 zijn en laat vallen updates die uit AS400 komen.

Gebruik van de opdracht set as-path prepend

In sommige situaties moet u de padinformatie manipuleren om het BGP-besluitvormingsproces te kunnen manipuleren. De opdracht die u voor een routekaart gebruikt is:

<#root>

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

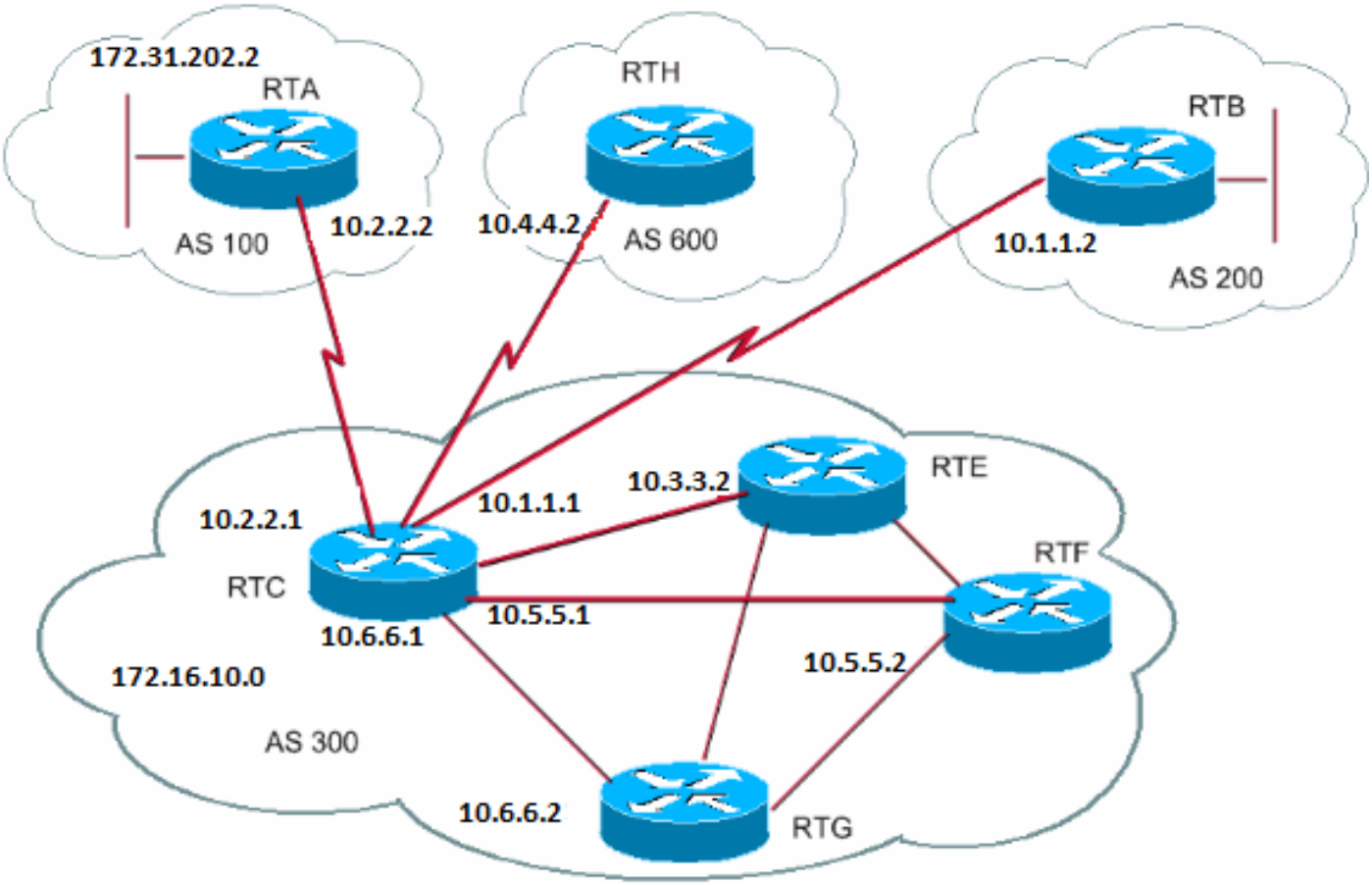
Stel dat RTC in het diagram in de sectieBGP Buren en routekaarten zijn eigen netwerk 172.16.10.0 adverteert naar twee verschillende AS's, AS100 en AS200. Wanneer de informatie wordt doorgegeven aan AS600, hebben de routers in AS600 netwerkbereikbaarheidsinformatie over 172.16.10.0 via twee verschillende routes. De eerste route is via AS100 met pad (100, 300), en de tweede is via AS400 met pad (400, 200, 300). Wanneer alle andere kenmerken hetzelfde zijn, selecteert AS600 het kortste pad en kiest de route via AS100.

AS300 krijgt al het verkeer via AS100. Als u deze beslissing van de AS300 wilt beïnvloeden, kunt u ervoor zorgen dat het pad door AS100 langer lijkt dan het pad dat door AS400 loopt. Dit kunt u doen als u de AS-nummers wilt toevoegen aan de huidige padinformatie die wordt geadverteerd voor AS100. Een veel gebruikte werkwijze is om je eigen AS-nummer op deze manier te herhalen:

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

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

Als gevolg van deze configuratie ontvangt AS600 updates over 172.16.10.0 via AS100 met padinformatie van: (100, 300, 300, 300). Deze padinformatie is langer dan de (400, 200, 300) die AS600 van AS400 heeft ontvangen.

BGP-peergroepen



Een BGP-peergroep is een groep BGP-neighbors met hetzelfde updatebeleid. Routekaarten, distributielijsten en filterlijsten bepalen meestal het updatebeleid. U definieert niet hetzelfde beleid voor elke afzonderlijke buur; in plaats daarvan definieert u een peer group name en wijst u dit beleid toe aan de peer group.

Leden van de peergroep nemen alle configuratieopties van de peergroep over. U kunt leden ook zodanig configureren dat deze opties worden overschreven wanneer de opties geen invloed hebben op uitgaande updates. U kunt alleen opties overschrijven die zijn ingesteld voor inkomend verkeer.

Om een peergroep te definiëren, voert u de volgende opdracht uit:

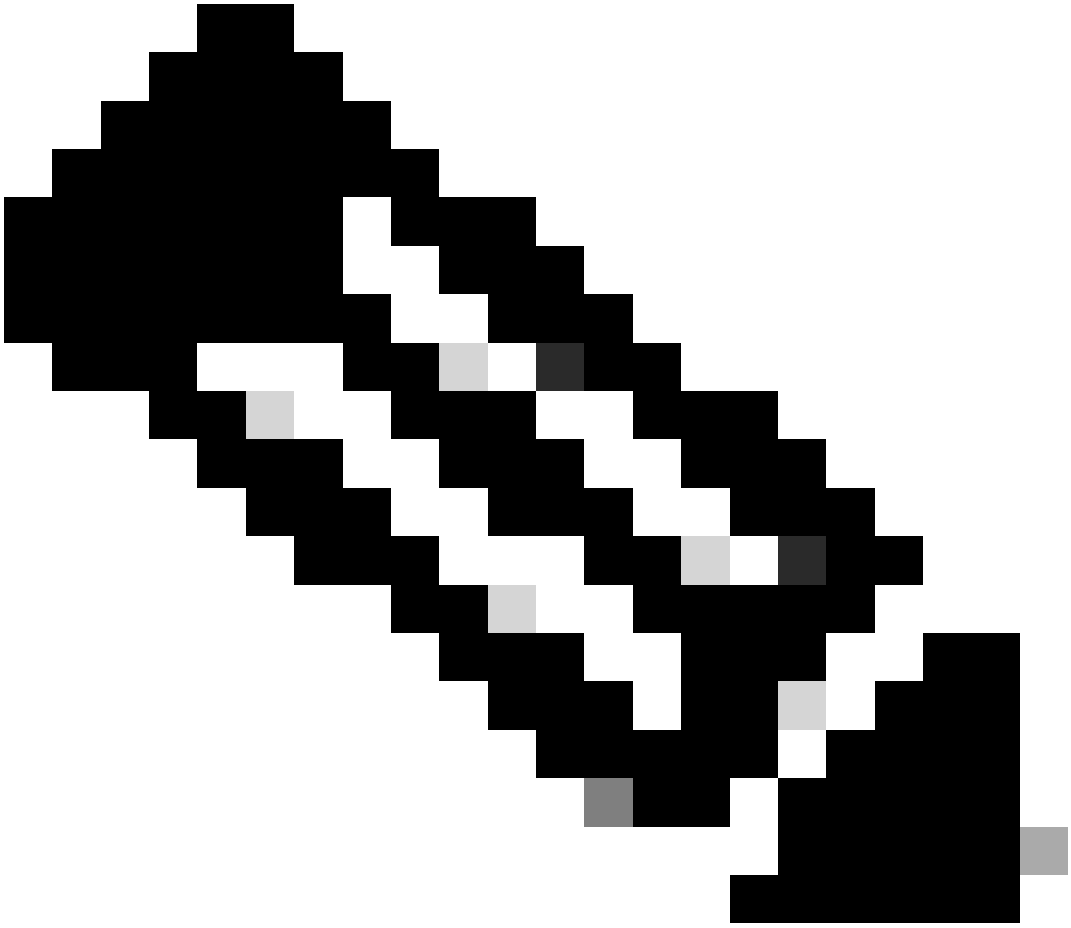
<#root>

neighbor peer-group-name peer-group

Dit voorbeeld past peergroepen toe op interne en externe BGP-neighbors:

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

Deze configuratie definieert een peergroep met de naam internalmap. De configuratie definieert een aantal beleidsregels voor de groep, zoals een routekaart **SETMETRIC** om de metriek in te stellen op 5 en twee verschillende filterlijsten, 1 en 2. De configuratie past de peer groep toe op alle interne buren, RTE, RTF en RTG. Daarnaast definieert de configuratie een afzonderlijke filterlijst 3 voor neighbor RTE. Deze filterlijst overschrijft filterlijst 2 binnen de peergroep.

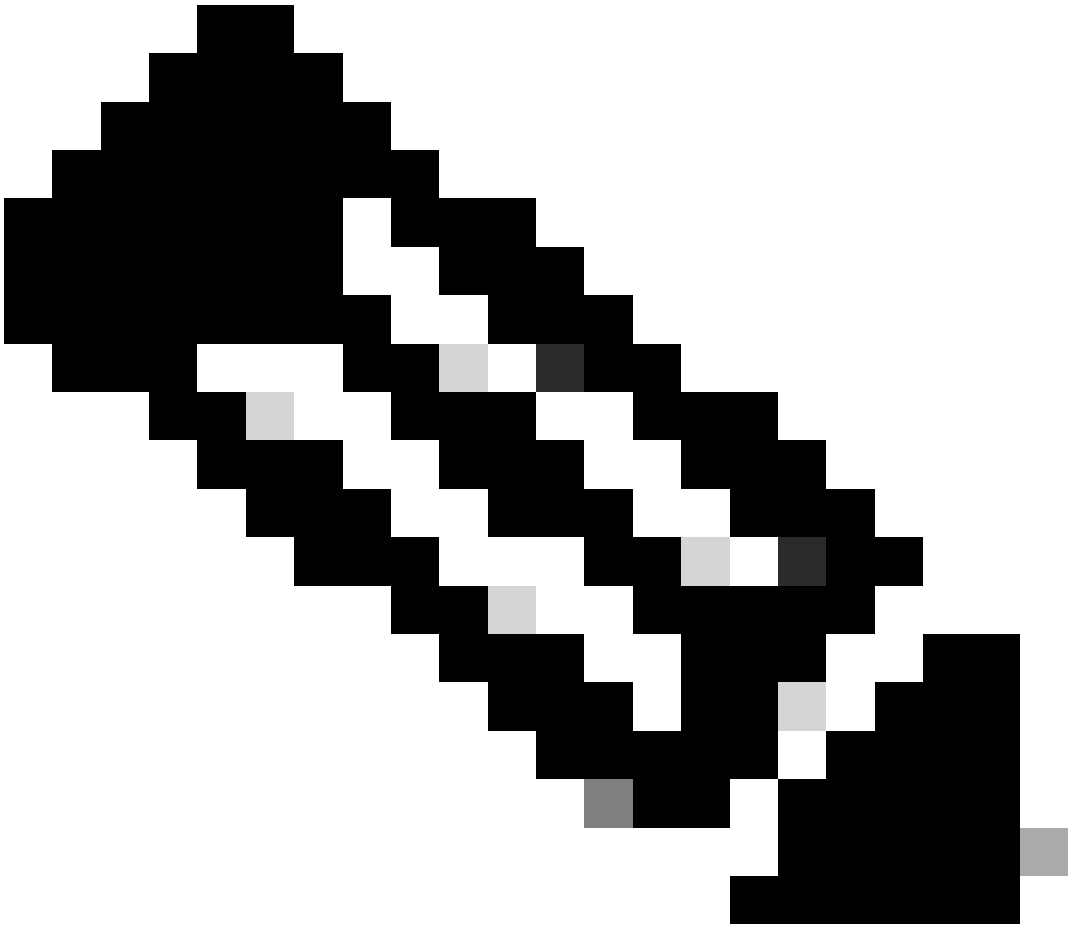


Opmerking: u kunt alleen opties negeren die invloed hebben op inkomende updates.

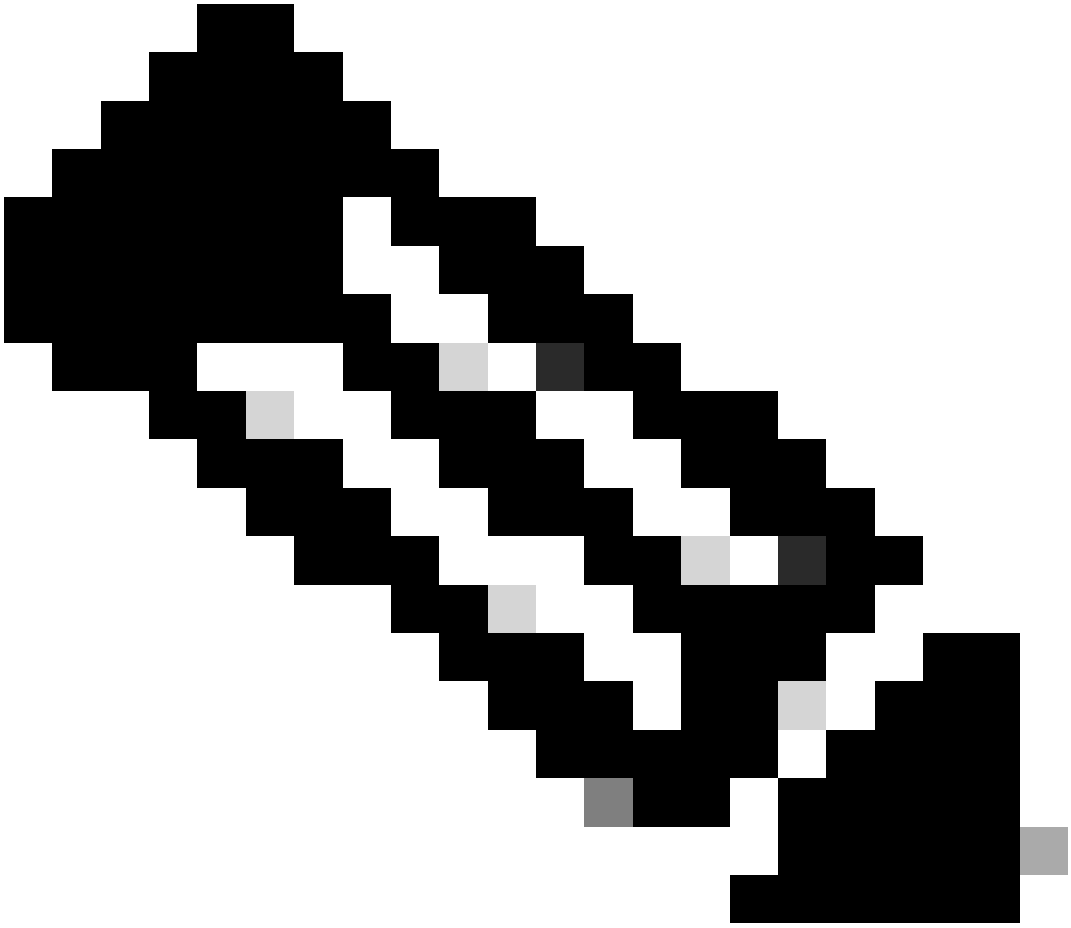
Laten we nu kijken hoe u peergroepen met externe neighbors kunt gebruiken. Met hetzelfde diagram uit deze sectie, configureert u RTC met een peergroep externalmap en past u de peergroep toe op externe neighbors.

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

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



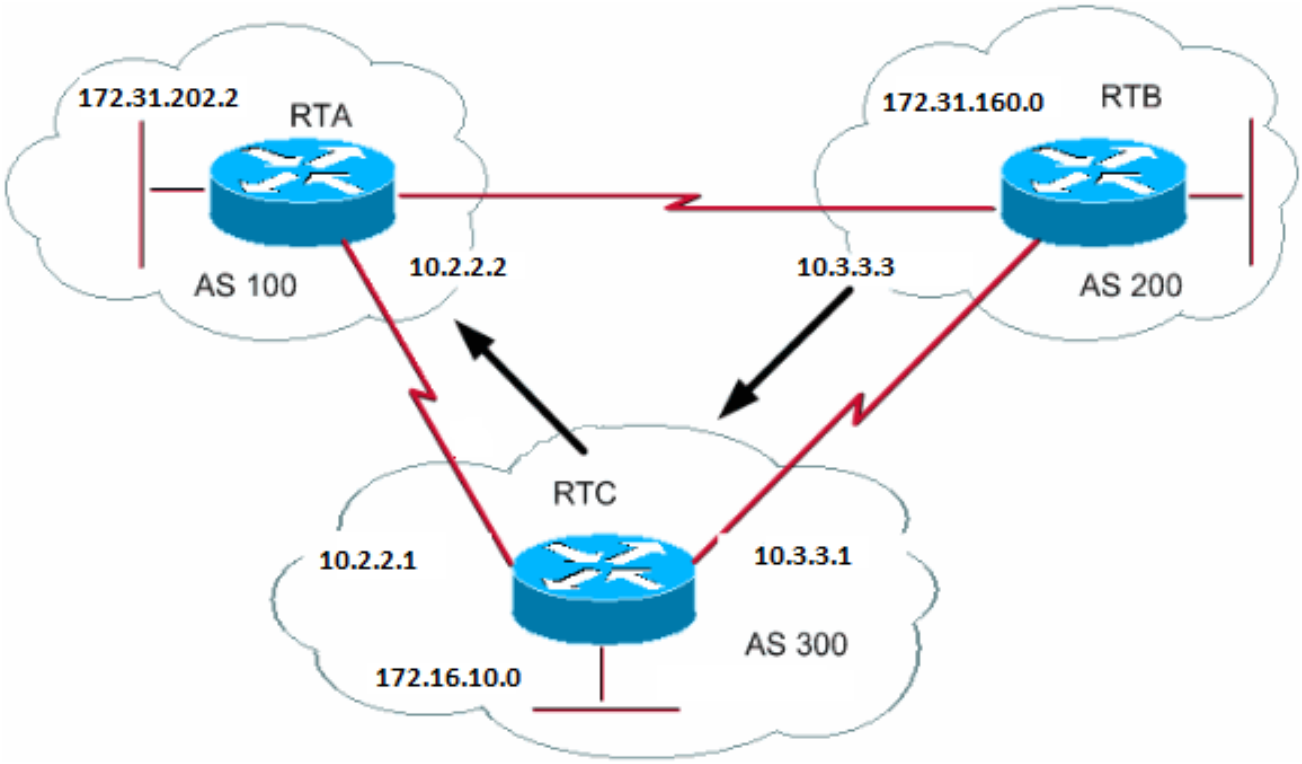
Opmerking: In deze configuraties definieert u de remote-as-instructies buiten de peergroep omdat u verschillende externe autonome systemen moet definiëren. Met het toewijzen van filterlijst 3 overschrijft u ook de inkomende updates van neighbor 10.1.1.2. Raadpleeg BGP-peergroepen voor meer informatie over peergroepen.



Opmerking: In Cisco IOS-softwarerelease 12.0(24)S introduceerde Cisco de functie om BGP-peergroepen dynamisch te updaten. Deze functie is ook beschikbaar in latere Cisco IOS-softwarereleases. De functie introduceert een nieuw algoritme dat dynamisch updategroepen berekent en optimaliseert van neighbors die hetzelfde uitgaande beleid delen. Deze neighbors kunnen dezelfde updateberichten delen. In eerdere releases van Cisco IOS-software werden BGP-updateberichten gegroepeerd op basis van peergroepconfiguraties. Deze methode om updates te groeperen was beperkend voor uitgaand beleid en specifieke sessieconfiguraties. De functie BGP-peergroep dynamisch updaten scheidt de replicatie van updategroepen van de configuratie van peergroepen. Deze scheiding verbetert de convergentietijd en de flexibiliteit van de configuratie van neighbors. Raadpleeg BGP-peergroepen dynamisch updaten voor meer informatie.

BGP-casestudy's 4

CIDR- en geaggregeerde adressen



Een van de belangrijkste verbeteringen van BGP4 ten opzichte van BGP3 is een Classless Interdomain Routing (CIDR). CIDR of supernetting is een nieuwe manier om te kijken naar IP-adressen. Met CIDR, is er geen notie van klassen, zoals klasse A, B, of C. Het netwerk 192.168.213.0 was ooit een illegaal klasse C-netwerk. Het netwerk is nu een wettelijk supernet, 192.168.213.0/16. 16 staat voor het aantal bits in het subnetmasker wanneer u van uiterst links van het IP-adres telt. Deze weergave is vergelijkbaar met 192.168.213.0 255.255.0.0.

U gebruikt aggregaten om de omvang van de routingtabellen te minimaliseren. Aggregatie is het proces waarbij de kenmerken van verschillende routes zodanig worden gecombineerd dat aankondiging van één route mogelijk is. In dit voorbeeld genereert RTB netwerk 172.31.160.0. U vormt RTC om een supernet van die route 192.168.160.0 aan RTA te propageren:

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

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

RTC geeft het geaggregeerde adres 192.168.160.0 door naar RTA.

Opdrachten voor aggregeren

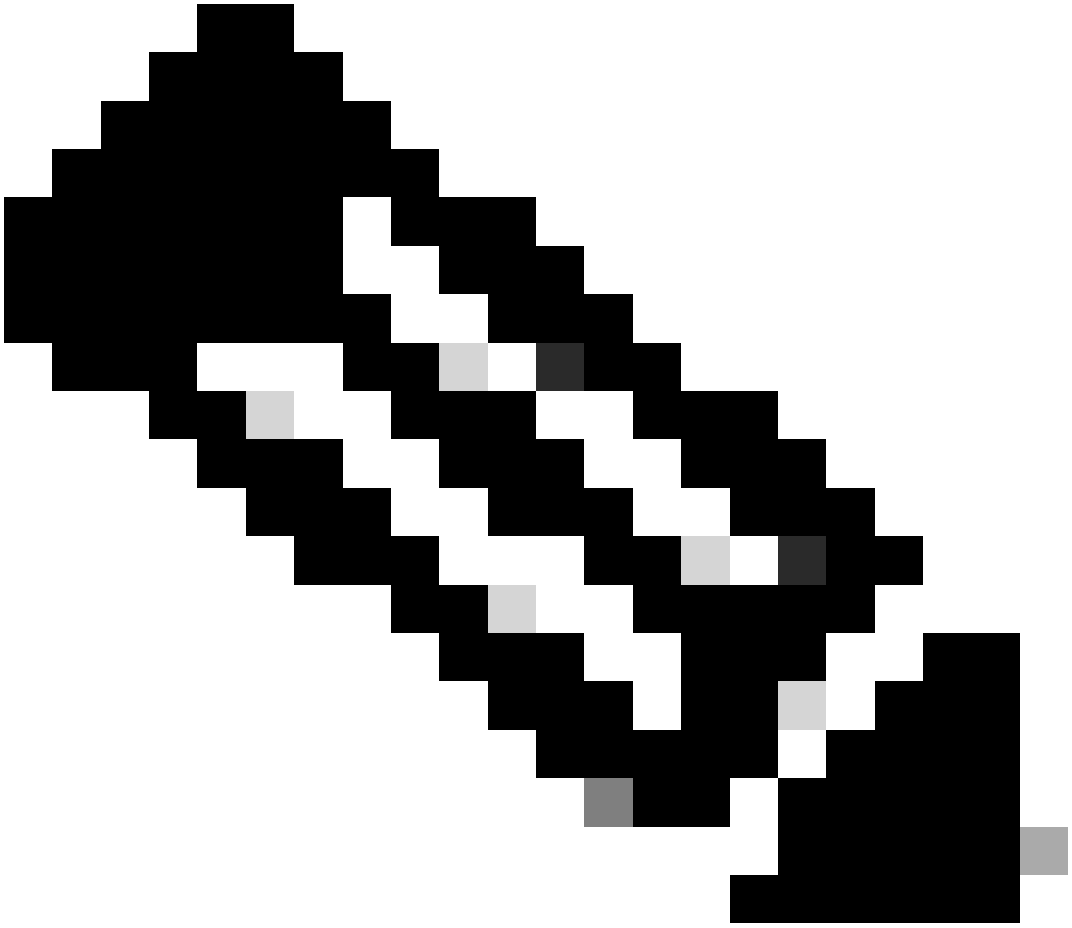
Er zijn veel verschillende opdrachten voor aggregeren (‘aggregate’). Het is belangrijk dat u begrijpt hoe alle opdrachten werken om het gewenste aggregatiegedrag te bewerkstelligen.

De eerste opdracht is die van het voorbeeld in de sectieCIDR en Gecombineerde Adressen:

<#root>

aggregate-address address-mask

Deze opdracht kondigt de route met voorvoegsel en alle meer specifieke routes aan. Het commando **aggregaat-adres 192.168.160.0** verspreidt een extra netwerk 192.168.160.0 maar voorkomt niet de verspreiding van 172.31.160.0 naar RTA. Het resultaat is dat zowel netwerk 192.168.160.0 als netwerk 172.31.160.0 worden aangekondigd bij RTA. Dus zowel het voorvoegsel als de meer specifieke route worden aangekondigd.



Opmerking: Een adres kan niet worden geaggregeerd als er geen meer specifieke route is van het betreffende adres in de BGP-routingtabel.

RTB kan bijvoorbeeld geen aggregaat voor 192.168.160.0 genereren als RTB geen specifiekere vermelding van 192.168.160.0 in de BGP-tabel heeft. Een injectie van de meer specifieke route in de BGP-tabel is mogelijk. De injectie van de route is mogelijk via:

- Inkomende updates van andere autonome systemen
-

Herdistributie van een IGP of statische route in BGP

- De opdracht network, bijvoorbeeld network 172.31.160.0

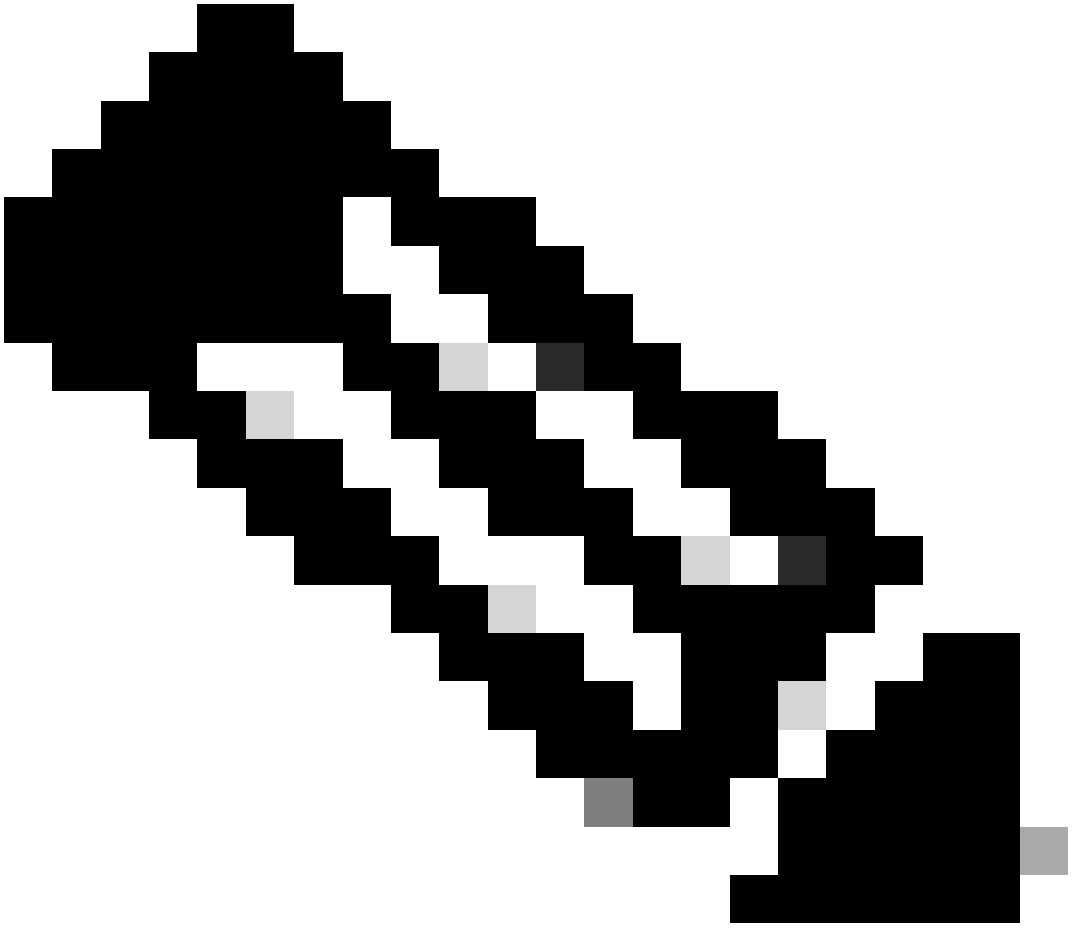
Als u wilt dat RTC alleen het netwerk 192.168.160.0 doorgeeft en niet de meer specifieke route, geeft u deze opdracht:

<#root>

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

Deze opdracht kondigt alleen het voorvoegsel aan. De opdracht onderdrukt alle meer specifieke routes.

Het commando **aggregaat 192.168.160.0 255.0.0 samenvatting-only** verspreidt netwerk 192.168.160.0 en onderdrukt de meer specifieke route 172.31.160.0.



Opmerking: Als u een netwerk aggregeert dat in uw BGP is geïnjecteerd is via de instructie network, dan injecteert de netwerkvermelding altijd in BGP-updates. Deze injectie vindt plaats, ook al gebruikt u de opdracht aggregate summary-only. Het voorbeeld in de sectie CIDR-voorbeeld 1 beschrijft deze situatie.

<#root>

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

Deze opdracht kondigt het voorvoegsel en de meer specifieke routes aan. Maar de opdracht omvat as-set-informatie in de padinformatie van de routingupdates.

<#root>

aggregate 192.168.0.0 255.0.0.0 as-set

De sectieCIDR Voorbeeld 2 (as-set) bespreekt deze opdracht.

Als u meer specifieke routes wilt onderdrukken bij de aggregatie, definieer dan een routekaart en pas de routekaart toe op de aggregaten. Met deze actie kunt u selectief zijn wat betreft de meer specifieke routes die u wilt onderdrukken.

<#root>

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

Deze opdracht kondigt het voorvoegsel en de meer specifieke routes aan. Maar de opdracht onderdrukt aankondigingen op basis van een routekaart. Veronderstel dat, met het diagram in de sectie CIDR en Geaggregeerde Adressen, u 192.168.160.0 wilt samenvoegen, de specifiekere route 192.168.160.20 wilt onderdrukken, en de propagatie van 172.31.160.0 toestaan. Gebruik deze routekaart:

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

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

Door het definiëren van de suppress-map worden de updates van pakketten die de toegangslijst toestaat onderdrukt.

Pas de routekaart vervolgens toe op de aggregate-instructie.

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

Een andere variatie is:

<#root>

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

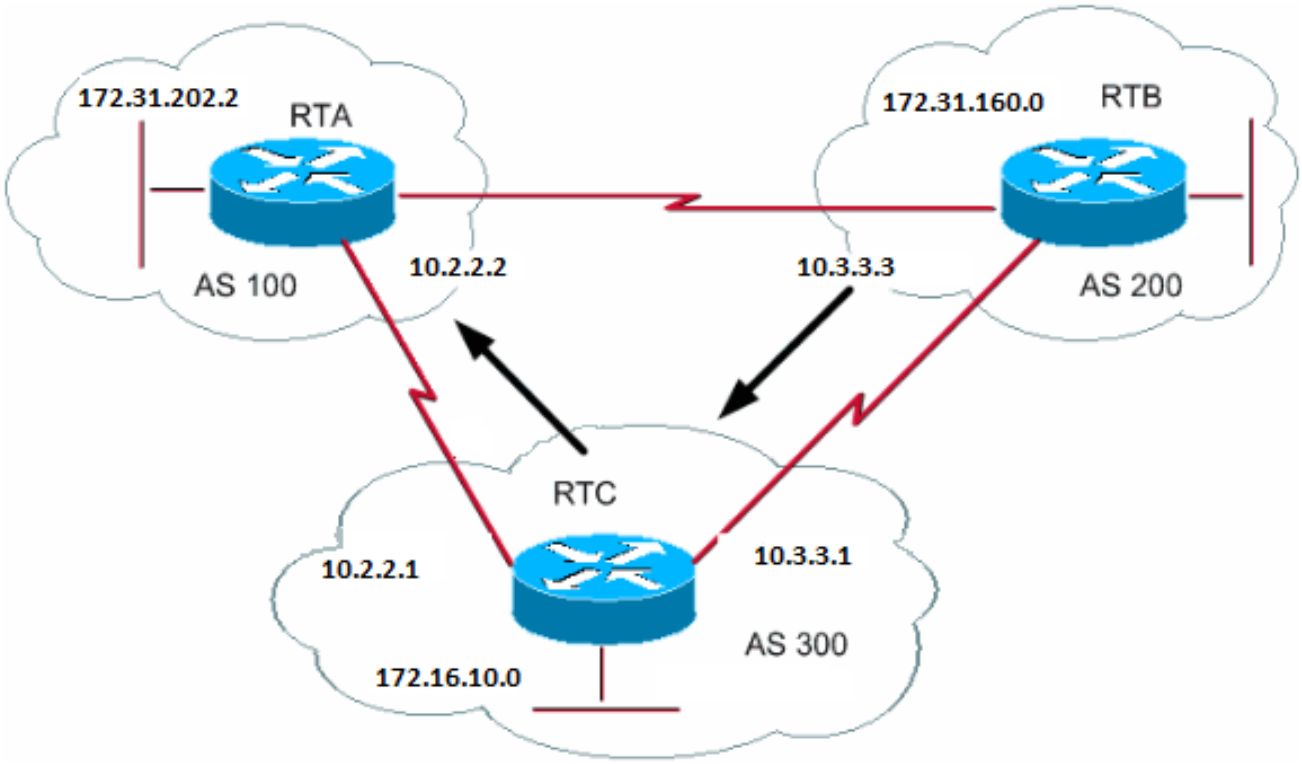
Met deze opdracht kunt u de kenmerken, zoals metric, instellen op het moment dat aggregaten worden verzonden. Om de oorsprong van de aggregaten voor IGP in te stellen, past u deze routekaart toe op de opdracht aggregate attribute-map:

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

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

Raadpleeg Routeaggregatie in BGP begrijpen voor meer informatie.

CIDR-voorbeeld 1



Verzoek: Laat RTB het prefix 192.168.160.0 te adverteren en alle meer specifieke routes te onderdrukken. Het probleem met dit verzoek is dat netwerk 172.31.160.0 lokaal is aan AS200, wat betekent dat AS200 de initiator van 172.31.160.0 is. U kunt geen RTB genereren van een prefix voor 192.168.160.0 zonder het genereren van een ingang voor 172.31.160.0, zelfs als u de **samengestelde samenvatting-slechts** opdracht gebruikt. RTB genereert beide netwerken omdat RTB de maker is van 172.31.160.0. Er zijn twee oplossingen voor dit probleem.

De eerste oplossing is een statische route te gebruiken en herdistributie naar BGP. Het resultaat hiervan is dat RTB het aggregaat aankondigt met een onvolledige oorsprong (?).

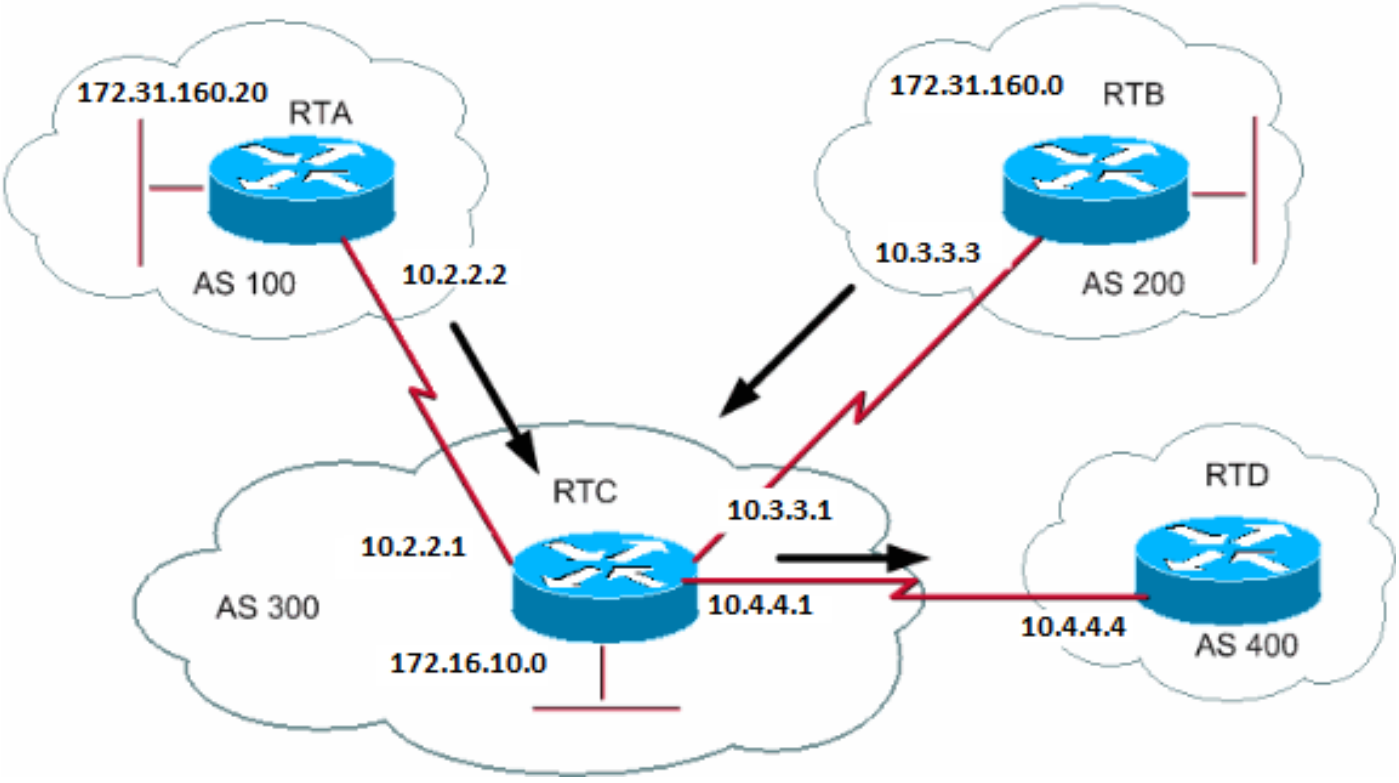
```
RTB#
router bgp 200
 neighbor 10.3.3.1 remote-as 300
 redistribute static

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

ip route 192.168.160.0 255.0.0.0 null0
```

Bij de tweede oplossing voegt u naast de statische route een vermelding toe voor de opdracht network. Deze vermelding heeft hetzelfde effect, behalve dat de vermelding de oorsprong instelt van de update voor IGP.

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

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

 neighbor 10.3.3.1 remote-as 300
 redistribute static

ip route 192.168.160.0 255.0.0.0 null0
```

CIDR-voorbeeld 2 (as-set)

U gebruikt de instructie as-set in een aggregatie om de grootte van de padinformatie te beperken. Met as-set wordt het AS-nummer maar één keer vermeld, ongeacht hoeveel keer het AS-nummer in verschillende geaggregeerde paden is verschenen. U gebruikt de opdracht aggregate as-set in situaties waarin de aggregatie van informatie leidt tot verlies van informatie met betrekking tot het kenmerk pad. In dit voorbeeld krijgt RTC updates over 192.168.160.20 van RTA en updates over 172.31.160.0 van RTB. Stel dat RTC netwerk 192.168.160.0/8 wil aggregeren en het netwerk wil verzenden naar RTD. RTD kent de oorsprong van die route niet. Als u de instructie aggregate as-set toevoegt, dwingt u RTC om padinformatie te genereren in de vorm van een set {}. Die set omvat alle padinformatie, ongeacht welk pad eerst was.




```
RTB#
router bgp 200
```

```
 network 172.31.160.0
 neighbor 10.3.3.1 remote-as 300

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

Situatie 1:

RTC heeft geen as-set-instructie. RTC stuurt een update 192.168.160.0/8 naar RTD met padinformatie (300), alsof de route afkomstig was van AS300.

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

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

Situatie 2:

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

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

De volgende twee onderwerpen, BGP Confederation en Route Reflectors, zijn voor Internet Service Providers (ISP's) die verdere controle willen over de explosie van iBGP peering in hun AS's.

BGP-confederatie

De implementatie van BGP-confederatie vermindert de iBGP-mesh binnen een AS. De truc is om een AS in meerdere autonome systemen te verdelen en de hele groep toe te wijzen aan één confederatie. Elk AS afzonderlijk heeft iBGP als onderdeel van een full mesh en heeft verbindingen met andere autonome systemen binnen de confederatie. Hoewel deze autonome systemen eBGP-peers hebben naar autonome systemen binnen de confederatie, wisselen de autonome systemen de routing uit alsof ze iBGP gebruiken. Op deze manier behoudt de confederatie informatie over next-hop, metric en local-preference. Van buitenaf gezien lijkt de confederatie één AS te zijn.

Om een BGP-confederatie te configureren voert u deze opdracht uit:

```
<#root>
```

bgp confederation identifier <autonomous-system>

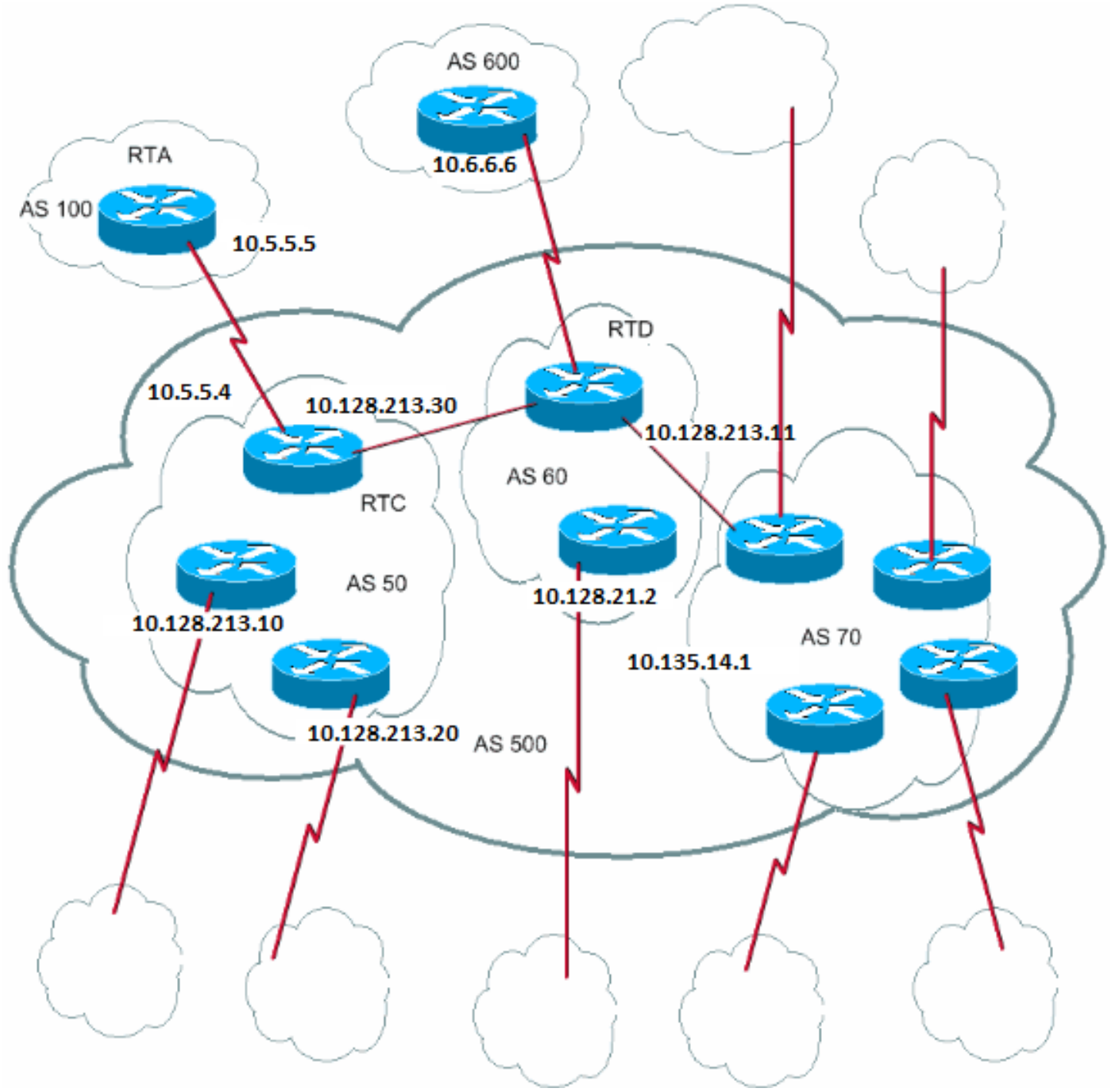
De confederatie-id is het AS-nummer van de confederatiegroep.

Door deze opdracht uit te voeren, vindt er peering plaats tussen meerdere autonome systemen binnen de confederatie:

```
<#root>
```

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

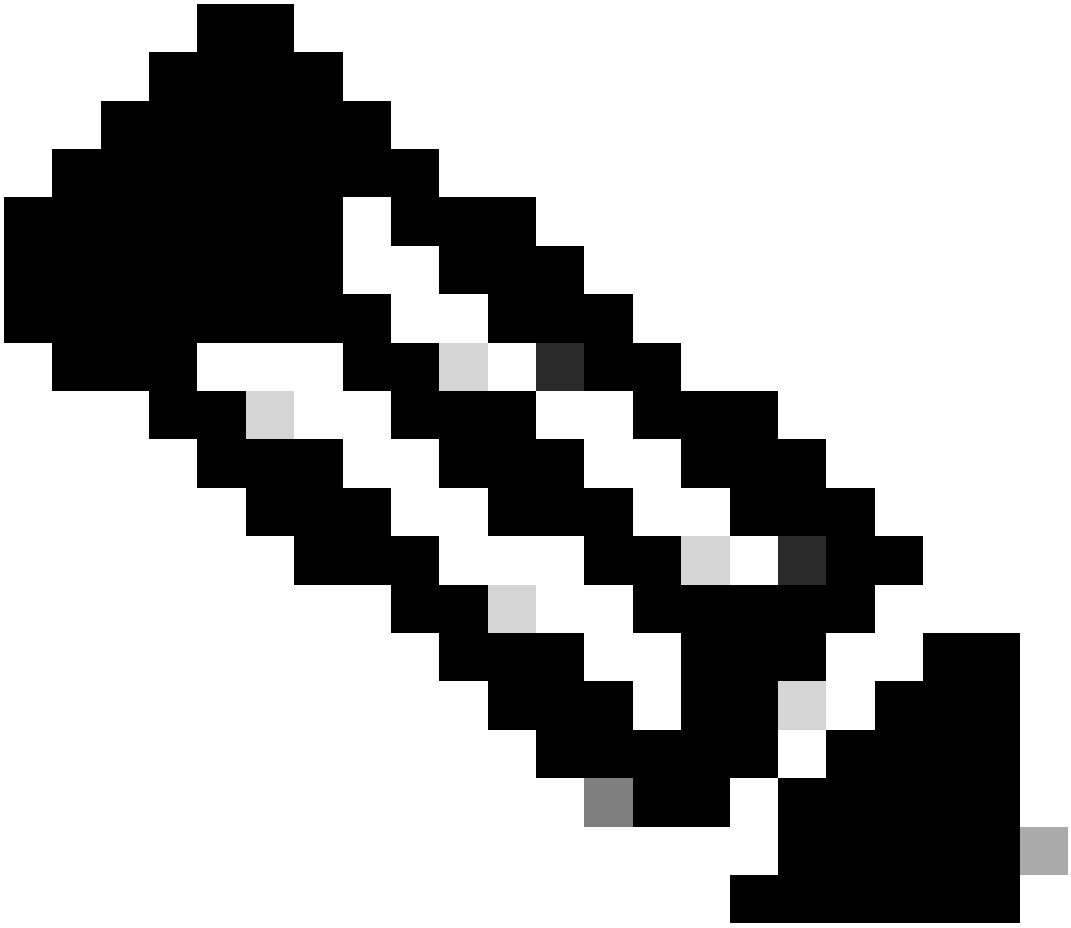
Hier ziet u een voorbeeld van confederatie:



Stel dat u een AS500 heeft die bestaat uit negen BGP-speakers. Er bestaan ook andere niet-BGP-speakers, maar u bent alleen geïnteresseerd in de BGP-speakers met eBGP-verbindingen naar andere autonome systemen. Als u een volledige iBGP-mesh wilt maken binnen AS500, dan heeft u voor elke router negen peerverbindingen nodig. U heeft acht iBGP-peers en één eBGP-peer nodig naar externe autonome systemen.

Als u confederatie gebruikt, kunt u AS500 in meerdere ASs verdelen: AS50, AS60 en AS70. U geeft het AS een confederatie-ID van 500. De buitenwereld ziet slechts één AS, AS500. Voor elk van AS50, AS60 en AS70 definieert u een volledig netwerk van iBGP-peers en definieert u de lijst van confederatiepeers met de opdracht **bgp confederation peers**.

Hier volgt een voorbeeldconfiguratie van routers RTC, RTD en RTA:



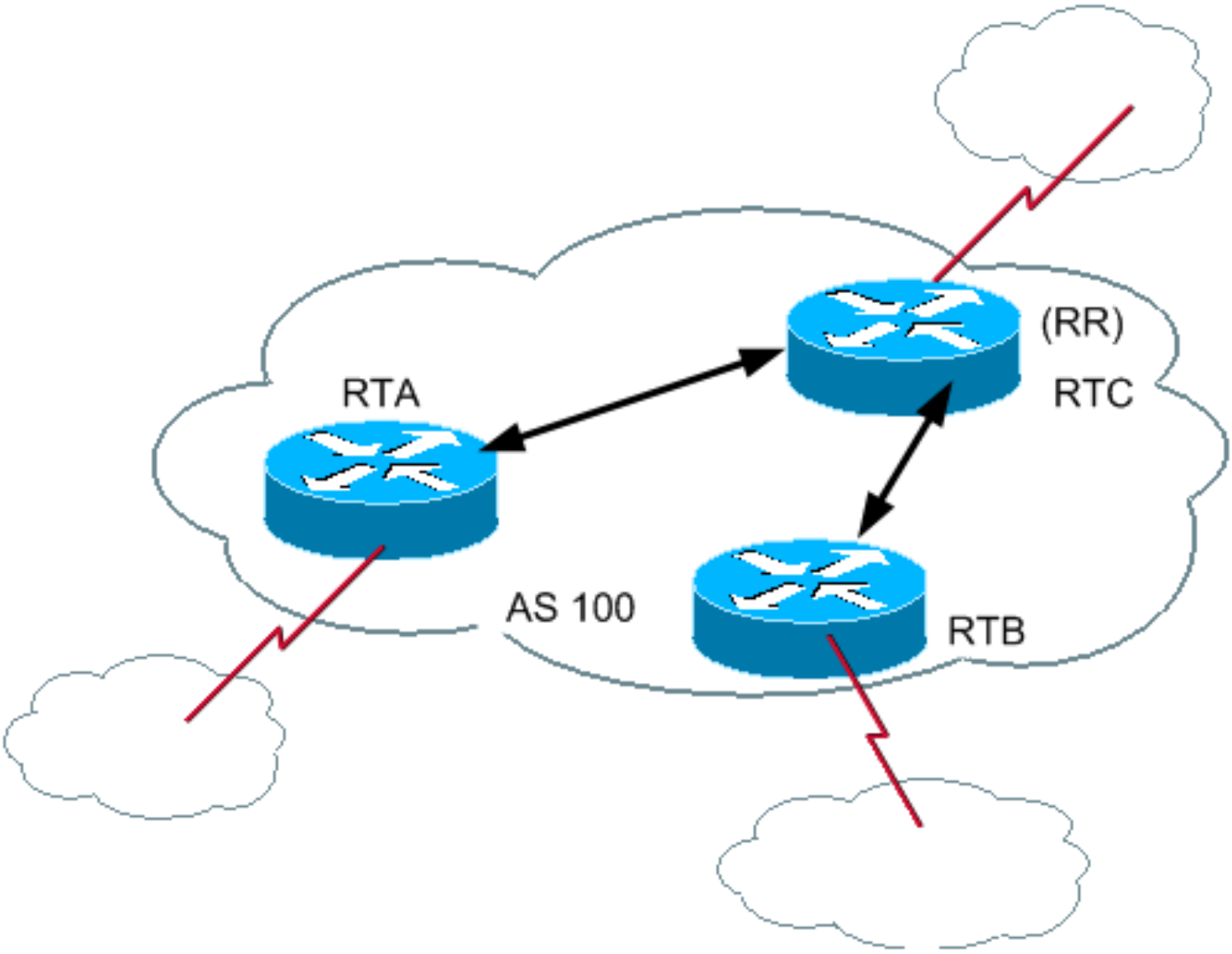
Opmerking: RTA kent geen AS50, AS60 of AS70. RTA kent alleen AS500.

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

RTD#
router bgp 60
```

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

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

Routereflectors

Een andere oplossing voor de explosie van iBGP-peering binnen een AS zijn routereflectors (RR's). Zoals het gedeelte iBGP laat zien, adverteert een BGP-luidspreker niet een route die de BGP-luidspreker via een andere iBGP-luidspreker heeft geleerd aan een derde iBGP-luidspreker. U kunt deze beperking iets versoepelen en extra controleopties inbouwen zodat een router via iBGP geleerde routes kan aankondigen of reflecteren naar andere iBGP-speakers. Deze routereflectie vermindert het aantal iBGP-peers binnen een AS.



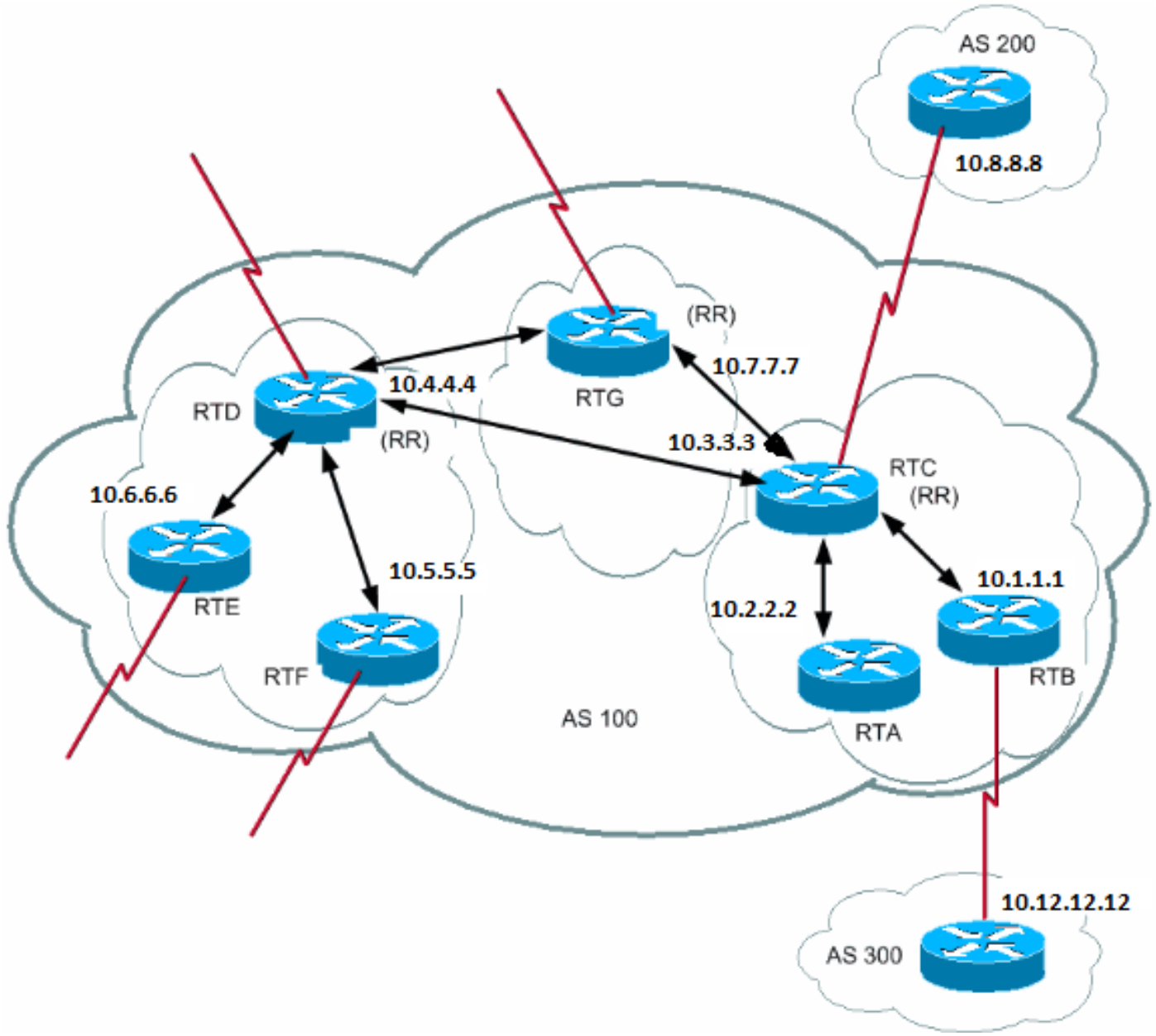
In normale gevallen moet u een volledig iBGP-mesh tussen RTA, RTB en RTC handhaven binnen AS100. Als u het RR-concept gebruikt, kan RTC worden geselecteerd als een RR. Op deze manier heeft RTC een gedeeltelijke iBGP-peering met RTA en RTB. Peering tussen RTA en RTB is niet nodig omdat RTC een RR is voor de updates die afkomstig zijn van RTA en RTB.

<#root>

neighbor <ip address> route-reflector-client

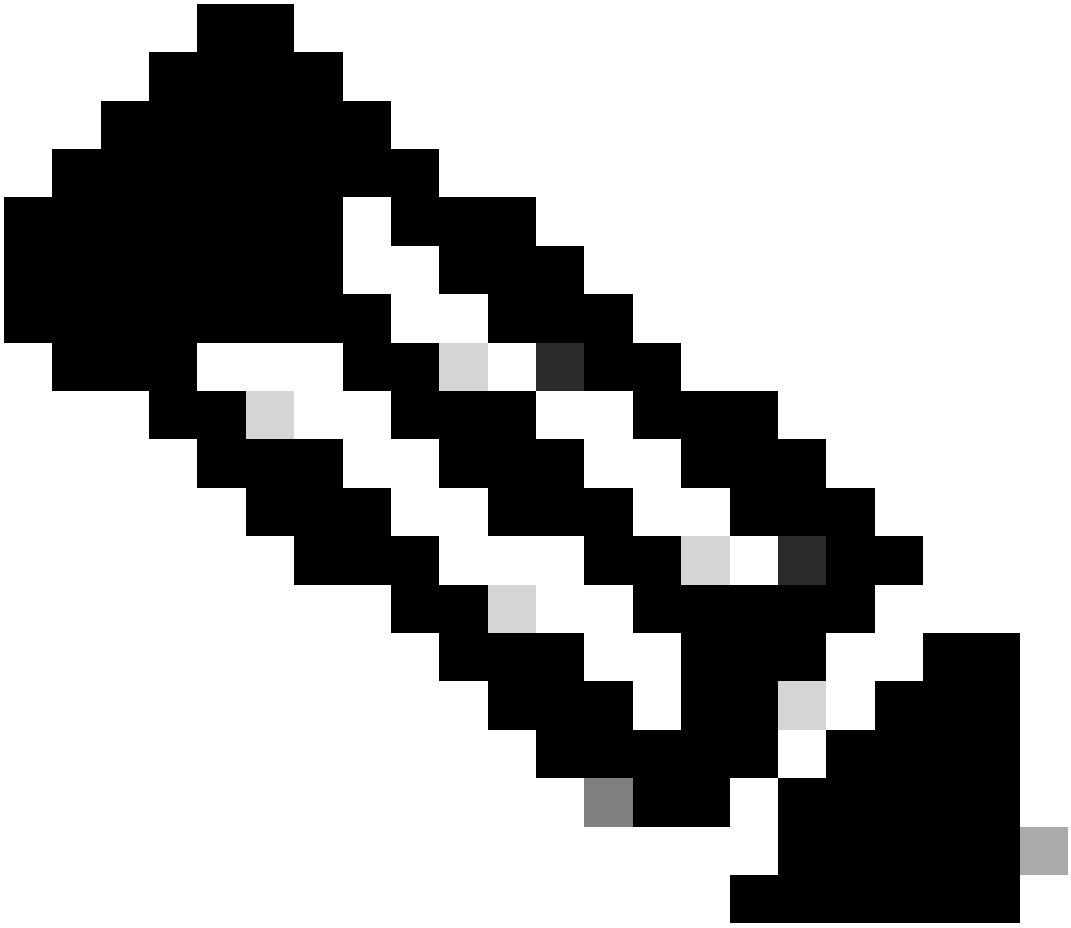
De router met deze opdracht is de RR, en de neighbors waar de opdracht naar verwijst zijn de clients die RR. In het voorbeeld gebruikt de RTC-configuratie de opdracht neighbor route-reflector-client die wijst naar de IP-adressen van RTA en RTB. De combinatie van de RR en de clients is een ‘cluster’. In dit voorbeeld vormen RTA, RTB en RTC een cluster met één RR binnen AS100.

Andere iBGP-peers van de RR die geen clients zijn, zijn niet-clients.



Een AS kan meer dan één RR hebben. In deze situatie behandelt een RR andere RR's net als andere iBGP-speakers. Andere RR's kunnen tot hetzelfde cluster (clientgroep) behoren of tot andere clusters. Bij een eenvoudige configuratie kunt u het AS in meerdere clusters verdelen. U configureert elke RR met andere RR's als nonclientpeers in een full mesh-topologie. Clients mogen niet peer met iBGP-luidsprekers buiten het clientcluster.

In het vorige diagram vormen RTA, RTB en RTC één cluster. RTC is de RR. Voor RTC zijn RTA en RTB clients en al het andere is een nonclient. Onthoud dat de opdracht neighbor route-reflector-client wijst naar clients van een RR. Dezelfde RTD is de RR voor clients RTE en RTF. RTG is een RR in een derde cluster.



Opmerking: RTD, RTC en RTG zijn volledig vermaasd, maar routers binnen een cluster zijn dat niet.

Wanneer een RR een route ontvangt, routeert de RR deze zoals in deze lijst is aangegeven. Deze activiteit is echter afhankelijk van het type peer:

- Routes van een nonclientpeer: reflecteren naar alle clients binnen het cluster.
- Routes van een clientpeer: reflecteren naar alle nonclientpeers en ook naar de clientpeers.

- Routes van een eBGP-peer: versturen de update naar alle client- en nonclientpeers.

Dit is de relatieve BGP-configuratie van routers RTC, RTD en RTB:

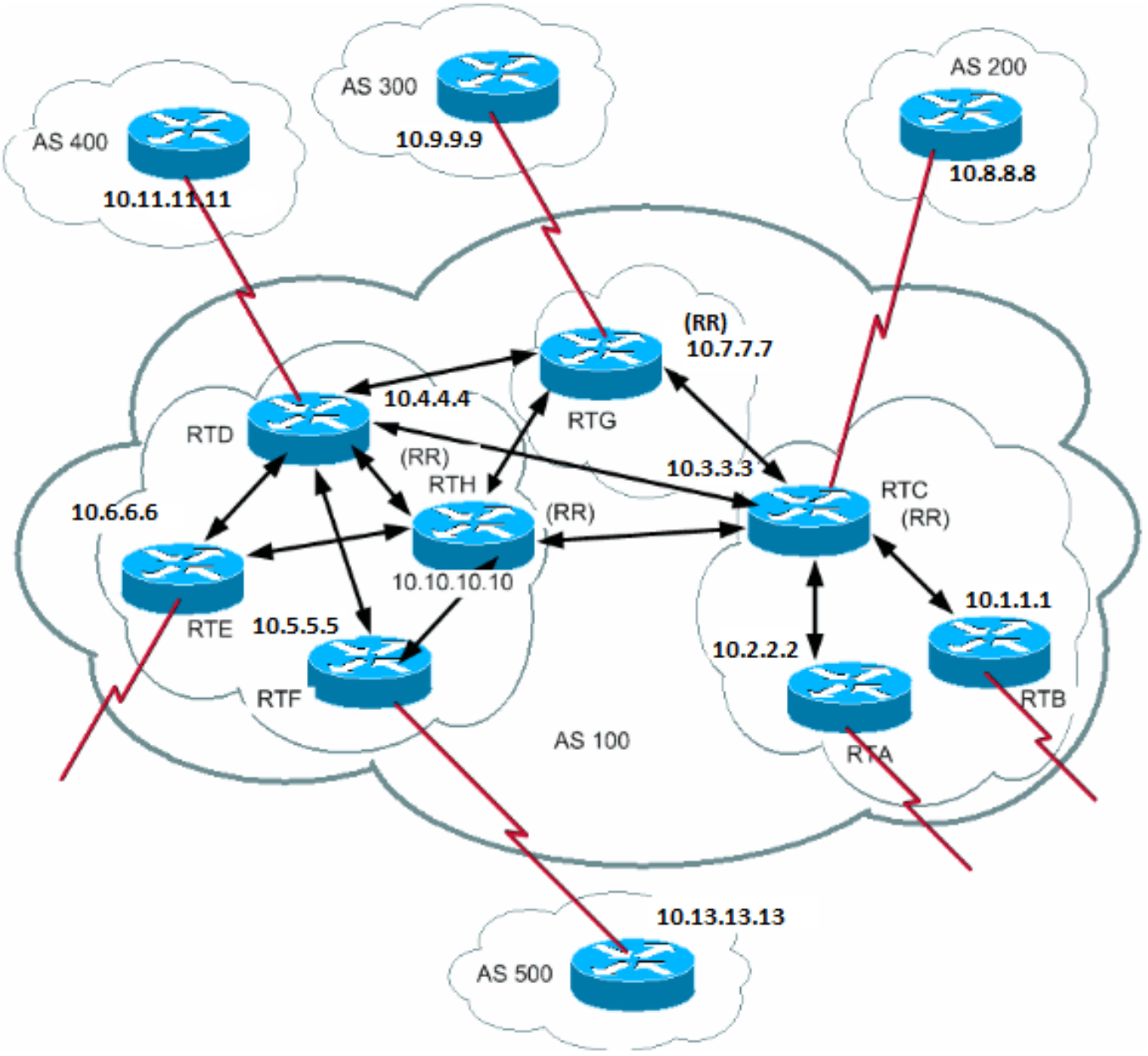
```
RTC#
router bgp 100
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 route-reflector-client
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.7.7.7 remote-as 100
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.8.8.8 remote-as 200

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

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

Omdat er een reflectie is van de via iBGP geleerde routes, kan er een lus zijn met routinginformatie. Het RR-systeem kan deze lus op een aantal manieren vermijden:

- **originator-id** — Dit is een optioneel, niet-transitief BGP attribuut dat 4 bytes lang is. Een RR maakt dit kenmerk. Het kenmerk bevat de router-ID (RID) van de oorsprong van de route in het lokale AS. Als, door een slechte configuratie, de routinginformatie terugkomt bij de oorsprong, dan wordt de informatie genegeerd.

- **clusterlijst** — de sectieMeervoudige RR's binnen een Cluster omvat clusterlijst.

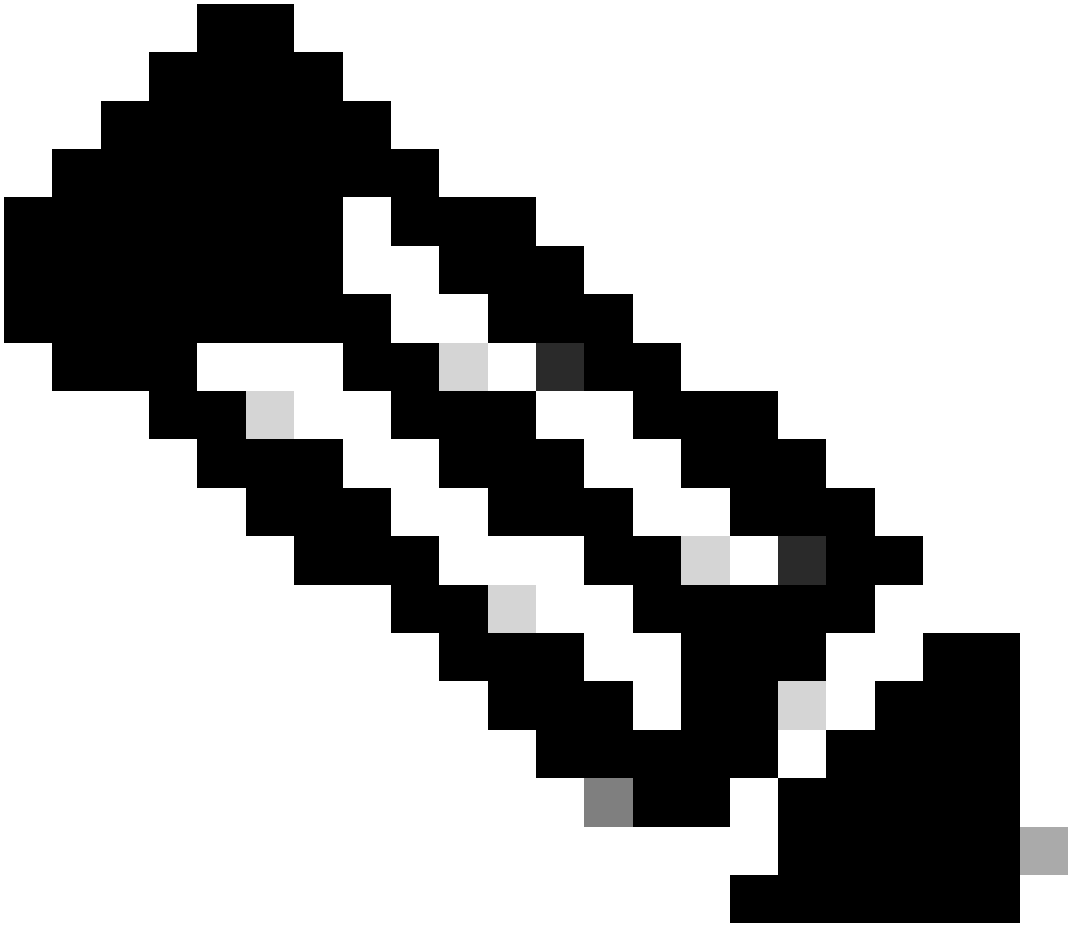
Meerdere RR’s binnen een cluster



Een cluster van clients heeft doorgaans één RR. In dit geval identificeert de router-ID van de RR het cluster. Om de redundantie te verhogen en storingspunten te vermijden, kan een cluster meer dan één RR hebben. U moet alle RR’s in hetzelfde cluster configureren met een cluster-ID van 4 bytes, zodat een RR updates van RR’s in hetzelfde cluster kan herkennen.

Een clusterlijst is een reeks cluster-ID’s die de route heeft doorlopen. Wanneer een RR een route van de RR-clients reflecteert naar nonclients buiten het cluster, dan voegt de RR de lokale cluster-ID toe aan de clusterlijst. Als deze update een lege clusterlijst heeft, creëert de RR er een. Met dit kenmerk kan een RR identificeren of de routinginformatie door een slechte configuratie via een lus weer terug is gekomen bij hetzelfde cluster. Als de lokale cluster-ID in de clusterlijst staat, wordt de aankondiging genegeerd.

In het diagram in deze sectie behoren RTD, RTE, RTF en RTH tot één cluster. Zowel RTD als RTH zijn RR’s voor hetzelfde cluster.

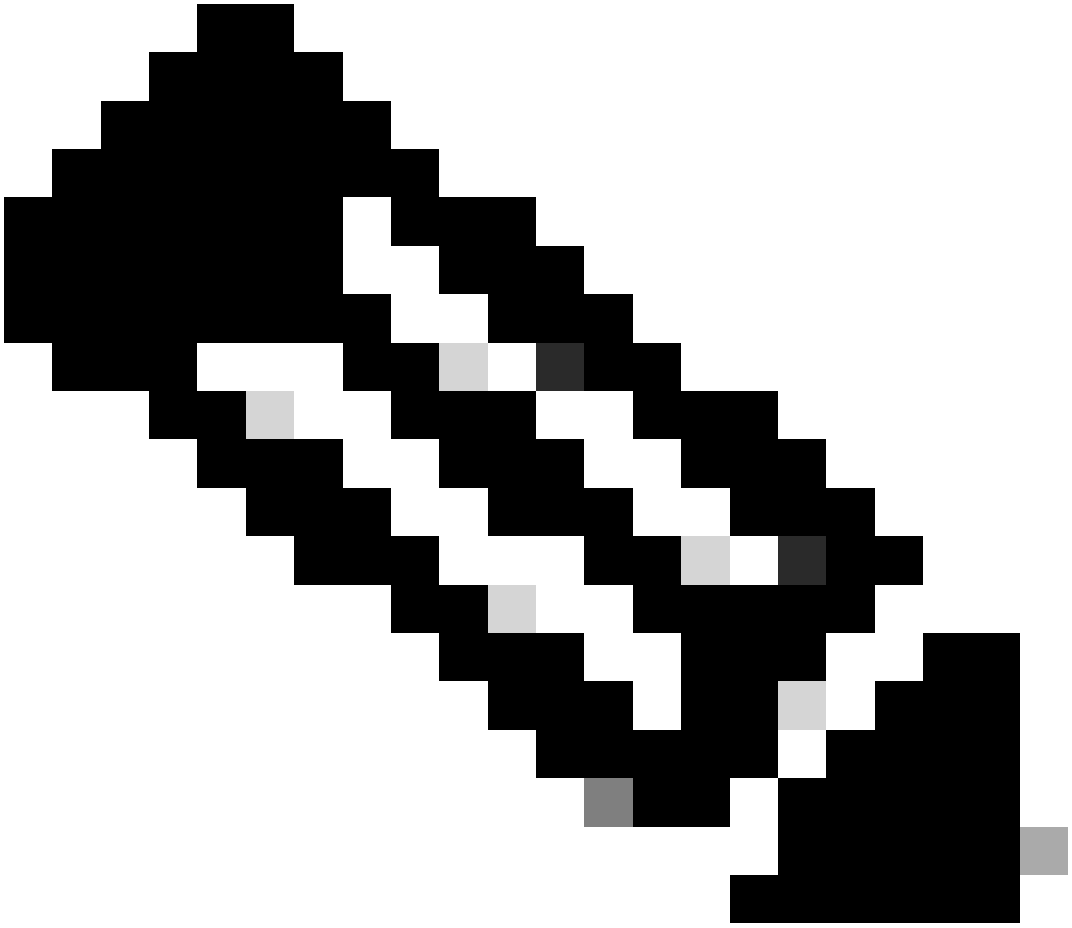


Opmerking: Er is redundantie omdat RTH full mesh peering heeft met alle RR's. Als RTD wegvalt, neemt RTH de plaats in van RTD.

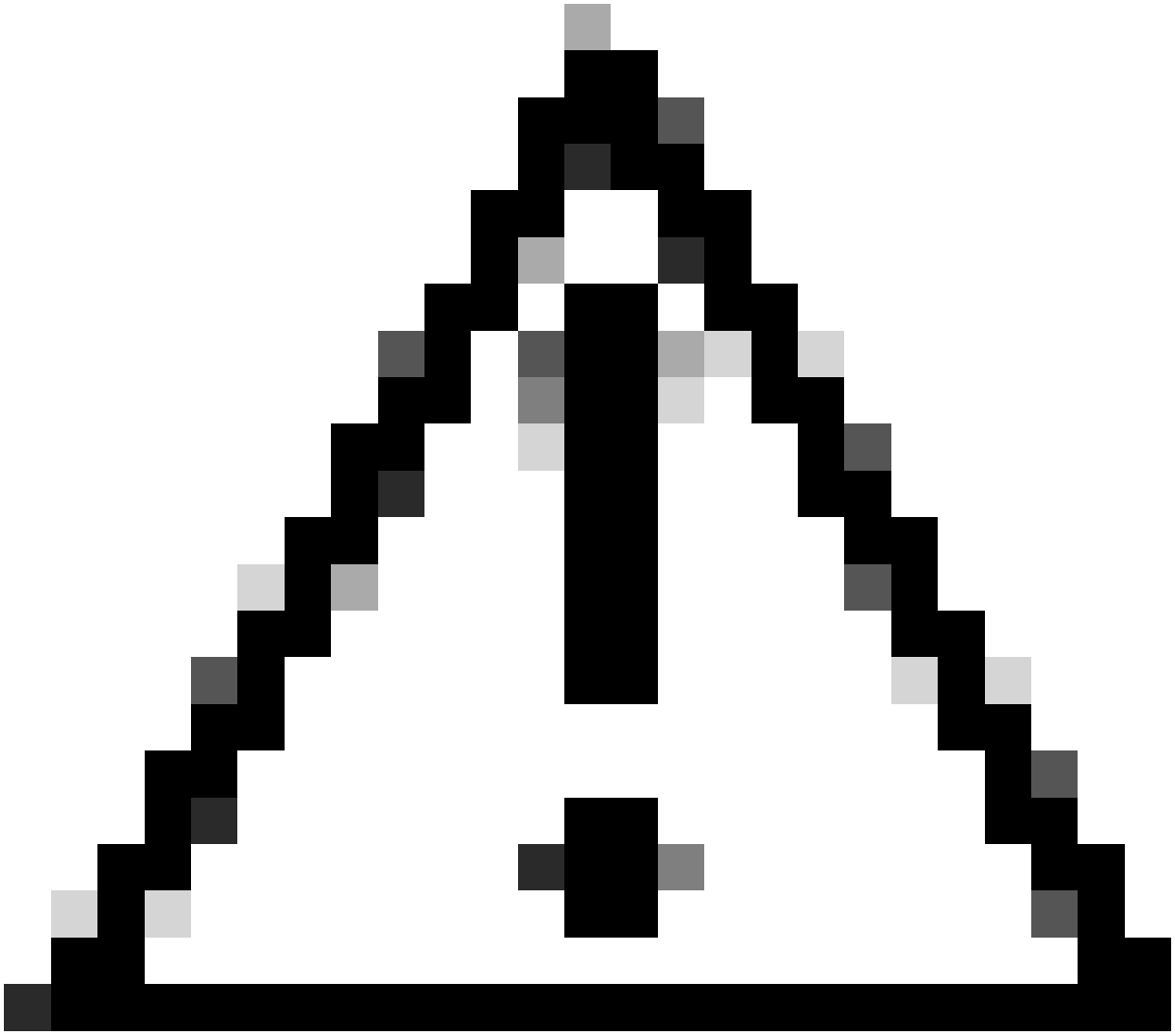
Dit is de configuratie van RTH, RTD, RTF en RTC:

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

```
 neighbor 10.3.3.3 remote-as 100
 neighbor 10.9.9.9 remote-as 300
 bgp cluster-id 10

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

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

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



Opmerking: U heeft de opdracht bgp cluster-id niet nodig voor RTC omdat er maar één RR in dat cluster bestaat.



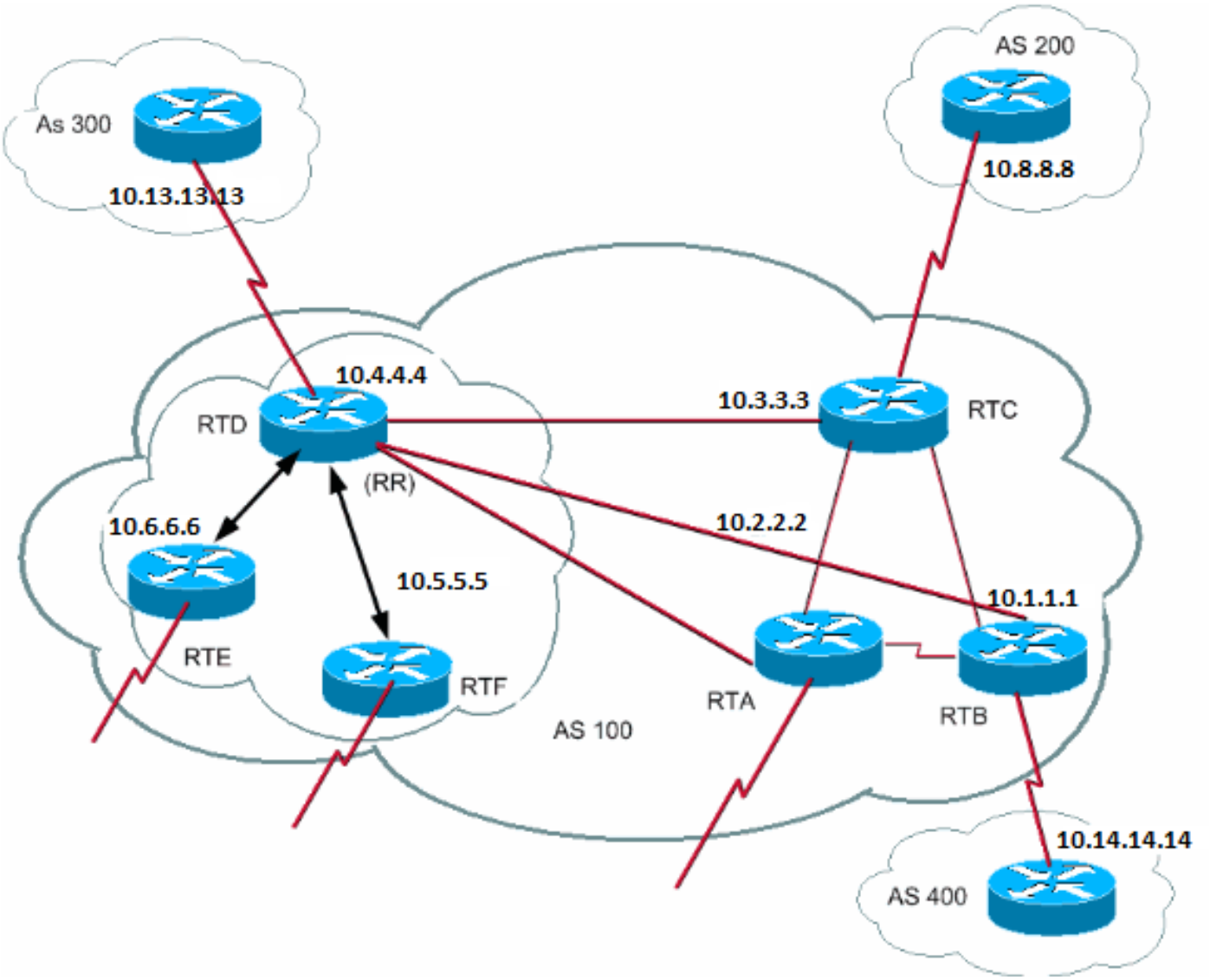
Waarschuwing: deze configuratie gebruikt geen peer-groepen. Gebruik geen peergroepen wanneer de clients binnen een cluster onderling geen directe iBGP-peers hebben en de clients updates uitwisselen via de RR. Als u peergroepen configureert, dan wordt een mogelijke terugtrekking naar de bron van een route op de RR doorgegeven aan alle clients binnen het cluster. Deze transmissie kan problemen veroorzaken.

De subopdracht voor de router **bgp client-to-client reflection** **is standaard ingeschakeld op de RR.** Als u client-to-client BGP-reflectie op de RR uitschakelt en zorgt voor redundante BGP-peering tussen de clients, dan kunt u veilig gebruikmaken van peergroepen. Raadpleeg Beperkingen van peergroepen voor meer informatie.

RR- en conventionele BGP-speakers

Een AS kan BGP-speakers hebben die het concept van RR's niet begrijpen. In dit document worden deze routers conventionele BGP-speakers

genoemd. In het RR-systeem zijn dergelijke conventionele BGP-speakers toegestaan. Deze routers kunnen lid zijn van een clientgroep of van een nonclientgroep. Deze routers maken een eenvoudige en geleidelijke migratie van het huidige iBGP-model naar het RR-model mogelijk. U kunt beginnen met het maken van clusters wanneer u één router als RR configureert en andere RR's en RR-clients instelt als normale iBGP-peers. Vervolgens kunt u geleidelijk meer clusters maken.



In dit diagram gebruiken RTD, RTE en RTF het concept van routereflectie. RTC, RTA en RTB zijn conventionele routers. U kunt deze routers niet als RR's configureren. Er kan een normale iBGP-mesh worden gebruikt tussen deze routers en RTD. Wanneer u later klaar bent om te upgraden, kunt u van RTC een RR maken met clients RTA en RTB. Clients hoeven de routereflectieregeling niet te begrijpen; alleen de RR's vereisen de upgrade.

Dit is de configuratie van RTD en RTC:

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

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

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

Wanneer u klaar bent om RTC te upgraden en van RTC een RR te maken, verwijdert u de iBGP full mesh maakt u RTA en RTB clients van RTC.

Lus van routinginformatie voorkomen

Tot nu toe zijn in dit document twee eigenschappen genoemd die u kunt gebruiken om potentiële informatielussen te voorkomen: **originator-id** en **clusterlijst**.

Een andere manier om lussen te controleren is meer beperkingen in te stellen voor de set-**clausule van uitgaande routekaarten.** De set-clausule voor uitgaande routekaarten heeft geen invloed op routes die reflecteren naar iBGP-peers.

Je kunt ook meer beperkingen stellen aan **next-hop-self**, wat een per-buurconfiguratie optie is. Wanneer u **next-hop**-selfo in RRs gebruikt, heeft de clausule alleen invloed op de volgende hop van eBGP geleerde routes omdat de volgende hop van gereflecteerde routes niet moet worden gewijzigd.

Beperking van routefluctuatie

Cisco IOS-softwarerelease 11.0 heeft routebeperking geïntroduceerd. Routebeperking is een mechanisme om de instabiliteit te minimaliseren die wordt veroorzaakt door routefluctuatie. Routebeperking vermindert ook de oscillatie via het netwerk. U definieert criteria om slecht presterende routes te identificeren. Een fluctuerende route krijgt een krijgt 1000 ‘strafpunten’ voor elke fluctuatie. Zodra de cumulatieve boete een vooraf bepaalde onderdrukkingsgrens bereikt, komt onderdrukking van de routereclame voor. De straf vervalt exponentieel op basis van een vooraf ingestelde halfwaardetijd. Zodra de boete afneemt onder een vooraf bepaalde hergebruikslimiet, wordt de routeradvertentie niet langer onderdrukt.

Routebeperking is niet van toepassing op routes die extern zijn voor een AS en die geleerd zijn via iBGP. Op deze manier vermijdt routebeperking een hogere straf voor de iBGP-peers voor routes buiten het AS.

De straf vervalt met intervallen van 5 seconden. De routes zijn niet onderdrukt bij een granulariteit van 10 seconden. De router houdt de bevochtigende informatie tot de sanctie minder dan de helft van de hergebruikslimiet wordt. Op dat moment schoont de router de informatie op.

Aanvankelijk is routebeperking standaard uitgeschakeld. Als er een behoefte is, kan deze eigenschap standaard enablement in de toekomst worden gegeven. Deze opdrachten controleren de routebeperking:

- **bgp bevochtiging** — schakelt bevochtiging in.

- **geen bgp bevochtiging** — schakelt bevochtiging uit.

- **bgp dempende halfwaardetijd**— verandert de halfwaardetijd.

Een opdracht die alle parameters tegelijkertijd instelt, is:

- **bgp demping halfwaardetijd-tijd-tijdgebruiksuppressmaximum-suppressietijd**

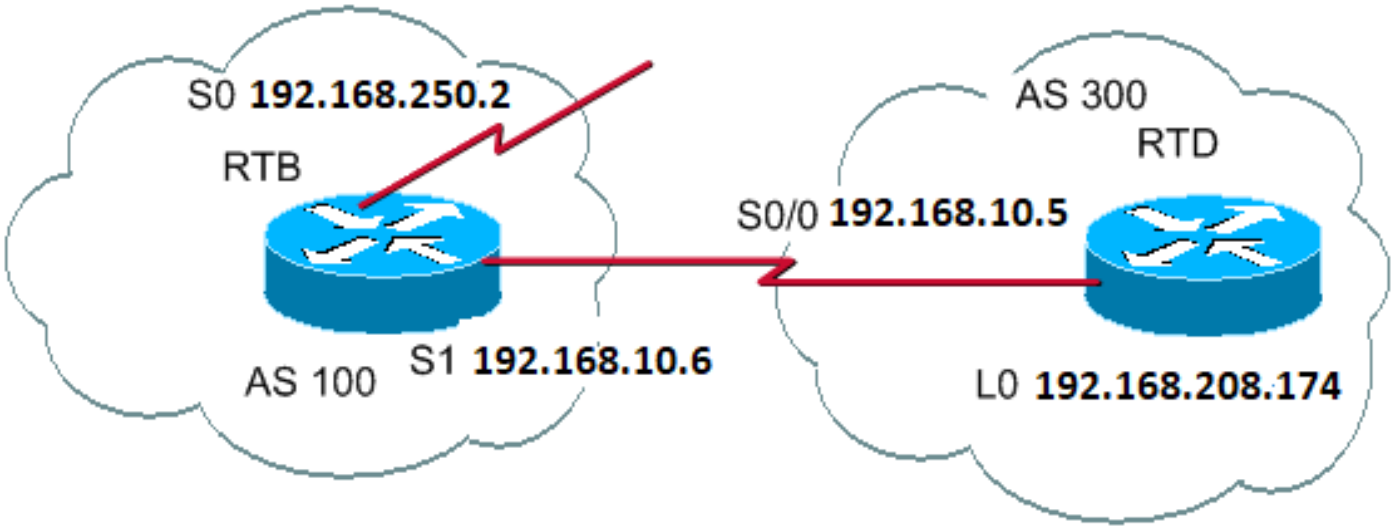
In deze lijst wordt de syntaxis beschreven:

- **halfwaardetijd**— De spreiding is 1-45 minuten, en het huidige gebrek is 15 minuten.

- **hergebruik-waarde** - Het bereik is 1-20.000 en de standaardwaarde is 750.

- **onderdruk-waarde** - De waaier is 1-20.000, en het gebrek is 2000.

- **max-suppressietijd** - Dit is de maximale duur voor het onderdrukken van een route. Het bereik is 1-255 minuten en de standaard is vier keer de halveringstijd.




```
RTB#
hostname RTB

interface Serial0
 ip address 192.168.250.2 255.255.255.252

interface Serial1
 ip address 192.168.10.6 255.255.255.252

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

```
RTD#
hostname RTD

interface Loopback0
 ip address 192.168.208.174 255.255.255.192

interface Serial0/0
 ip address 192.168.10.5 255.255.255.252

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

De configuratie van RTB is bedoeld voor routebeperking met standaardparameters. Wanneer we aannemen dat de eBGP-link naar RTD stabiel is, ziet de RTB BGP-tabel er als volgt uit:

<#root>

```
RTB#
```

show ip bgp

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

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

Om een routefluctuatie te simuleren moet de opdracht clear ip bgp 192.168.10.6 worden uitgevoerd op RTD. De RTB BGP-tabel ziet er als volgt uit:


```
RTB#
```

show ip bgp

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

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

De BGP vermelding voor 192.168.10.0 is in historystate. Deze plaatsing betekent dat u geen beste pad naar de route heeft, maar dat er nog informatie is over de routefluctuatie.

```
RTB#
```

show ip bgp 192.168.10.0

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

De route heeft een boete gekregen voor flappen, maar de straf is nog steeds onder de onderdrukkingslimiet. De standaardwaarde is 2000. Er heeft nog geen routeonderdrukking plaatsgevonden. Wanneer de route nog een aantal keer fluctueert, dan ziet u:

<#root>

```
RTB#
```

show ip bgp

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

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

RTB#
```

show ip bgp 192.168.10.0

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

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

De route is bevochtigd of onderdrukt. De route wordt opnieuw gebruikt wanneer de straf de ‘waarde voor hergebruik’ heeft bereikt. In dit geval is de waarde voor hergebruik de standaard, 750. De bevochtigingsinformatie wordt gewist wanneer de boete minder dan de helft van de hergebruikslimiet bedraagt. In dit geval vindt het wissen plaats wanneer de straf 375 wordt (750/2=375). Met deze opdrachten kunt u statistische informatie over fluctuatie weergeven en wissen.

- **toon ip bgp flap-statistics** — Hier worden flap statistieken voor alle paden weergegeven.

- **toon ip bgp flap-statistics regexpregular-expressie**— Hier worden flap statistieken weergegeven voor alle paden die overeenkomen met de reguliere expressie.

- **ip bgp flap-statistics filter-listlist tonen** - Hier worden flap statistieken weergegeven voor alle paden die door het filter gaan.

- **toon ip bgp flap-statisticsA.B.C.D m.m.m.m**— Toont flap statistieken voor één enkele ingang.

- **toon ip bgp flap-statisticsA.B.C.D m.m.m.mlater-prefix** — Hier worden flap statistieken voor specifiekere items weergegeven.

- **show ip bgp neighbor [dampened-routes] | [flap-statistics]**— Hier worden flap-statistieken weergegeven voor alle paden vanuit een buur.

- **clear ip bgp flap-statistics**— Clears flap statistieken voor alle routes.

- **duidelijke ip bgp flap-statistics regexpregular-expressie**— Ontruimt flap statistieken voor alle paden die overeenkomen met de reguliere expressie.

- **wis ip bgp flap-statistics filter-listlist -** ontruimt flap statistieken voor alle paden die de filter passeren.

- **clear ip bgp flap-statisticsA.B.C.D m.m.m.m**— maakt flap statistieken op voor één ingang.

- **duidelijke ip bgpA.B.C.Dflap-statistics** — ontruimt flap statistieken voor alle paden van een buur.

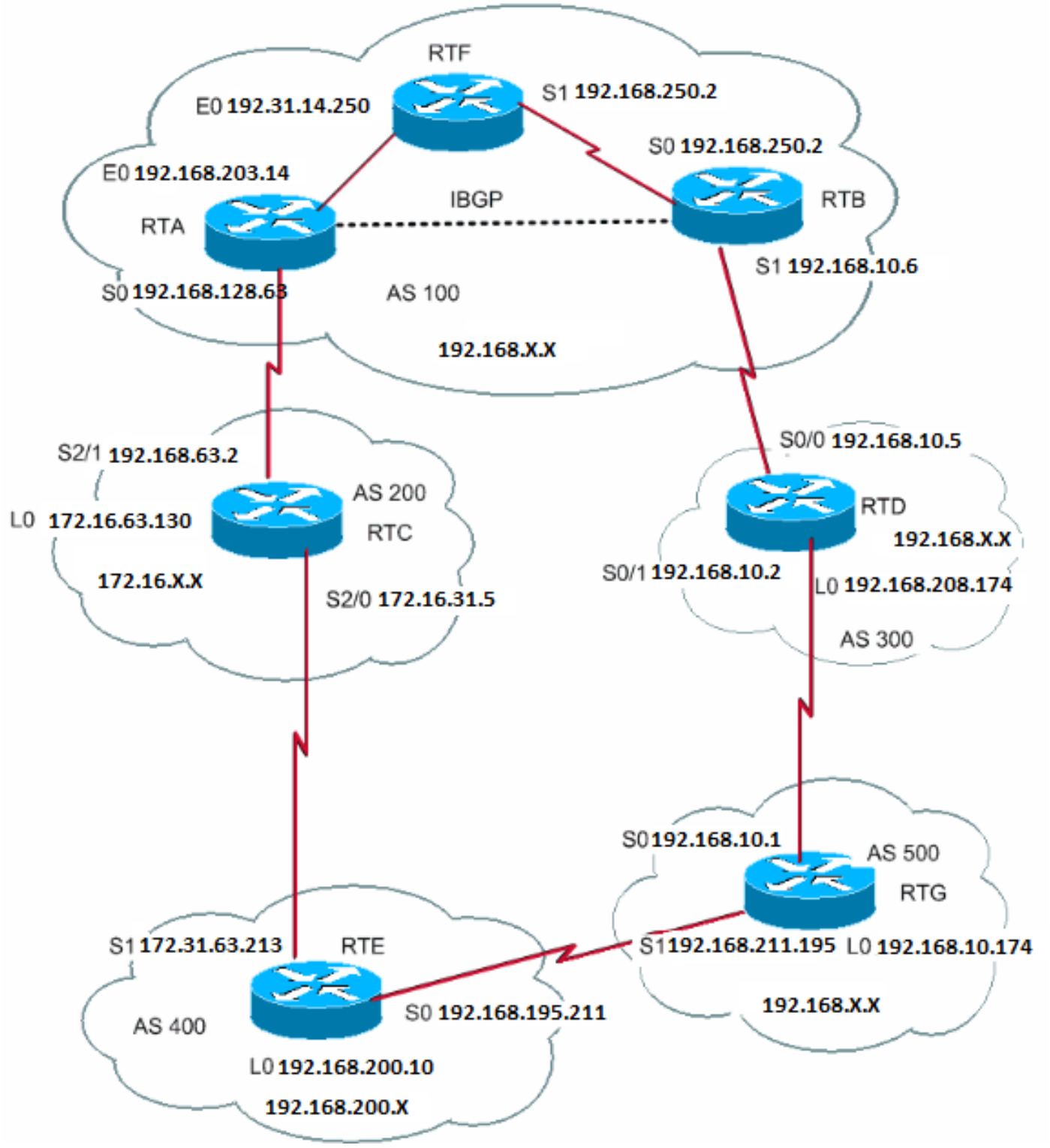
Hoe BGP een pad selecteert

Nu u bekend bent met de BGP-kenmerken en terminologie, gaat u verder met BGP-algoritme voor selectie van het beste pad.

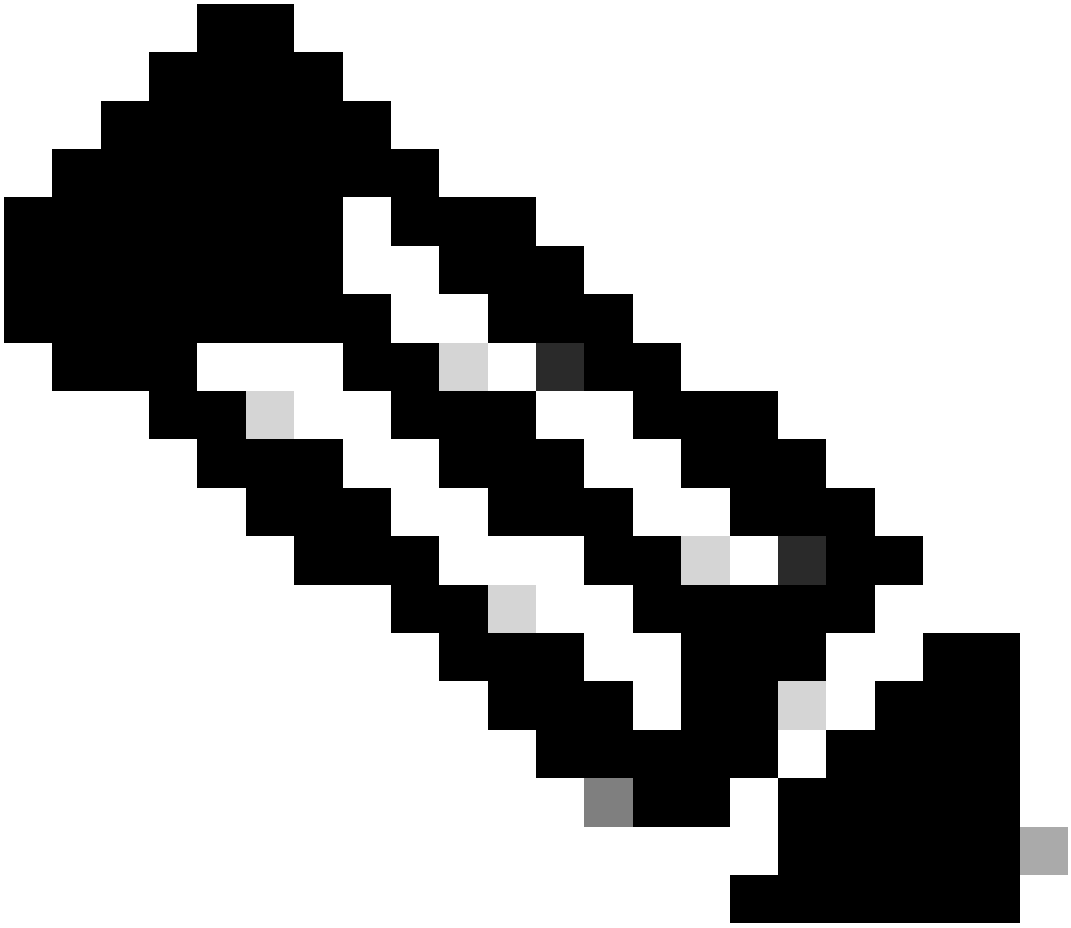
BGP-casestudy's 5

Praktisch ontwerpvoorbeeld

Deze sectie bevat een ontwerpvoorbeeld van de configuratie en routingtabellen zoals de tabellen er daadwerkelijk uitzien op Cisco-routers.



In deze sectie leert u deze configuratie stap voor stap op te bouwen en wat er fout kan gaan tijdens het proces. Als u een AS heeft dat via eBGP verbinding maakt met twee ISP's, voer iBGP dan altijd binnen uw AS uit om uw routes beter te kunnen beheren. In dit voorbeeld wordt iBGP binnen AS100 uitgevoerd tussen RTA en RTB, en OSPF wordt uitgevoerd als IGP. Veronderstel dat u met twee ISPs, AS200 en AS300 verbindt. Dit is de eerste run van de configuraties voor alle routers:



Opmerking: Deze configuraties zijn niet de definitieve configuraties.

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 192.168.203.250 255.255.255.0

interface Ethernet0
ip address 192.168.203.14 255.255.255.0

interface Serial0
```

```
 ip address 192.168.128.63 255.255.255.252

router ospf 10
 network 192.168.203.25 0.0.255.255 area 0

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

RTF#
hostname RTF

ip subnet-zero

interface Ethernet0
 ip address 172.31.14.250 255.255.255.0

interface Serial1
 ip address 172.16.15.250 255.255.255.252

router ospf 10
 network 192.168.203.25 0.0.255.255 area 0

RTB#
hostname RTB

ip subnet-zero

interface Serial0
 ip address 192.168.250.2 255.255.255.252

interface Serial1
 ip address 192.168.10.6 255.255.255.252

router ospf 10
 network 192.168.203.25 0.0.255.255 area 0

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

RTC#
hostname RTC

ip subnet-zero

interface Loopback0
 ip address 192.168.128.6330 255.255.255.192

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

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

```
 neighbor 172.31.63.213 remote-as 400

RTD#
hostname RTD

ip subnet-zero

interface Loopback0
ip address 192.168.208.174 255.255.255.192

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

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

RTE#
hostname RTE

ip subnet-zero

interface Loopback0
ip address 192.168.200.10 255.255.255.0

interface Serial0
 ip address 192.168.195.211 255.255.255.252

interface Serial1
 ip address 172.31.63.213 255.255.255.252
 clockrate 1000000

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

RTG#
hostname RTG

ip subnet-zero

interface Loopback0
 ip address 192.168.211.19574 255.255.255.192

interface Serial0
 ip address 192.168.10.1 255.255.255.252

interface Serial1
 ip address 192.168.211.195 255.255.255.252

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

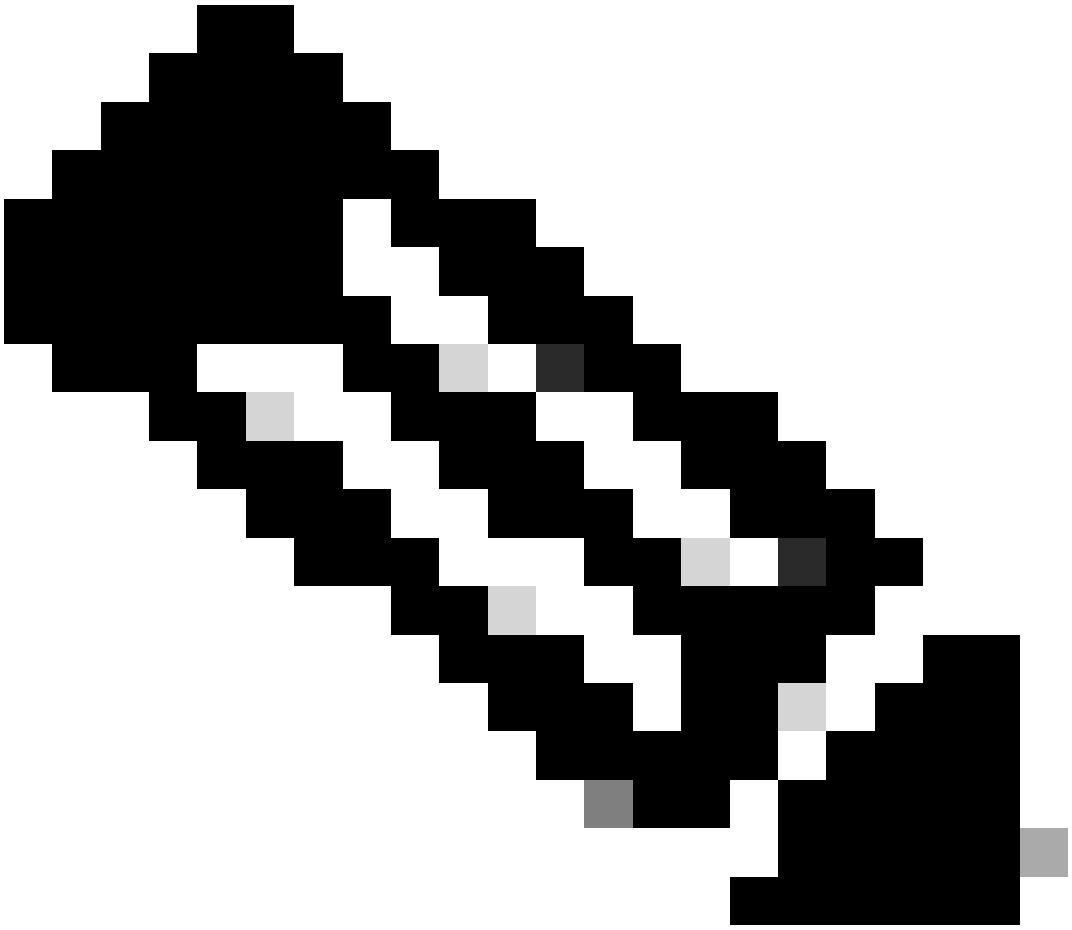
Gebruik altijd de network opdracht of verdeel statische vermeldingen opnieuw in BGP om netwerken te adverteren. Deze methode is beter dan een herdistributie van IGP in BGP. In dit voorbeeld wordt de network opdracht gebruikt om netwerken in BGP te injecteren.

Hier start u met de interface s1 bij de shutdown van RTB, alsof de link tussen RTB en RTD niet bestaat. Dit is de RTB BGP-tabel:

```
    <#root>

RTB#

show ip bgp BGP

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

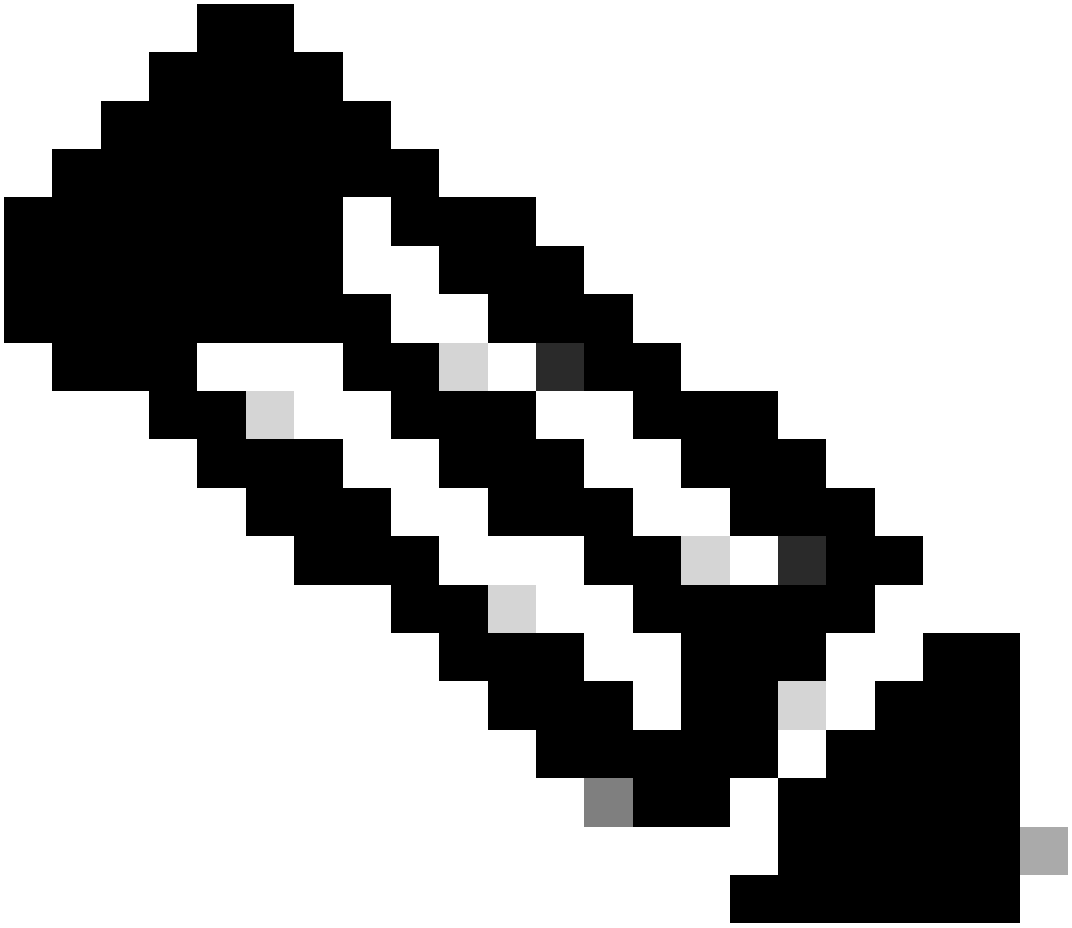
In deze tabel worden de volgende notaties gebruikt:

- Aniat the start—Geeft aan dat de invoer via een iBGP-peer is geleerd.
- Aniat the end—Geeft aan dat de oorsprong van de padinformatie IGP is.
- Pathinformatie — Deze informatie is intuïtief. Netwerk 172.31.10.0 wordt bijvoorbeeld geleerd via pad 200 met een volgende hop van 172.31.63.250.



Opmerking: Elke lokaal gegenereerde vermelding, zoals 192.168.250.15, heeft een volgende hop van 0.0.0.0.

- Het teken > – geeft aan dat BGP de beste route heeft gekozen. BGP gebruikt de beslissingsstappen die zijn beschreven in het document BGP-algoritme voor selectie van het beste pad. BGP kiest één beste pad om een bestemming te bereiken, installeert het pad in de IP-routingtabel en kondigt het pad aan bij andere BGP-peers.



Opmerking: Let op het kenmerk next-hop. RTB is op de hoogte van 172.31.10.0 via een volgende hop van 172.31.63.250, wat de eBGP Next Hop is die wordt doorgegeven naar iBGP.

Bekijk de IP-routingtabel:

<#root>

RTB#

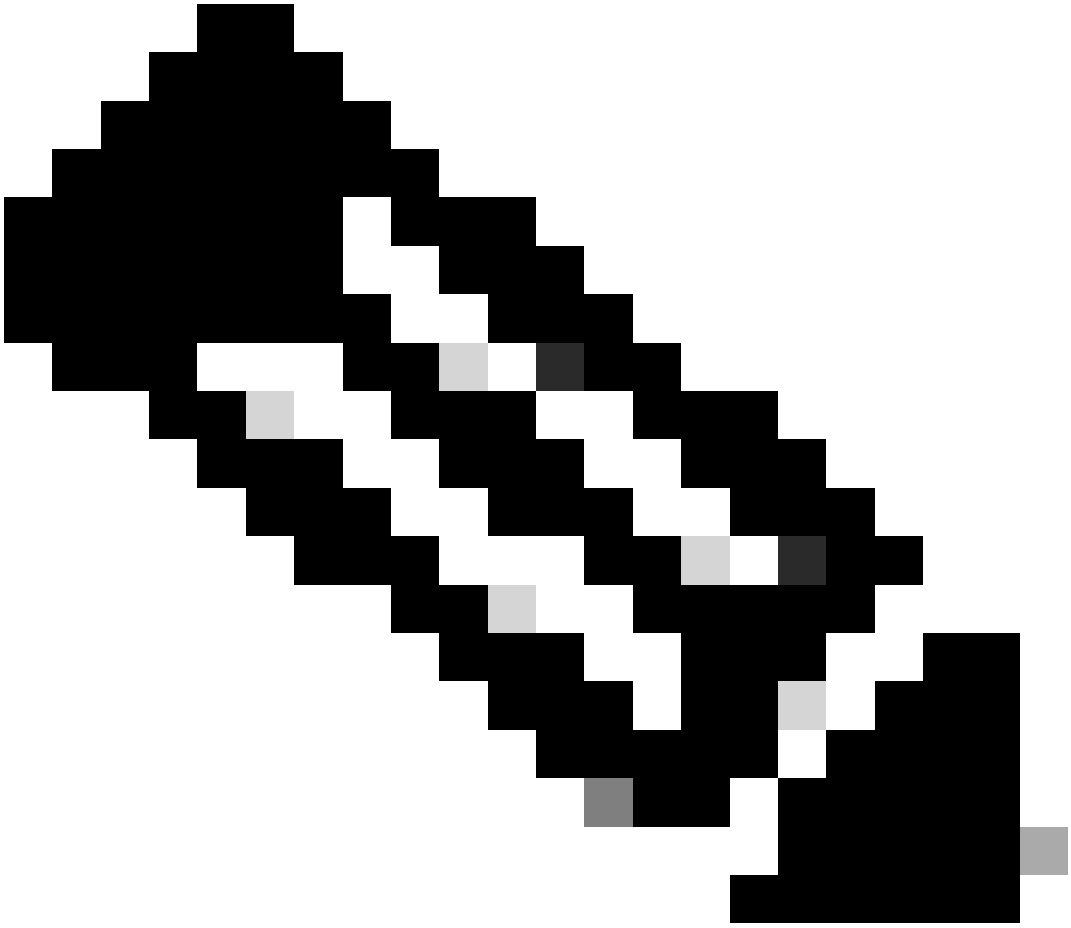
show ip route

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

Gateway of last resort is not set

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

Blijkbaar heeft geen van de BGP-vermeldingen de routingtabel bereikt. Er zijn twee problemen.

Het eerste probleem is dat de volgende hop voor deze vermeldingen, 172.31.63.250, onbereikbaar is. Die volgende hop kan niet worden bereikt via dit IGP, wat OSPF is. RTB heeft niet geleerd over 192.168.213.63 via OSPF. U kunt OSPF op de RTA s0 interface draaien en passief maken; op deze manier weet RTB hoe de volgende hop 172.31.63.250 te bereiken. Deze RTA-configuratie verschijnt hier:

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 192.168.203.250 255.255.255.0

interface Ethernet0
ip address 192.168.203.14 255.255.255.0

interface Serial0
 ip address 192.168.128.63 255.255.255.252

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

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



Opmerking: U kunt de opdracht tussen RTA en RTB uitgevenbgp nexthop self om de volgende hop te veranderen.

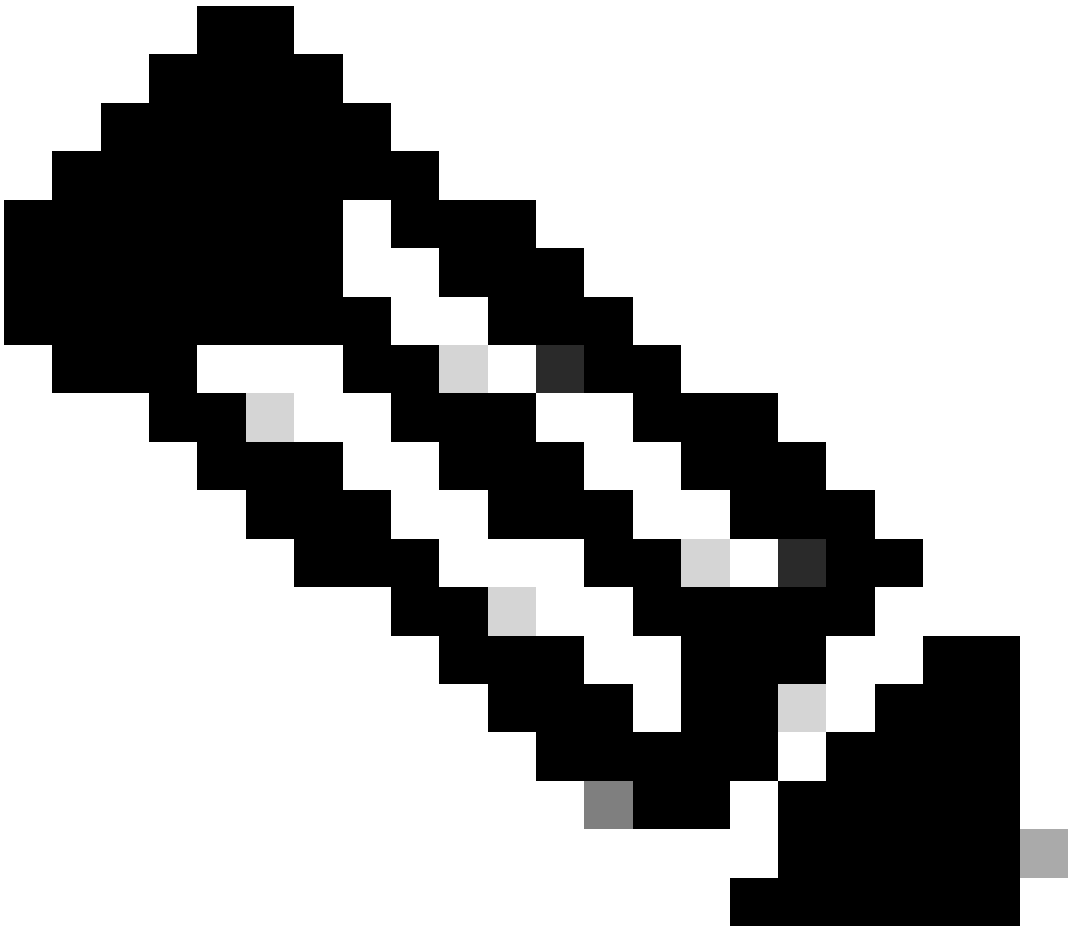
De nieuwe BGP-tabel op RTB ziet er als volgt uit:

<#root>

RTB#

show ip bgp

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

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



Opmerking: Alle vermeldingen hebben een >, wat betekent dat BGP de volgende hop kan bereiken.

Bekijk de routingtabel:

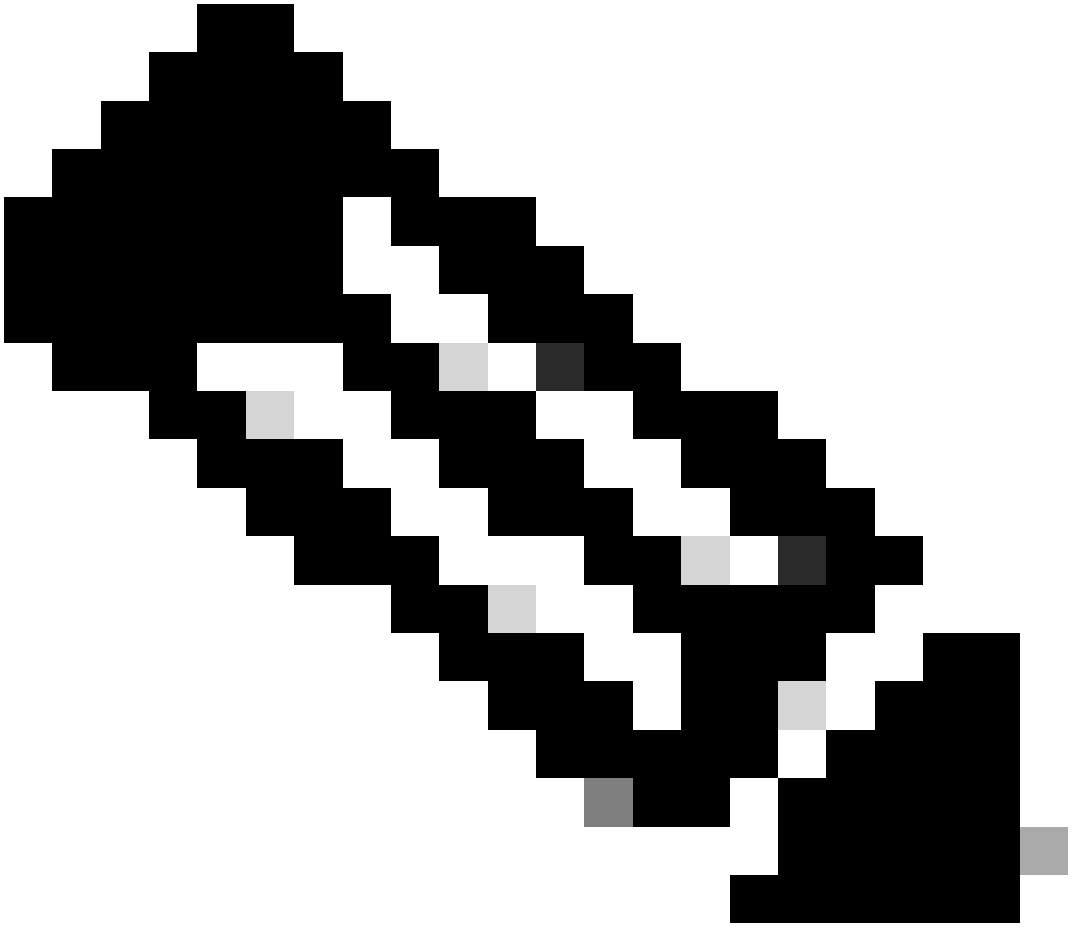
<#root>

RTB#

show ip route

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

Gateway of last resort is not set

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

Het tweede probleem is dat u BGP-vermeldingen in de routingtabel nog altijd niet ziet. Het enige verschil is dat 192.168.213.63 nu bereikbaar is via OSPF. Dit probleem is een synchronisatieprobleem. BGP plaatst deze vermeldingen niet in de routingtabel en verstuurt de vermeldingen niet in BGP-updates door het ontbreken van synchronisatie met het IGP.



Opmerking: RTF heeft geen informatie over netwerken 192.168.10.0 en 192.168.211.10 omdat u BGP nog niet heeft geherdistribueerd in OSPF.

Wanneer u in dit scenario synchronisatie uitschakelt, verschijnen de vermeldingen in de routingtabel. De connectiviteit blijft echter verbroken.

Als u de synchronisatie op RTB uitschakelt, gebeurt het volgende:

<#root>

RTB#

show ip route

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

Gateway of last resort is not set

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

De routingtabel ziet er goed uit, maar de netwerken kunnen niet worden bereikt. RTF in het midden weet niet hoe de netwerken kunnen worden bereikt:

```
    <#root>

RTF#
```

show ip route

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

```
Gateway of last resort is not set

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

Wanneer u de synchronisatie in deze situatie uitschakelt, blijft het probleem bestaan. Maar u heeft synchronisatie later nodig voor andere zaken.

Herdistribueer BGP in OSPF op RTA, en gebruik voor metric de waarde 2000:

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 192.168.203.250 255.255.255.0

interface Ethernet0
ip address 192.168.203.14 255.255.255.0

interface Serial0
 ip address 192.168.128.63 255.255.255.252

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

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

De routingtabel ziet er als volgt uit:

```
    <#root>

RTB#
```

show ip route

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

Gateway of last resort is not set

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

De BGP-vermeldingen zijn verdwenen omdat OSPF een betere afstand heeft dan iBGP. De OSPF-afstand is 110, terwijl de iBGP-afstand 200 is.

Schakel de synchronisatie op RTA uit zodat RTA 192.168.250.15 kan adverteren. Deze actie is noodzakelijk omdat RTA niet met OSPF wegens het verschil in maskers synchroniseert. Houd synchronisatie uit op RTB zodat RTB 192.168.203.13 kan adverteren. Deze actie is noodzakelijk op RTB om dezelfde reden.

Geef nu de RTB s1-interface weer om te zien hoe de routes eruit zien. Schakel ook OSPF op Serial1 van RTB in om dit passief te maken. Deze stap informeert RTA ook over de volgende hop 192.168.10.5 via IGP. Als u deze stap niet uitvoert, ontstaan er routinglussen omdat u de andere route via eBGP moet gebruiken, om de volgende hop 192.168.10.5 te bereiken. Dit zijn de nieuwe configuraties van RTA en RTB:

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 192.168.203.250 255.255.255.0

interface Ethernet0
ip address 192.168.203.14 255.255.255.0

interface Serial0
```

```
 ip address 192.168.128.63 255.255.255.252

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

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

RTB#
hostname RTB

ip subnet-zero

interface Serial0
 ip address 192.168.250.2 255.255.255.252

interface Serial1
 ip address 192.168.10.6 255.255.255.252

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

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

De BGP-tabellen zien er als volgt uit:

<#root>

```
RTA#
```

show ip bgp

```
BGP table version is 117, local router ID is 192.168.203.250
```

```
Status codes: s suppressed, d damped, h history, * valid, > best,
i -internal Origin codes: i - IGP, e - EGP, ? - incomplete

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

RTB#
```

show ip bgp

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

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

Er zijn meerdere manieren om uw netwerk te ontwerpen om te praten met de twee verschillende ISP's, AS200 en AS300. Eén manier is om een primaire ISP en een back-up ISP te hebben. U kunt gedeeltelijke routes leren van één van de ISP's en standaardroutes naar beide ISP's. In dit voorbeeld, ontvangt u gedeeltelijke routes van AS200 en alleen lokale routes van AS300. Zowel RTA als RTB genereren standaardroutes naar OSPF, met RTB als voorkeur vanwege de lagere metriek. Op deze manier kunt u het uitgaande verkeer tussen de twee ISP's goed verdelen.

Er kan asymmetrie optreden als het verkeer dat RTA verlaat, terugkomt via RTB. Dit kan gebeuren wanneer u dezelfde pool van IP-adressen en hetzelfde hoofdnetwerk gebruikt wanneer u met de twee ISP's communiceert. Door aggregatie kan uw gehele AS er van buitenaf als één geheel uitzien. Ingangspunten op uw netwerk zijn mogelijk via RTA of RTB. U kunt ontdekken dat al het inkomende verkeer naar uw AS binnenkomt via één punt, zelfs wanneer u meerdere punten naar het internet heeft. In het voorbeeld heeft u twee verschillende hoofdnetwerken wanneer u met de twee ISP's communiceert.

Een andere mogelijke oorzaak van asymmetrie is een verschil in de aangekondigde padlengte om uw AS te bereiken. De ene serviceprovider kan dichter bij een bepaalde bestemming zijn dan een andere. In het voorbeeld komt verkeer van AS400 dat uw netwerk als bestemming heeft altijd binnen via RTA vanwege het kortere pad. U kunt proberen die beslissing te beïnvloeden. Gebruik de opdracht set as-path om padnummers toe te

voegen aan uw updates en de padlengte langer te laten lijken. Maar met attributen zoals lokale voorkeur, metriek, of gewicht, kan AS400 het uitgangspunt hebben ingesteld op AS200. In dit geval is er niets dat je kunt doen.

Deze configuratie is de definitieve configuratie voor alle routers:

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 192.168.203.250 255.255.255.0

interface Ethernet0
 ip address 192.168.203.14 255.255.255.0

interface Serial0
 ip address 192.168.128.63 255.255.255.252

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

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

ip classless
ip default-network 172.31.200.200

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

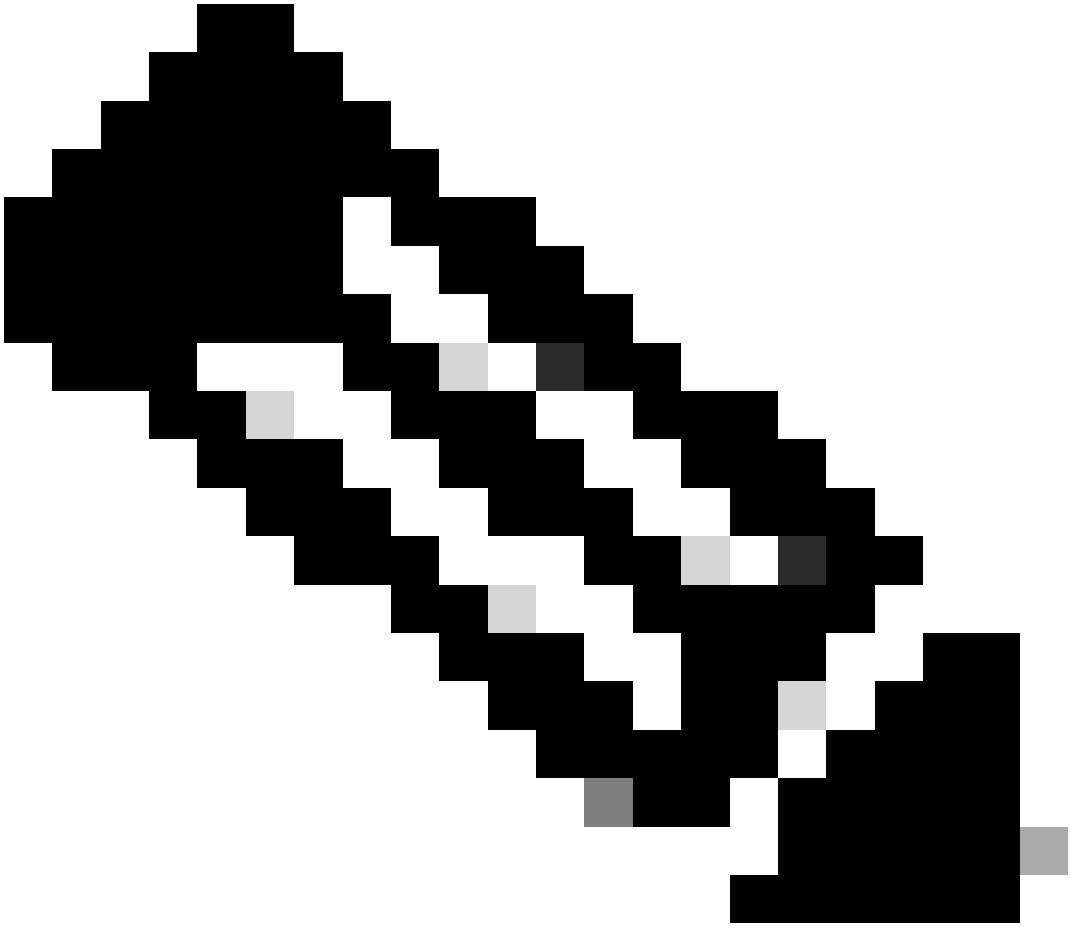
Op RTA is de lokale voorkeur voor routes die afkomstig zijn van AS200 vastgesteld op 200. Ook het netwerk 172.31.200.200 is de keuze voor de kandidaat-standaard. De opdracht ip default-network stelt u in staat de standaardwaarde te kiezen.

In dit voorbeeld wordt door het uitvoeren van de opdracht **default-information originate** **met OSPF de standaardroute binnen het OSPF-domein geïnjecteerd.** Dit voorbeeld gebruikt deze opdracht ook met het IS-IS-protocol (Intermediate System to Intermediate System) en BGP. Voor RIP is er een automatische herdistributie in RIP van 0.0.0.0, zonder aanvullende configuratie. Voor IGRP en EIGRP vindt de injectie van de standaardinformatie in het IGP-domein plaats na herdistributie van BGP in IGRP en EIGRP. Daarnaast kunt u met IGRP en EIGRP een statische route herdistribueren naar 0.0.0.0 in het IGP-domein.

```
RTF#
```

```
hostname RTF

ip subnet-zero

interface Ethernet0
 ip address 172.31.14.250 255.255.255.0

interface Serial1
 ip address 172.16.15.250 255.255.255.252

router ospf 10
 network 192.168.203.25 0.0.255.255 area 0

ip classless

RTB#
hostname RTB

ip subnet-zero

interface Loopback1
 ip address 172.16.15.2500 255.255.255.252

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

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

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

Voor RTB is de lokale voorkeur voor updates die afkomstig zijn van AS300 ingesteld op 300. Deze waarde is hoger dan de lokale voorkeurswaarde van iBGP-updates die afkomstig zijn van RTA. Zo kiest AS100 RTB voor de lokale routes van AS300. Eventuele andere routes op RTB, als er andere routes bestaan, verzenden intern met een lokale voorkeur van 100. Deze waarde is lager dan de lokale voorkeur van 200, die van RTA komt. RTA is de voorkeur.



Opmerking: U heeft alleen de lokale routes van AS300 aangekondigd. Alle padinformatie die niet overeenkomt met ^300$ wordt afgewezen. Als u de lokale routes en neighbor-routes, ofwel de klanten van de ISP, wilt aankondigen, gebruik dan ^300_[0-9]*.

Hier ziet u de output van de reguliere expressie die de lokale routes van AS300 aangeeft:

<#root>

RTB#

show ip bgp regexp ^300$

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

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

```
RTC#
hostname RTC

ip subnet-zero

interface Loopback0
 ip address 192.168.128.6330 255.255.255.192

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

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

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

Op RTC verzamelt u 172.31.10.0/16 en geeft u de specifieke injectieroutes voor AS100 aan. Als de ISP weigert deze taak uit te voeren, moet u filteren op het inkomende uiteinde van AS100.

```
RTD#
hostname RTD

ip subnet-zero

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

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

RTG#
hostname RTG

ip subnet-zero

interface Loopback0
 ip address 192.168.211.19574 255.255.255.192

interface Serial0
 ip address 192.168.10.1 255.255.255.252

interface Serial1
 ip address 192.168.211.195 255.255.255.252

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

Een demonstratie van hoe u communityfiltering gebruikt is op RTG te vinden. Je voegt een no-export community toe aan 192.168.211.0 updates voor OTO. Op die manier exporteert RTD deze route niet naar RTB. In dit geval accepteert RTB deze routes echter sowieso niet.

```
RTE#
hostname RTE

ip subnet-zero

interface Loopback0
 ip address 192.168.200.10 255.255.255.0

interface Serial0
 ip address 192.168.195.211 255.255.255.252

interface Serial1
 ip address 172.31.63.213 255.255.255.252

router bgp 400
 network 192.168.10.10
```

```
 aggregate-address 172.31.200.200 255.255.0.0 summary-only
 neighbor 172.16.31.5 remote-as 200
 neighbor 192.168.211.195 remote-as 500

ip classless
```

RTE-aggregaten 172.31.200.200/16. Hier zijn de definitieve BGP- en routeringstabellen voor RTA, RTF en RTB:

```
    <#root>


RTA#
```

show ip bgp

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

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

RTA#
```

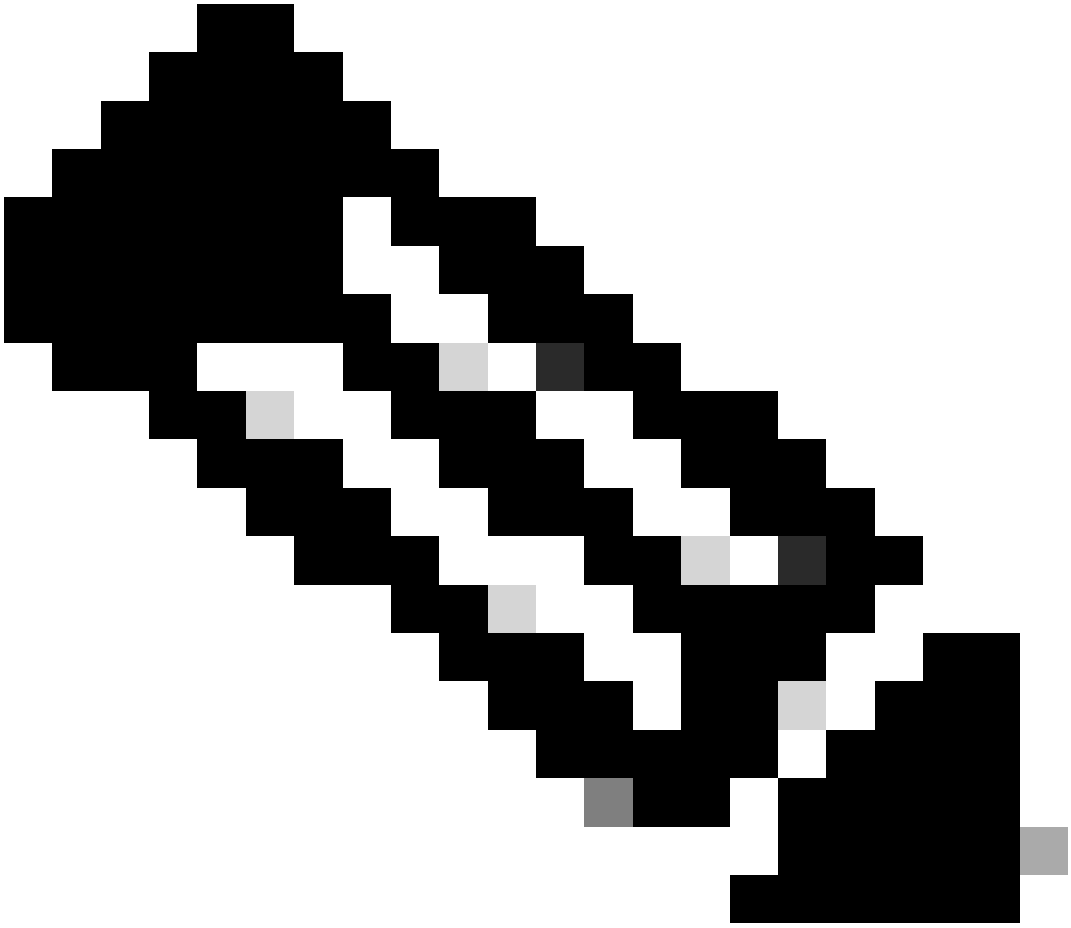
show ip route

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

Gateway of last resort is 172.31.63.250 to network 172.31.200.200

     192.168.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    192.168.10.0 255.255.255.0
```

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

RTF#
```

show ip route

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

Gateway of last resort is 192.168.250.2 to network 0.0.0.0

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



Opmerking: De RTF-routingtabel geeft aan dat netwerken die lokaal zijn voor AS300, zoals 192.168.10.0, via RTB worden bereikt. Andere bekende netwerken, zoals 172.31.200.200, moeten via RTA worden bereikt. De gateway of last resort is ingesteld op RTB. Als er iets gebeurt met de verbinding tussen RTB en RTD, dan wordt de door RTA aangekondigde standaard gebruikt met een metric-waarde van 2000.

<#root>

RTB#

show ip bgp

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

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

RTB#
```

show ip route

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

Gateway of last resort is 192.168.10.5 to network 192.168.10.0

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

Gerelateerde informatie

- **BGP: veelgestelde vragen**
- **Voorbeeldconfiguraties van BGP met een PIX-firewall**
- **HSRP gebruiken om redundantie te bieden in een BGP-omgeving met meerdere netwerken**
- **Configureer de redundantie van één routermodus en BGP op een Cat600 MSFC**
- **Optimale routing realiseren en BGP-geheugengebruik verminderen**
- **Probleemoplossing voor algemene BGP-problemen**
- **Problemen oplossen bij hoge CPU's als gevolg van het BGP-scanner- of routerproces**
- **Understand Load Share with BGP in Single and Multihomed Environments (Inzicht in workloadverdeling met BGP in omgevingen met één netwerk of meerdere netwerken)**
- **Ondersteuningspagina voor BGP**
- **Cisco technische ondersteuning en downloads**

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.