Studio degli scenari d'uso del protocollo BGP (Border Gateway Protocol)

Sommario

Introduzione

In questo documento vengono illustrati cinque scenari d'uso del protocollo BGP (Border Gateway Protocol).

Prerequisiti

Requisiti

Nessun requisito specifico previsto per questo documento.

Componenti usati

Il documento può essere consultato per tutte le versioni software o hardware.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento Cisco sulle convenzioni

nei suggerimenti tecnici.

Case study BGP 1

Il BGP, definito in RFC 1771, consente di creare un routing interdominio privo di loop tra sistemi autonomi (AS). Un AS è l'insieme dei router di un'unica amministrazione tecnica. I router in un AS possono utilizzare più IGP (Interior Gateway Protocol) per scambiare informazioni di routing all'interno dell'AS. I router possono utilizzare un protocollo gateway esterno per instradare i pacchetti all'esterno dell'AS.

Come funziona il BGP?

BGP utilizza TCP come protocollo di trasporto sulla porta 179. Due router BGP formano una connessione TCP tra loro. Questi router sono router peer. I router peer si scambiano messaggi per aprire e confermare i parametri di connessione.

I router BGP si scambiano informazioni sulla raggiungibilità della rete. Queste informazioni sono principalmente un'indicazione dei percorsi completi che un indirizzamento deve seguire per raggiungere la rete di destinazione. I percorsi sono numeri AS BGP. Queste informazioni consentono di creare un grafico di AS privi di loop. Il grafico mostra anche dove applicare le policy di indirizzamento per introdurre limitazioni sul comportamento di routing.

Coppie di router che formano una connessione TCP per scambiarsi informazioni di instradamento BGP sono detti "peer" o "neighbor". I peer BGP si scambiano inizialmente le tabelle di routing BGP complete. In seguito, inviano aggiornamenti incrementali in caso di modifiche delle tabelle di routing. BGP conserva il numero di versione della tabella BGP. Il numero di versione è lo stesso per tutti i peer BGP. Il numero di versione cambia ogni volta che BGP aggiorna la tabella con le informazioni di routing modificate. L'invio di pacchetti keepalive garantisce che la connessione tra peer BGP sia attiva. In risposta a errori o condizioni speciali vengono inviati pacchetti di notifica.

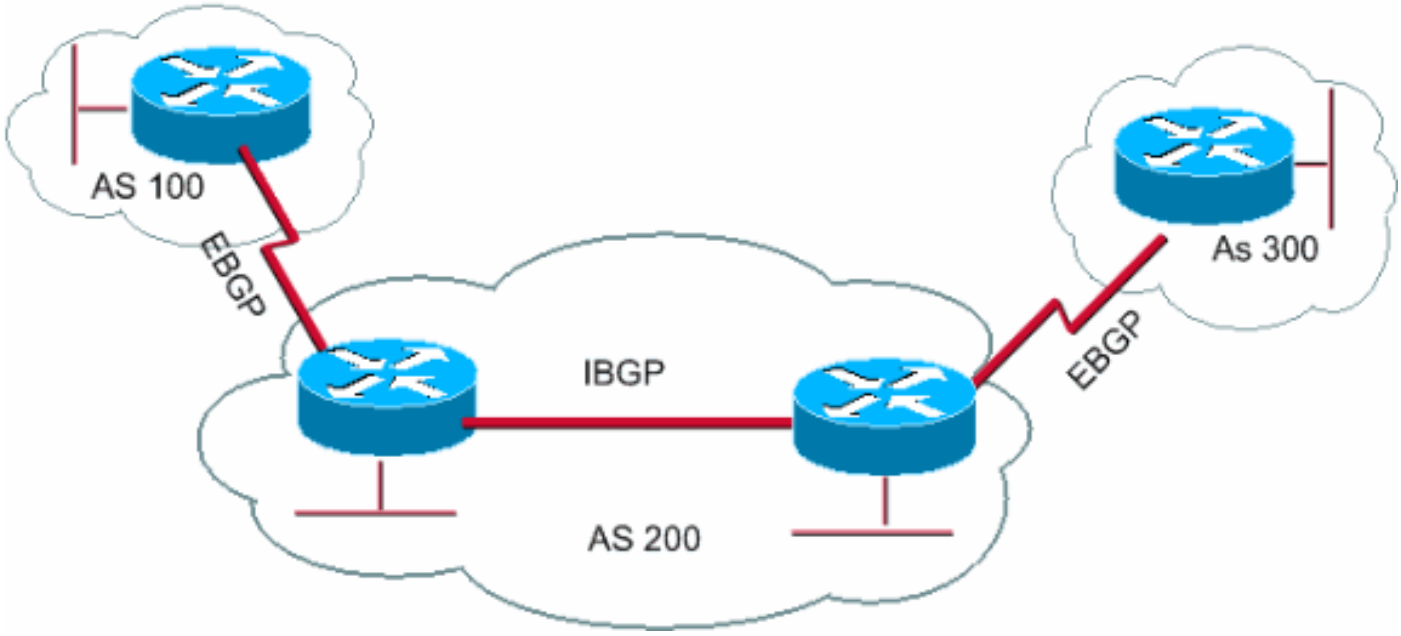
eBGP e iBGP

Se un AS ha più speaker BGP, può fungere transito per altri AS. Come mostra lo schema seguente in questa sezione, AS200 è un AS100 di transito e AS300 di transito.

Per inviare le informazioni ad AS esterni, è necessario garantire la raggiungibilità delle reti. I seguenti processi garantiscono la raggiungibilità della rete:

- Peering BGP (iBGP) interno tra router all'interno di un AS
- Ridistribuzione delle informazioni BGP agli IGP eseguiti nell'AS

Quando BGP viene eseguito tra router appartenenti a due AS diversi, è chiamato BGP esterno (eBGP). Quando BGP viene eseguito tra router nello stesso AS, è chiamato iBGP.



BGP in esecuzione tra router nello stesso AS

Abilitare il routing del BGP

Completare questi passaggi per abilitare e configurare il BGP.

Si supponga di volere due router, RTA e RTB, che comunicano tramite BGP. Nel primo esempio, RTA e RTB sono in AS diversi. Nel secondo esempio, entrambi i router appartengono allo stesso AS.

1. Definire il processo del router e il numero AS a cui appartengono i router.

   Utilizzare questo comando per abilitare il BGP su un router:

   <#root>

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

   ```
   RTA#
   router bgp 100

   RTB#
   router bgp 200
   ```

   Queste istruzioni indicano che RTA esegue BGP e appartiene a AS100. RTB esegue BGP e appartiene a AS200.

2. Definizione di BGP neighbor.

   I BGP neighbor indicano quali router tentano di comunicare tramite BGP. Nella sezione successiva viene illustrato questo processo.

Creare BGP neighbor

Due router BGP diventano neighbor dopo aver stabilito una connessione TCP. La connessione TCP è essenziale per consentire ai due router peer di avviare lo scambio degli aggiornamenti di routing.

Una volta attivata la connessione TCP, i router inviano messaggi aperti per scambiarsi i valori. I valori scambiati dai router includono il numero AS, la versione BGP eseguita dai router, l'ID del router BGP e il tempo di attesa keepalive. Dopo aver confermato e accettato questi valori, viene stabilita la connessione neighbor. Qualsiasi stato diverso da Established (Stabilita) indica che i due router non sono diventati neighbor e che non possono scambiarsi aggiornamenti BGP.

Per stabilire una connessione TCP, eseguire questo neighbor comando:

<#root>

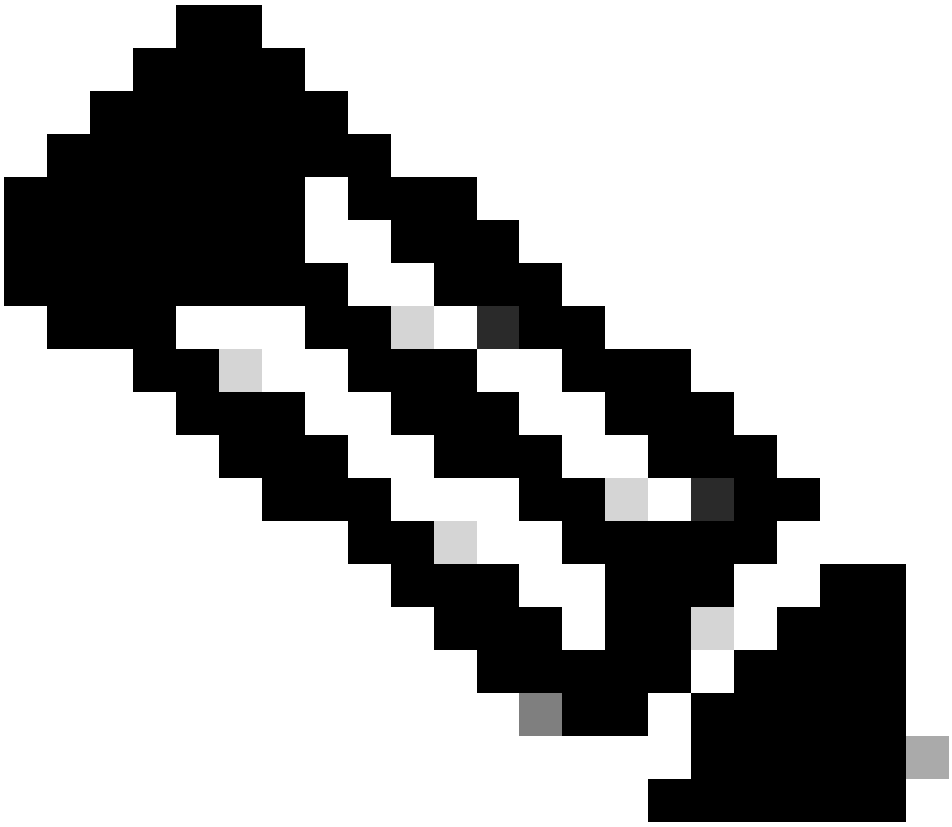
neighbor <ip-address> remote-as <number>

Il valore number del comando è il numero AS del router a cui si desidera connettersi con il BGP. Il valore ip-address è l'indirizzo next-hop con connessione diretta per eBGP. Per iBGP, ip-address è qualsiasi indirizzo IP sull'altro router.

I due indirizzi IP utilizzati nel comando neighbor dei router peer *devono* essere in grado di comunicare tra loro. Un modo per verificare la raggiungibilità è un ping esteso tra i due indirizzi IP. Il comando ping esteso forza il router di ping a usare come origine l'indirizzo IP specificato dal neighbor comando. Il router deve utilizzare questo indirizzo anziché l'indirizzo IP dell'interfaccia da cui proviene il pacchetto.

In caso di modifiche alla configurazione BGP, è necessario reimpostare la connessione neighbor per rendere effettivi i nuovi parametri. .

- clear ip bgp address



Nota: l'indirizzo è l'indirizzo adiacente

•

clear ip bgp *

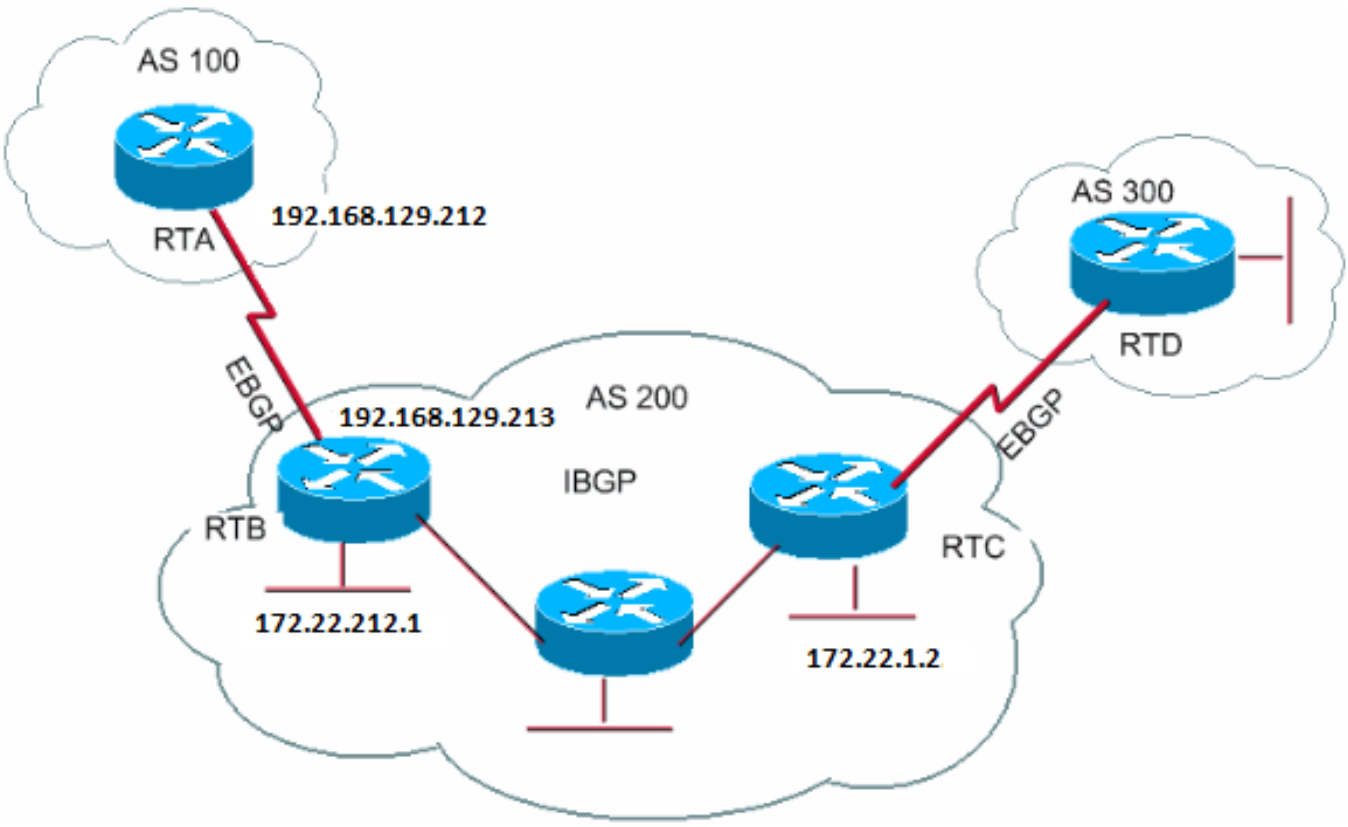
Questo comando elimina tutte le connessioni neighbor.

Per impostazione predefinita, le sessioni BGP iniziano con l'uso di BGP versione 4 e negoziano le versioni precedenti, se necessario. È possibile impedire le negoziazioni e forzare la versione BGP utilizzata dai router per comunicare con un neighbor. Utilizzare questo comando nella modalità di configurazione del router:

<#root>

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

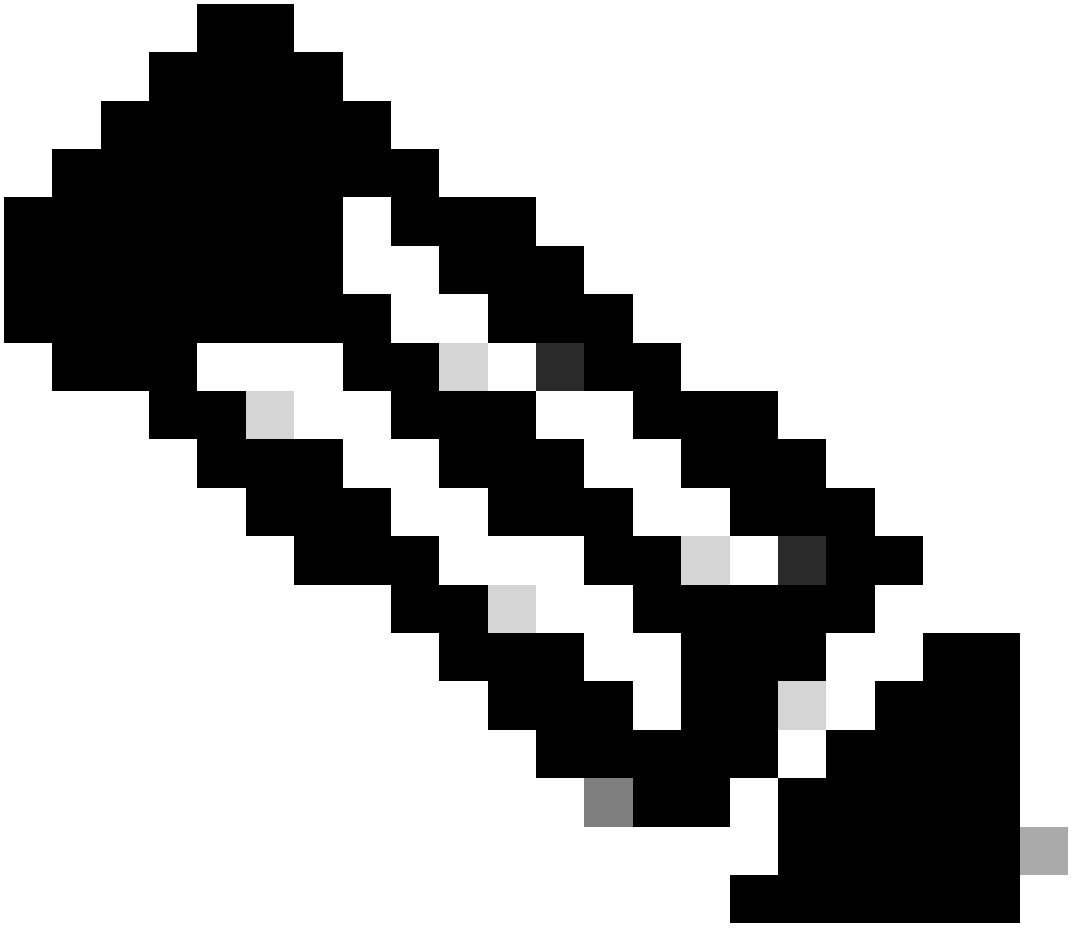
Di seguito è riportato un esempio della configurazione del neighbor comando:



```
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 questo esempio, RTA e RTB eseguono eBGP. RTB e RTC eseguono iBGP. Il numero AS remoto punta a un AS esterno o interno, che indica eBGP o iBGP. Inoltre, i peer eBGP dispongono di una connessione diretta, ma i peer iBGP non dispongono di una connessione diretta. I router iBGP non devono disporre di una connessione diretta. Ma deve esserci un'IGP che funzioni e permetta ai due vicini di raggiungersi l'un l'altro.

Questa sezione fornisce un esempio delle informazioni visualizzate dal comando show ip bgp neighbors .



Nota: prestare particolare attenzione allo stato del BGP. Qualsiasi stato diverso da Stabilito indica che i peer non sono attivi. Inoltre, tenere presente quanto segue:

- La versione BGP, che è 4
- L'ID del router remoto

  Questo numero è l'indirizzo IP più alto sul router o l'interfaccia loopback più alta, se presente.
- La versione della tabella

  La versione della tabella fornisce lo stato della tabella. Ogniqualvolta arrivano nuove informazioni, la versione della tabella aumenta. Una versione che continua ad aumentare indica che è presente un flapping della route che causa l'aggiornamento continuo degli indirizzamenti.

<#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 e interfacce loopback

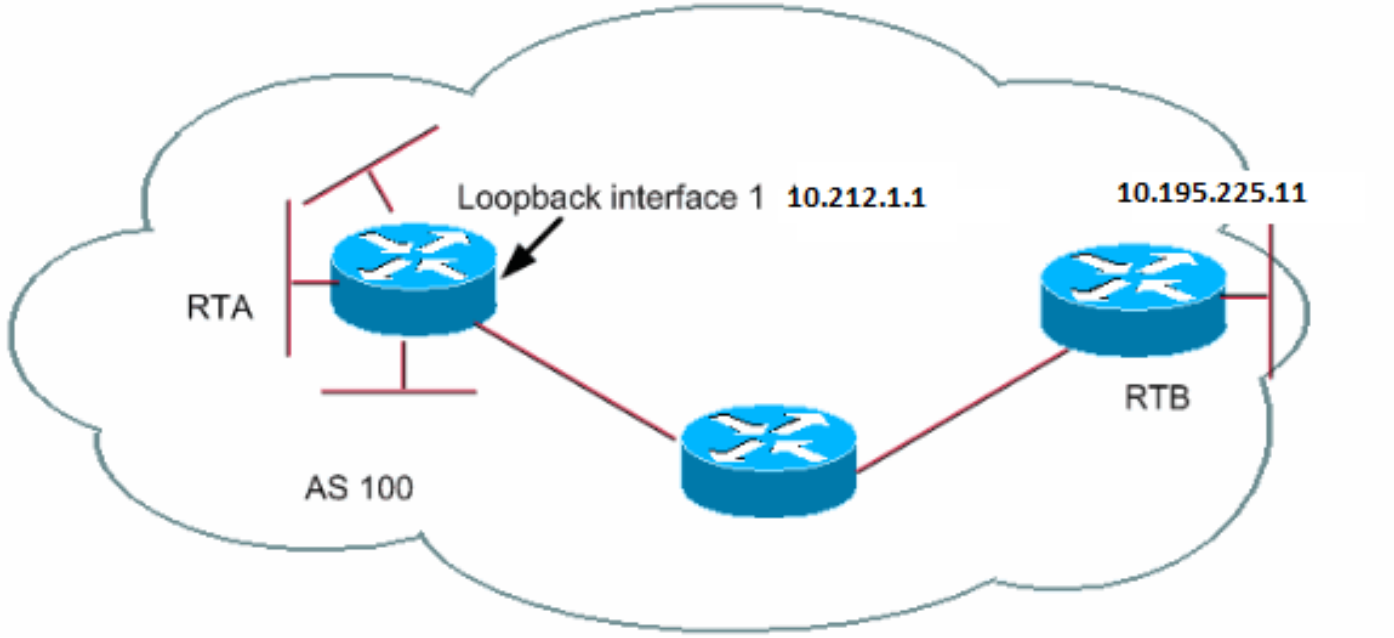
L'utilizzo di un'interfaccia di loopback per definire i router adiacenti è comune con iBGP, ma non con eBGP. Normalmente, si utilizza l'interfaccia loopback per assicurarsi che l'indirizzo IP del neighbor sia attivo e indipendente dall'hardware che funziona correttamente. Nel caso di eBGP, i router peer sono spesso dotati di una connessione diretta e il loopback non serve.

Se si utilizza l'indirizzo IP di un'interfaccia di loopback nel neighbor comando, è necessaria una configurazione aggiuntiva sul router adiacente. Il router neighbor deve segnalare al BGP l'utilizzo di un'interfaccia loopback in luogo di un'interfaccia fisica per avviare la connessione TCP con il BGP neighbor. Per indicare un'interfaccia loopback, utilizzare questo comando:

<#root>

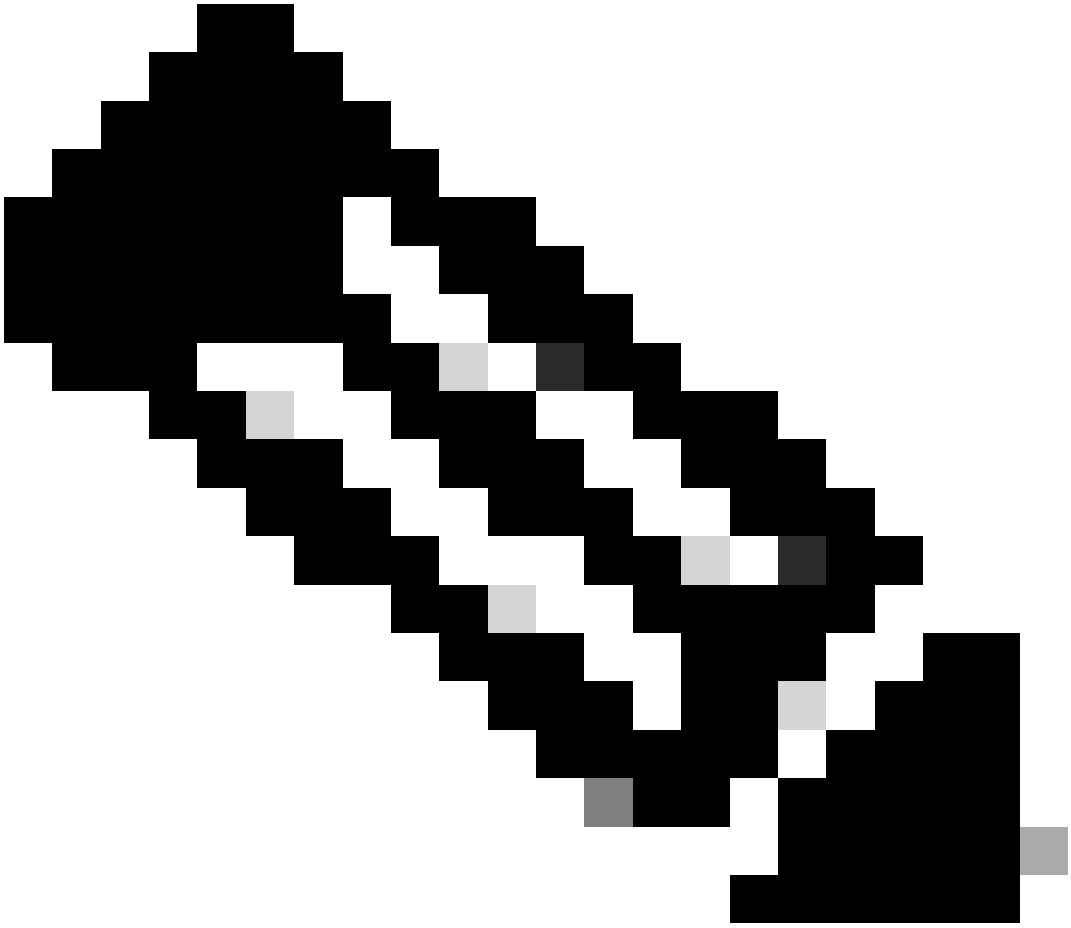
neighbor <ip-address> update-source <interface>

Questo esempio illustra l'uso del comando:




```
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
```

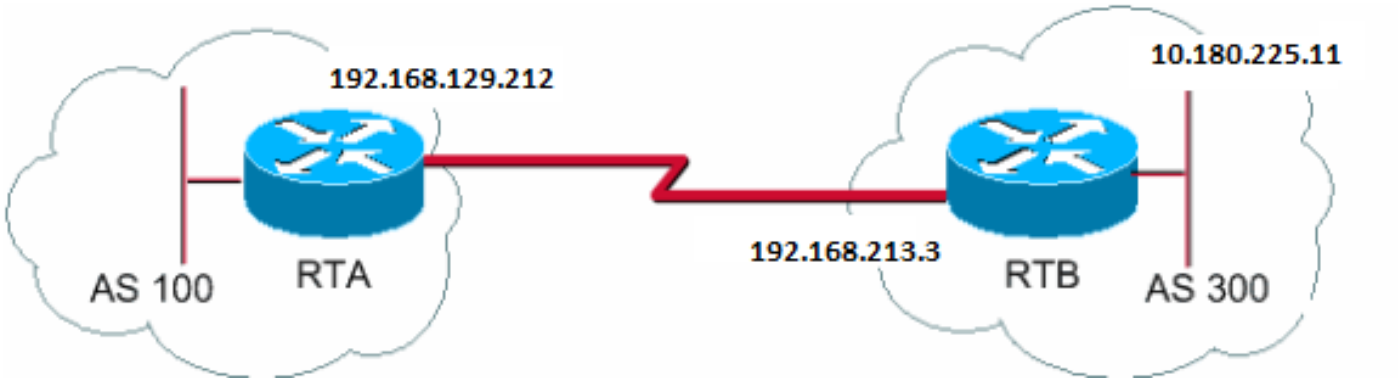
Nell'esempio, RTA e RTB eseguono iBGP all'interno di AS100. Nel neighbor comando, RTB utilizza l'interfaccia di loopback di RTA, 10.212.1.1. In questo caso, l'aggregazione in tempo reale deve imporre a BGP di utilizzare l'indirizzo IP di loopback come origine nella connessione adiacente TCP. Per imporre questa azione, l'aggregazione in tempo reale (RTA) aggiunge **update-source interface-type interface-number** in modo che il comando sia neighbor 10.195.225.11 update-source loopback 1. Questa istruzione forza BGP a utilizzare l'indirizzo IP dell'interfaccia di loopback quando BGP comunica con il router adiacente 10.195.225.11.



Nota: RTA ha utilizzato l'indirizzo IP dell'interfaccia fisica di RTB, 10.195.225.11, come neighbor. L'utilizzo di questo indirizzo IP è il motivo per cui RTB non richiede alcuna configurazione speciale. Fare riferimento a Configurazione di esempio per iBGP e eBGP con o senza un indirizzo loopback per una configurazione di esempio completa della rete.

Multihop eBGP

In alcuni casi, un router Cisco può eseguire eBGP con un router di terze parti che non consente la connessione diretta dei due peer esterni. Per ottenere la connessione, è possibile utilizzare il multihop eBGP. Il multihop eBGP consente una connessione neighbor tra due peer esterni che non dispongono di connessione diretta. Il multihop è solo per eBGP e non per iBGP. Questo esempio illustra il multihop eBGP:

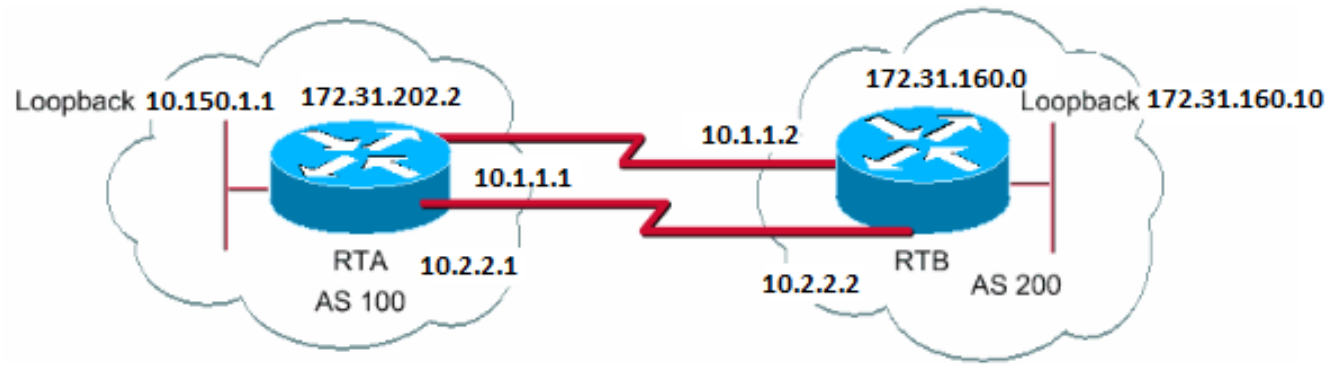



```
RTA#
router bgp 100
 neighbor 10.180.225.11 remote-as 300
 neighbor 10.180.225.11 ebgp-multihop

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

RTA indica un neighbor esterno senza connessione diretta. RTA deve indicare l'uso del comando neighbor ebgp-multihop . D'altra parte, RTB indica un router adiacente con connessione diretta, ovvero 192.168.129.212. A causa di questa connessione diretta, RTB non ha bisogno del neighbor ebgp-multihop comando. È inoltre necessario configurare un routing IGP o statico per consentire ai router adiacenti senza connessione di raggiungere gli altri.

L'esempio nella sezione Multihop BGP (bilanciamento del carico) mostra come ottenere il bilanciamento del carico con BGP nel caso in cui si disponga di eBGP su linee parallele.

Multihop eBGP (bilanciamento del carico)




```
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
```

In questo esempio viene illustrato l'utilizzo delle interfacce di loopback update-source, e ebgp-multihop. L'esempio è una soluzione alternativa per ottenere il bilanciamento del carico tra due speaker eBGP su linee seriali parallele. In situazioni normali, il BGP sceglie una delle linee su cui inviare i pacchetti e il bilanciamento del carico non avviene. Con l'introduzione delle interfacce loopback, il next-hop per eBGP è l'interfaccia loopback. Si utilizzano indirizzamenti statici, o IGP, per introdurre due percorsi equivalenti per raggiungere la destinazione. L'RTA ha due scelte per raggiungere l'hop successivo 172.31.160.10: un percorso tramite 10.1.1.2 e l'altro percorso tramite 10.2.2.2. RTB offre le stesse opzioni.

Route map

L'uso di route map con BGP è molto diffuso. Nel contesto BGP, la route map è un metodo per controllare e modificare le informazioni di routing. Il controllo e la modifica delle informazioni di routing avviene tramite la definizione delle condizioni per la ridistribuzione della route da un protocollo di routing a un altro. In alternativa, il controllo delle informazioni di routing può avvenire al momento dell'immissione e dell'uscita di BGP. Di seguito è riportato il formato della mappa dei percorsi:

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

Il tag della mappa è semplicemente un nome assegnato alla route map. È possibile definire più istanze della stessa route map o dello stesso tag

nome. Il numero di sequenza è semplicemente un'indicazione della posizione che una nuova route map deve avere nell'elenco di route map già configurate con lo stesso nome.

In questo esempio, esistono due istanze di route map definite, con il nome MYMAP. La prima istanza ha un numero di sequenza 10 e la seconda ha un numero di sequenza 20.

- **route-map MYMAP permit 10** (la prima serie di condizioni va qui.)

- **route-map MYMAP permit 20** (la seconda serie di condizioni va qui.)

Quando si applica la mappa dei percorsi MYMAP ai percorsi in entrata o in uscita, il primo insieme di condizioni viene applicato tramite l'istanza 10. Se il primo insieme di condizioni non viene soddisfatto, si passa a un'istanza superiore della mappa dei percorsi.

Comandi di configurazione match e set

Ogni mappa di percorso è costituita da un elenco di comandi match e di set configurazione. La corrispondenza specifica un match criterio e set specifica un'set azione se vengono soddisfatti i criteri applicati dal match comando.

Ad esempio, è possibile definire una route map che controlla gli aggiornamenti. Se esiste una corrispondenza per l'indirizzo IP 10.1.1.1, la metrica per l'aggiornamento viene impostata su 5. Questi comandi illustrano l'esempio:

<#root>

match ip address 10.1.1.1

set metric 5

Se i criteri di corrispondenza sono soddisfatti e si dispone di un'permit, si verifica una ridistribuzione o un controllo delle route, come specificato dall'azione di impostazione. Si esce dall'elenco.

Se i criteri di corrispondenza sono soddisfatti e si dispone di un deny, non vi è alcuna ridistribuzione o controllo della route. Si esce dall'elenco.

Se i criteri di corrispondenza non vengono soddisfatti e si dispone di un permit odeny , viene controllata l'istanza successiva della mappa del percorso. Ad esempio, viene selezionata l'istanza 20. Questo controllo di istanza successiva continua finché non si interrompono o terminano tutte le istanze della route map. Se si finisce l'elenco senza una corrispondenza, il percorso è not accepted nor forwarded.

Nel software Cisco IOS® versioni precedenti alla 11.2, quando si usano le route map per filtrare gli aggiornamenti BGP anziché ridistribuire i protocolli, non è possibile filtrare il traffico in entrata quando si usa un comando **match** sull'indirizzo IP. Si accetta un filtro in uscita. Il software Cisco IOS, versione 11.2 e successive, non prevede questa limitazione.

I comandi correlati per match sono:

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

matchmetric

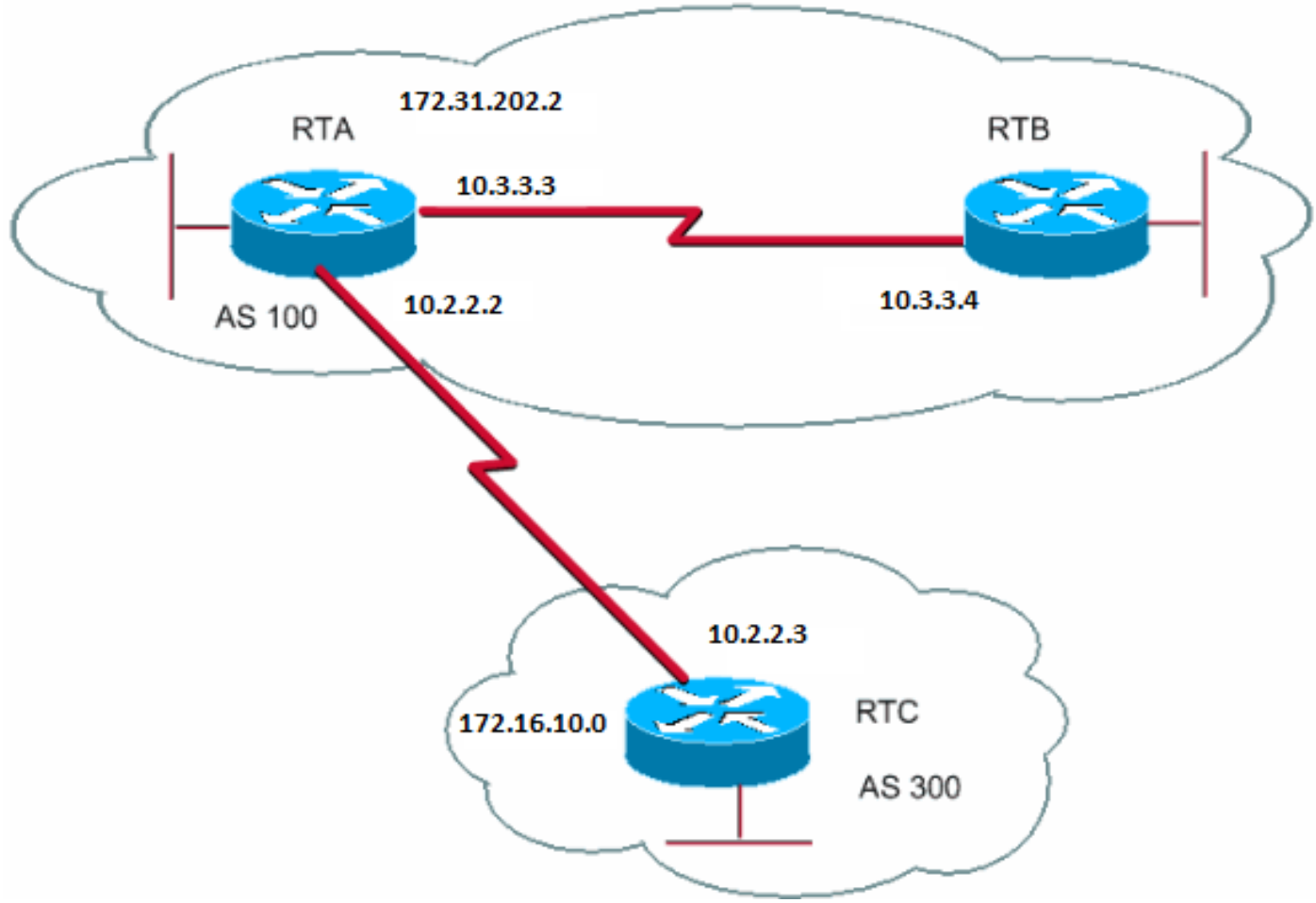
- match route-type
- match tag

I comandi correlati per set sono:

- 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

Seguono alcuni esempi di route map:



Esempi di mappe percorsi

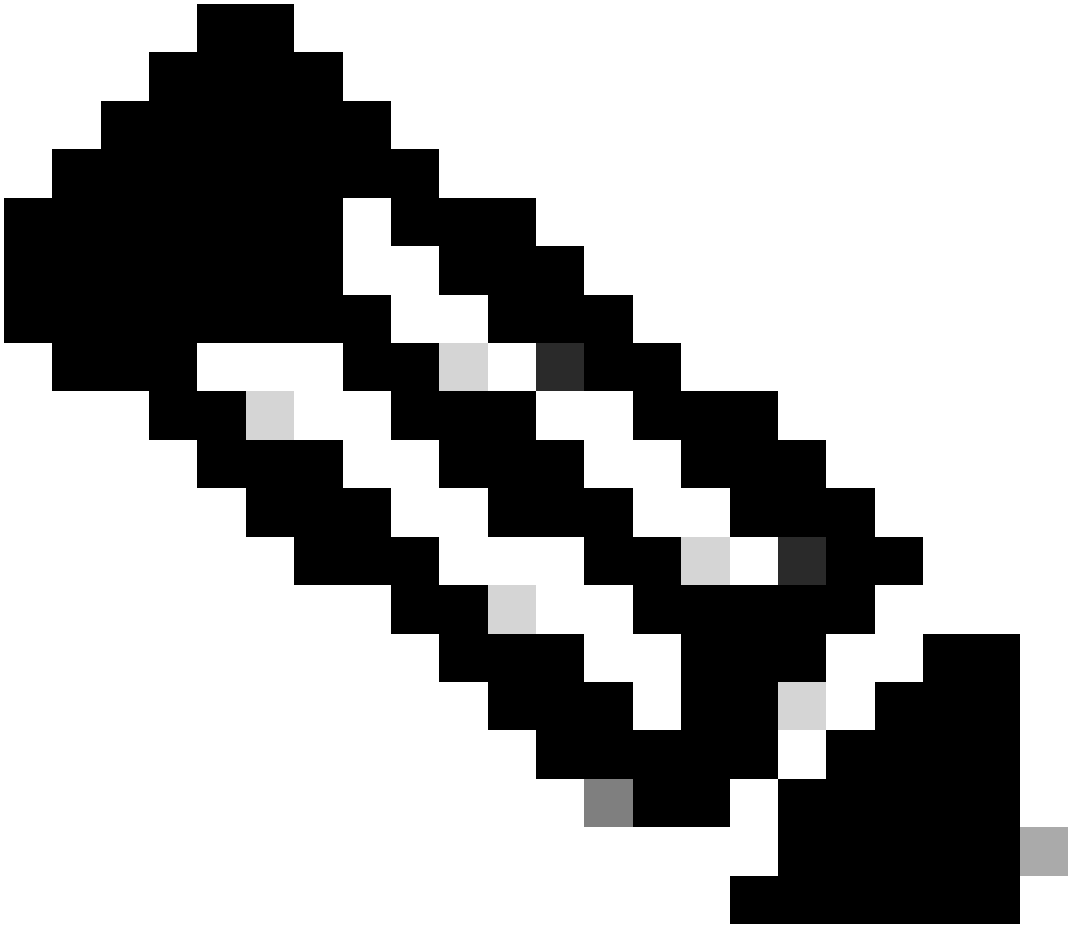
Esempio 1

Si supponga che RTA e RTB eseguano il protocollo RIP (Routing Information Protocol) e che RTA e RTC eseguano BGP. RTA ottiene gli aggiornamenti tramite BGP e ridistribuisce gli aggiornamenti a RIP. Si supponga che l'aggregazione in tempo reale desideri ridistribuire alle route RTB circa 172.16.10.0 con una metrica di 2 e tutte le altre route con una metrica di 5. In questo caso, è possibile utilizzare la configurazione seguente:

```
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
```

Nell'esempio, se una route corrisponde all'indirizzo IP 172.16.10.0, la metrica della route è 2. Quindi, si esce dall'elenco delle mappe percorsi. Se non vi sono corrispondenze, si procede verso il basso lungo l'elenco di mappe di route, che indica che tutto il resto è impostato sulla metrica 5.



Nota: porsi sempre la domanda "Che cosa succede alle route che non corrispondono a nessuna delle istruzioni di corrispondenza?" Queste route vengono eliminate per impostazione predefinita.

Esempio 2

Si supponga che nell'esempio 1 non si desideri che AS100 accetti gli aggiornamenti relativi a 172.16.10.0. Non è possibile applicare route map ai percorsi in ingresso se si utilizza come base un indirizzo IP. Pertanto, è necessario utilizzare una route map in uscita su 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
```

Ora che è noto come avviare il BGP e come definire un neighbor, vediamo come avviare lo scambio di informazioni di rete.

Esistono diversi modi per inviare le informazioni di rete con l'uso di BGP. Queste sezioni illustrano i vari metodi uno a uno:

- Comando network
- Ridistribuzione
- Indirizzamenti statici e ridistribuzione

Comando network

Il formato del network comando è:

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

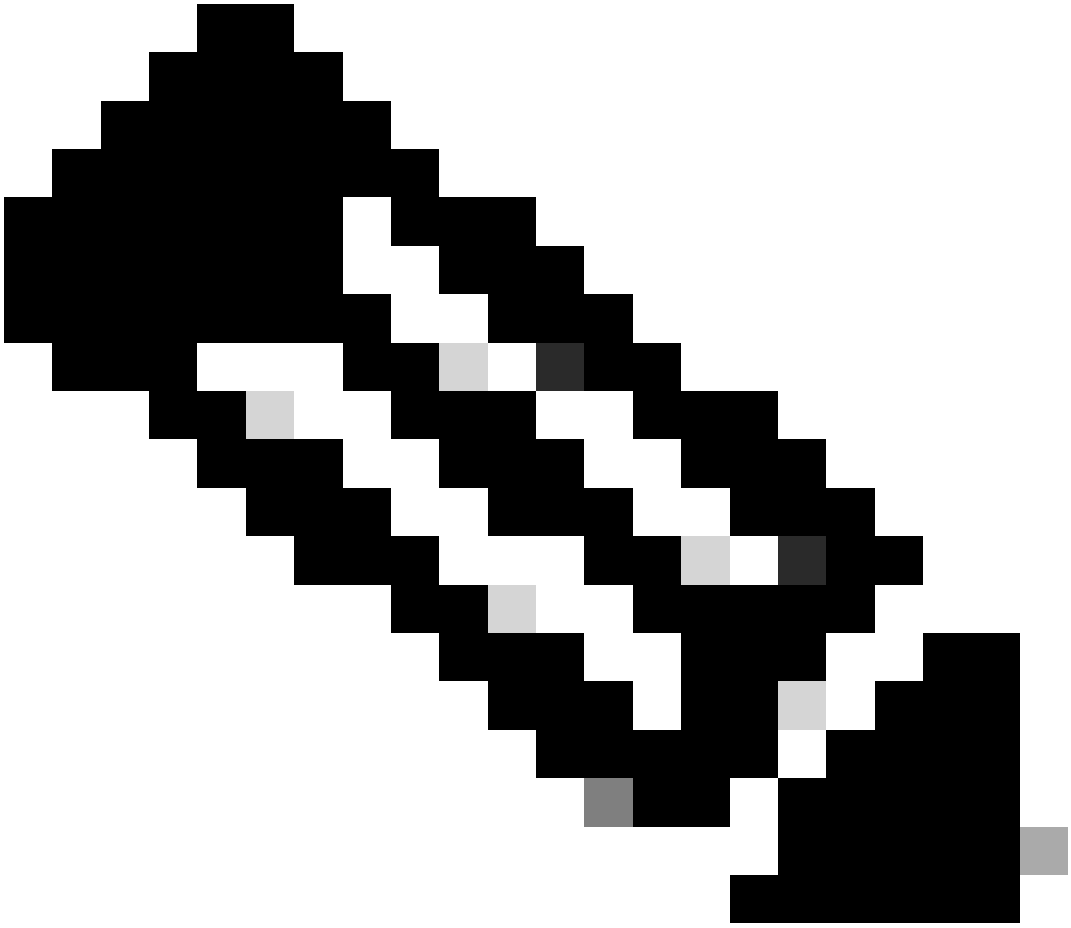
Il network comando controlla le reti che hanno origine da questa casella. Questo caso è diverso dalla configurazione familiare di IGRP (Interior Gateway Routing Protocol) e RIP. Con questo comando, non si cerca di eseguire il BGP su una determinata interfaccia. Si cerca invece di indicare a BGP quali reti BGP devono avere origine da questa casella. Il comando utilizza una porzione di maschera, perché il BGP versione 4 (BGP4) può gestire il subnetting e il supernetting. È accettabile un massimo di 200 voci del network comando.

Il network comando funziona se il router conosce la rete che si sta tentando di annunciare, sia essa connessa, statica o appresa dinamicamente.

Un esempio del comando network è:

```
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
```

Nell'esempio, il router A genera una voce di rete per 192.168.213.0/16. Il parametro /16 indica che viene utilizzata una supernet dell'indirizzo di classe C e che vengono annunciati i primi due ottetti, o i primi 16 bit.

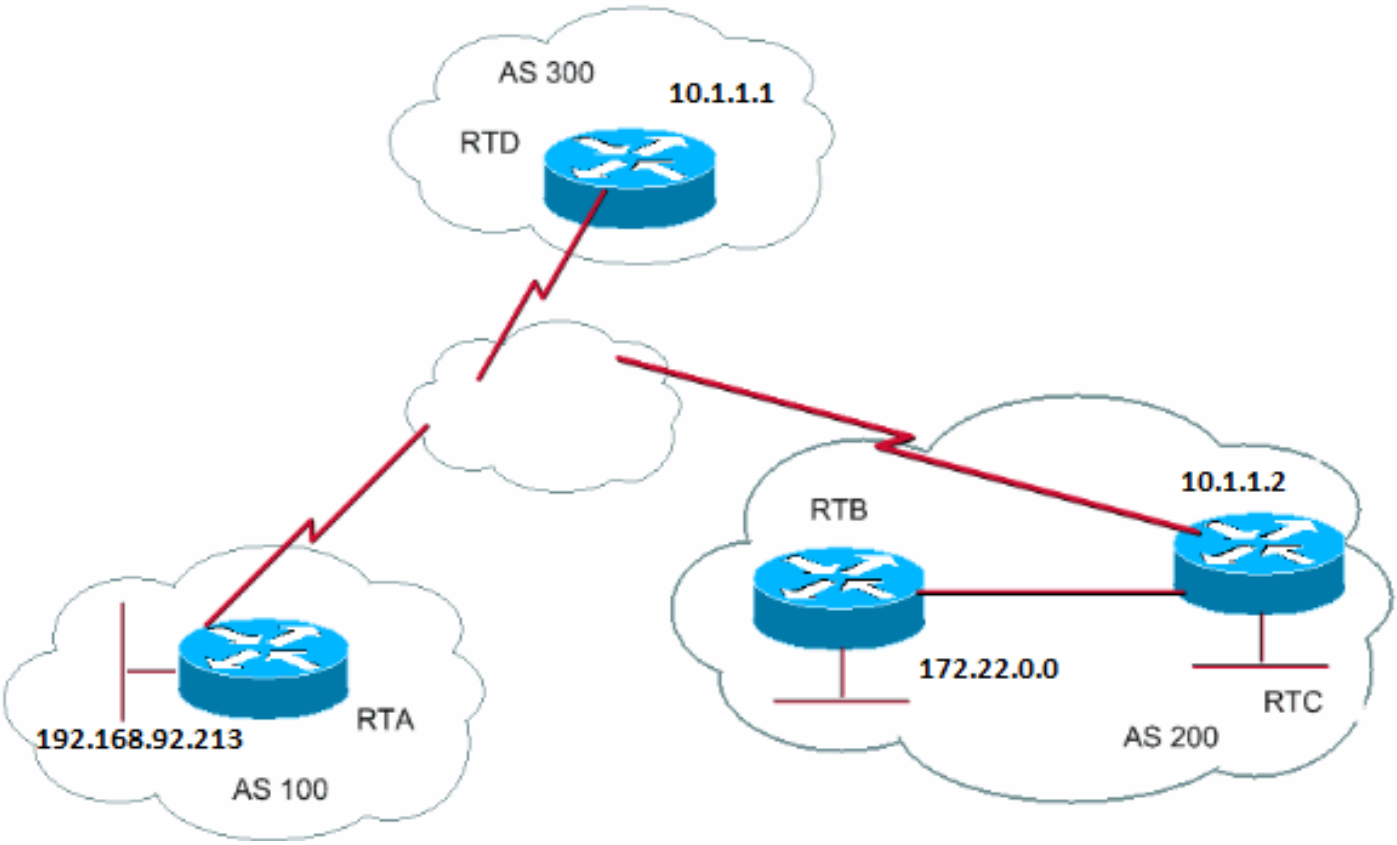


Nota: è necessario che l'indirizzamento statico raggiunga il router per generare 192.168.213.0, in quanto l'indirizzamento statico inserisce una voce corrispondente nella tabella di routing.

Ridistribuzione

Il network comando è uno dei modi per annunciare le reti tramite BGP. Un altro modo è ridistribuire l'IGP nel BGP. L'IGP può essere IGRP, protocollo OSPF (Open Shortest Path First), RIP, EIGRP (Enhanced Interior Gateway Routing Protocol) o un altro protocollo. Questa ridistribuzione può sembrare spaventosa perché ora si scaricano tutte le route interne in BGP; alcune di queste route possono essere state apprese tramite BGP e non è necessario inviarle di nuovo. Prestare attenzione quando si applica un filtro per assicurarsi di inviare i messaggi alle route Internet che si desidera annunciare e non a tutte le route disponibili. Ecco un esempio.

RTA annuncia 192.168.92.213 e RTC annuncia 172.22.0.0. Osservare la configurazione RTC:



Se si usa il networkcomando, si hanno:

```
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.
```

Se invece si utilizza la ridistribuzione, si ha:

```
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.
```

Questa ridistribuzione causa l'origine di 192.168.92.213 dall'AS. L'utente non è la fonte di 192.168.92.213; AS100 è la fonte. Pertanto, è necessario utilizzare filtri per impedire che l'appliance ASA estragga la sorgente dalla rete. La configurazione corretta è:

```
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
```

Il access-listcomando consente di controllare le reti che hanno origine da AS200.

La ridistribuzione di OSPF in BGP è leggermente diversa dalla ridistribuzione per altri IGP. La semplice questione di redistribute ospf 1sotto non router bgpfunziona. internal Per ridistribuire le rispettive route, **nssa-external** sono necessarie parole chiave specifiche, ad esempio, external e. Per ulteriori informazioni, fare riferimento a Informazioni sulla ridistribuzione delle route OSPF in BGP.

Indirizzamenti statici e ridistribuzione

Per creare una rete o una subnet è sempre possibile utilizzare indirizzamenti statici. L'unica differenza è che BGP ritiene che queste route abbiano un'origine incompleta o sconosciuta. È possibile ottenere lo stesso risultato ottenuto con l'esempio riportato nella sezione Redistributionsection:

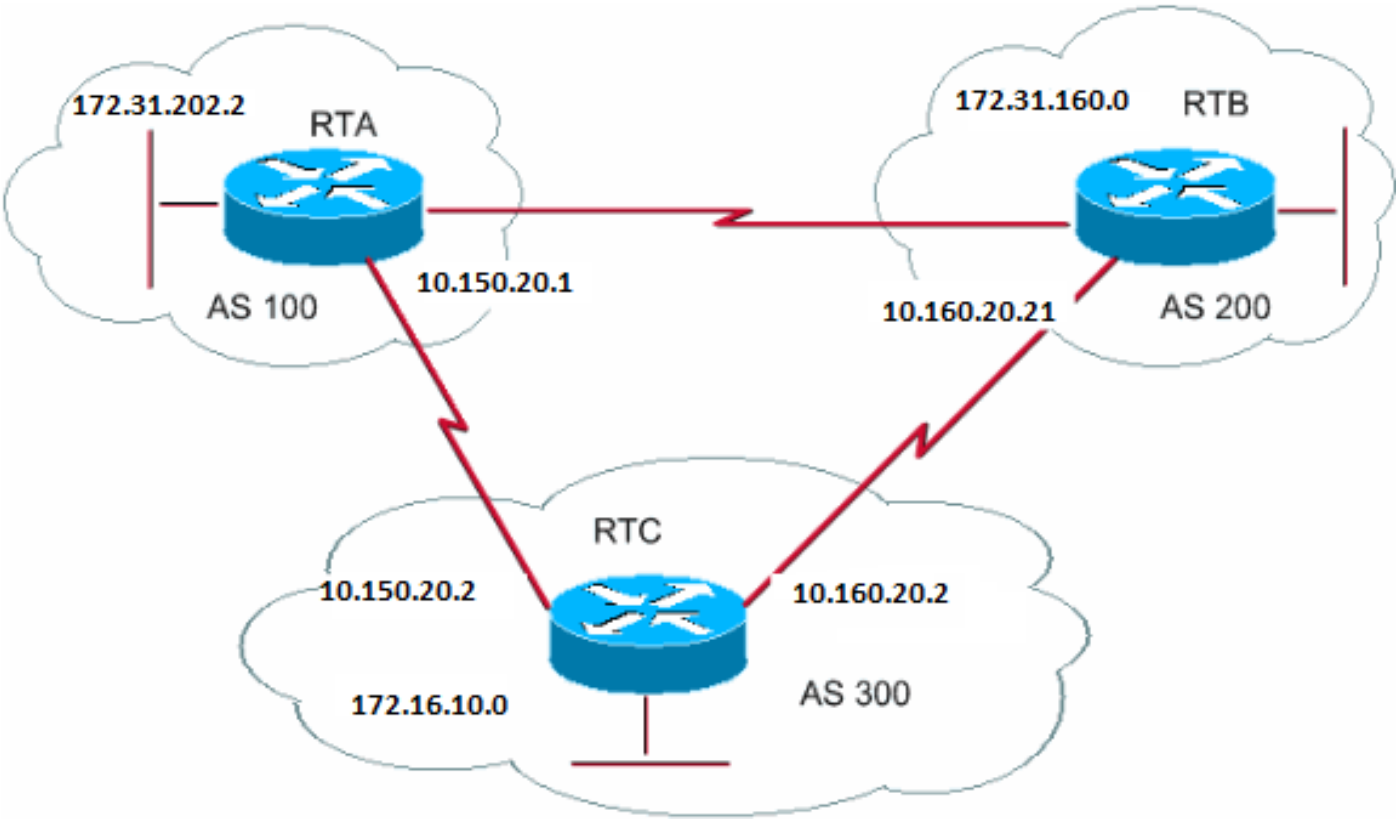
```
RTC#
router eigrp 10
 network 172.22.0.0
 redistribute bgp 200
 default-metric 1000 100 250 100 1500

router bgp 200
 neighbor 10.1.1.1 remote-as 300
 redistribute static

ip route 172.22.0.0 255.255.255.0 null0
```

L'null0 interfaccia significa che il pacchetto viene ignorato. Quindi, se si ottiene il pacchetto e esiste una corrispondenza più specifica di 172.22.0.0, il router invia il pacchetto alla corrispondenza specifica. In caso contrario, il router ignora il pacchetto. Questo metodo è utile per pubblicizzare una supernet.

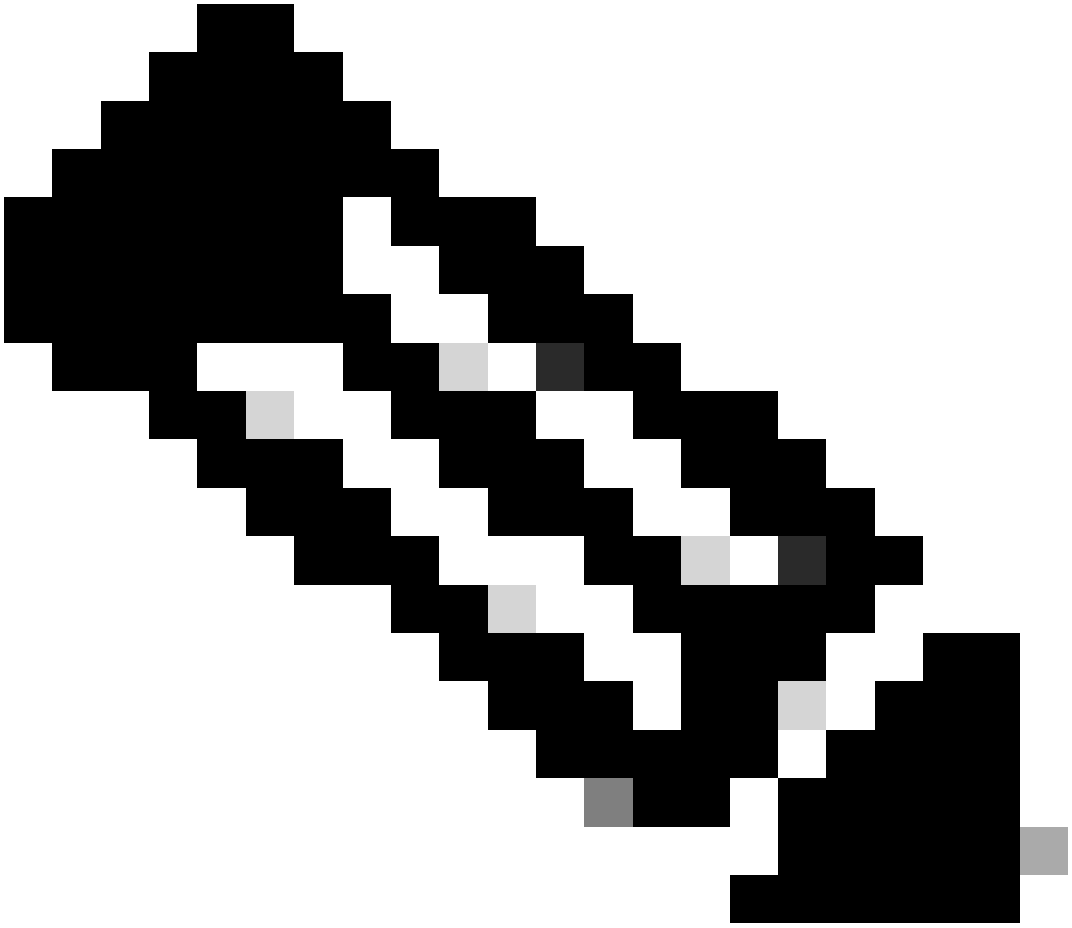
In questo documento abbiamo spiegato come utilizzare diversi metodi per originare le route dal proprio AS. Tenere presente che queste route vengono generate in aggiunta ad altre route BGP che BGP ha acquisito tramite i neighbor, sia interni che esterni. BGP passa le informazioni che BGP apprende da un peer ad altri peer. La differenza è che le route che generano dal network comando, dalla ridistribuzione o dallo stato statico indicano che il server AS è l'origine di queste reti.

La ridistribuzione è sempre il metodo per immettere BGP in IGP.

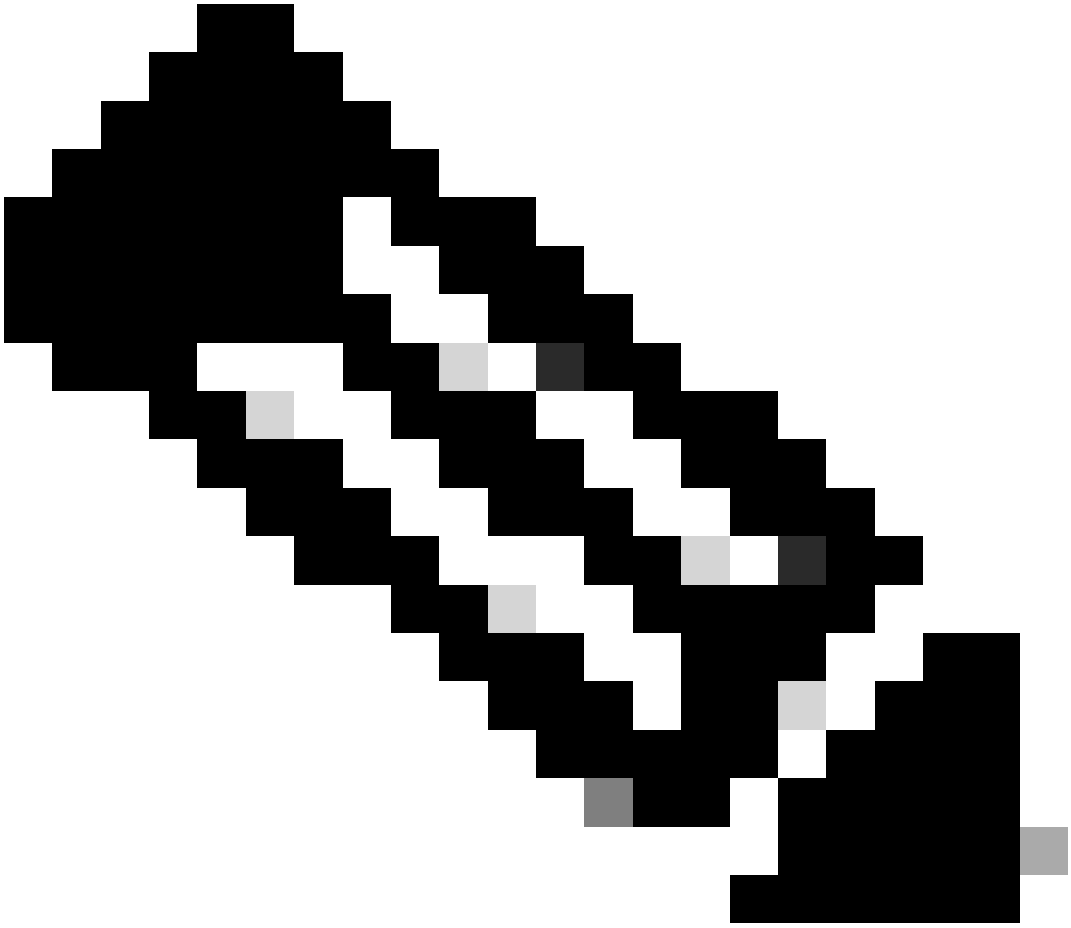
Di seguito è riportato un esempio:




```
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
```



Nota: la rete 172.31.202.2 o la rete 172.31.160.0 in RTC non sono necessarie a meno che non si desideri che RTC generi queste reti e le trasferisca sulle reti in arrivo da AS100 e AS200. Anche in questo caso, la differenza è che il comando network aggiunge un ulteriore annuncio per queste stesse reti, il che indica che AS300 è anche l'origine di queste route.



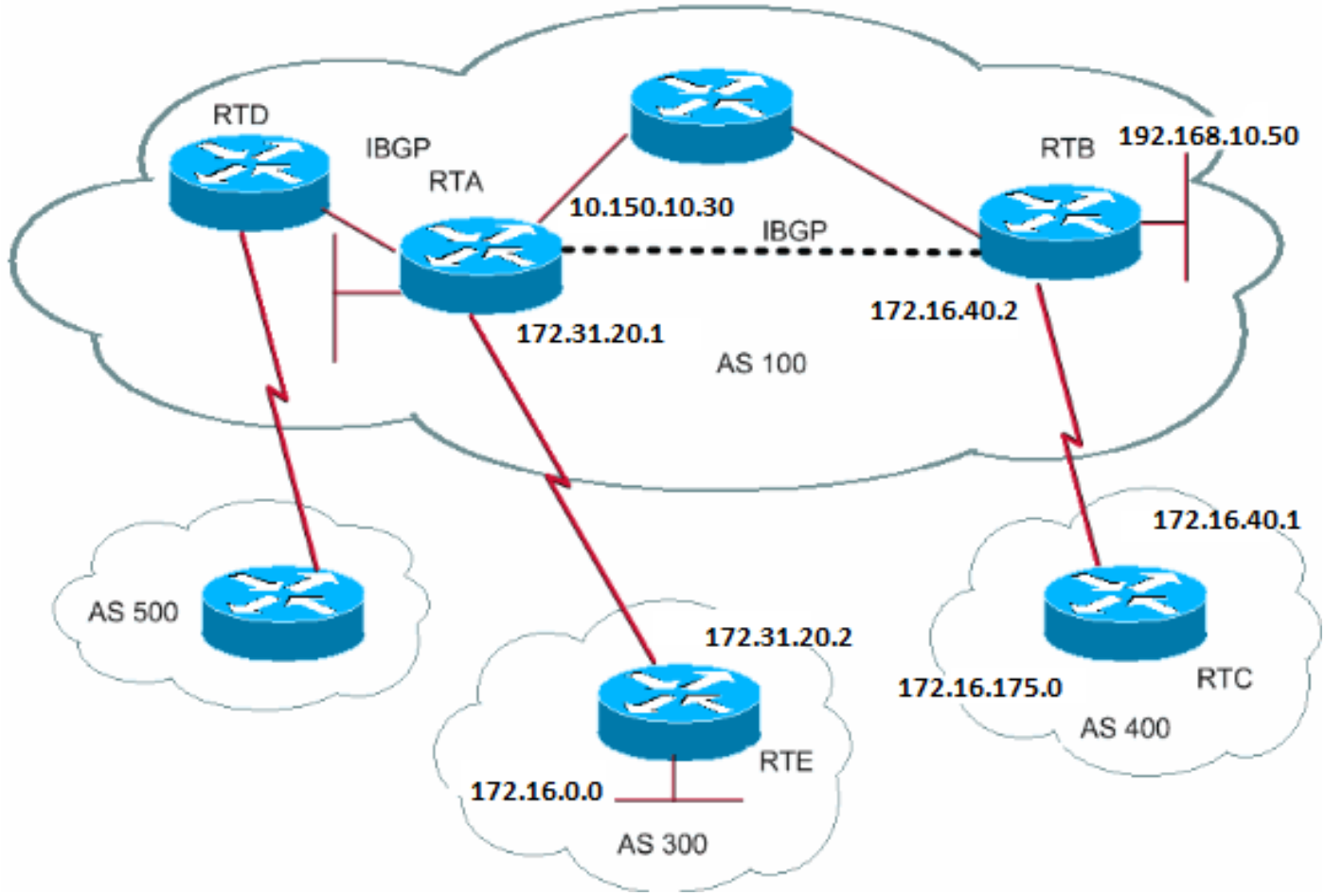
Nota: si ricordi che BGP non accetta gli aggiornamenti originati dal proprio AS. Questo rifiuto garantisce una topologia di interdominio priva di loop.

Si supponga, ad esempio, che AS200, come illustrato nell'esempio riportato in questa sezione, disponga di una connessione BGP diretta in AS100. RTA genera una route 172.31.202.2 e la invia a AS300. Quindi, RTC passa questo percorso a AS200 e mantiene l'origine come AS100. RTB passa 172.31.202.2 a AS100 con l'origine ancora AS100. RTA rileva che l'aggiornamento ha avuto origine dal proprio AS e ignora l'aggiornamento.

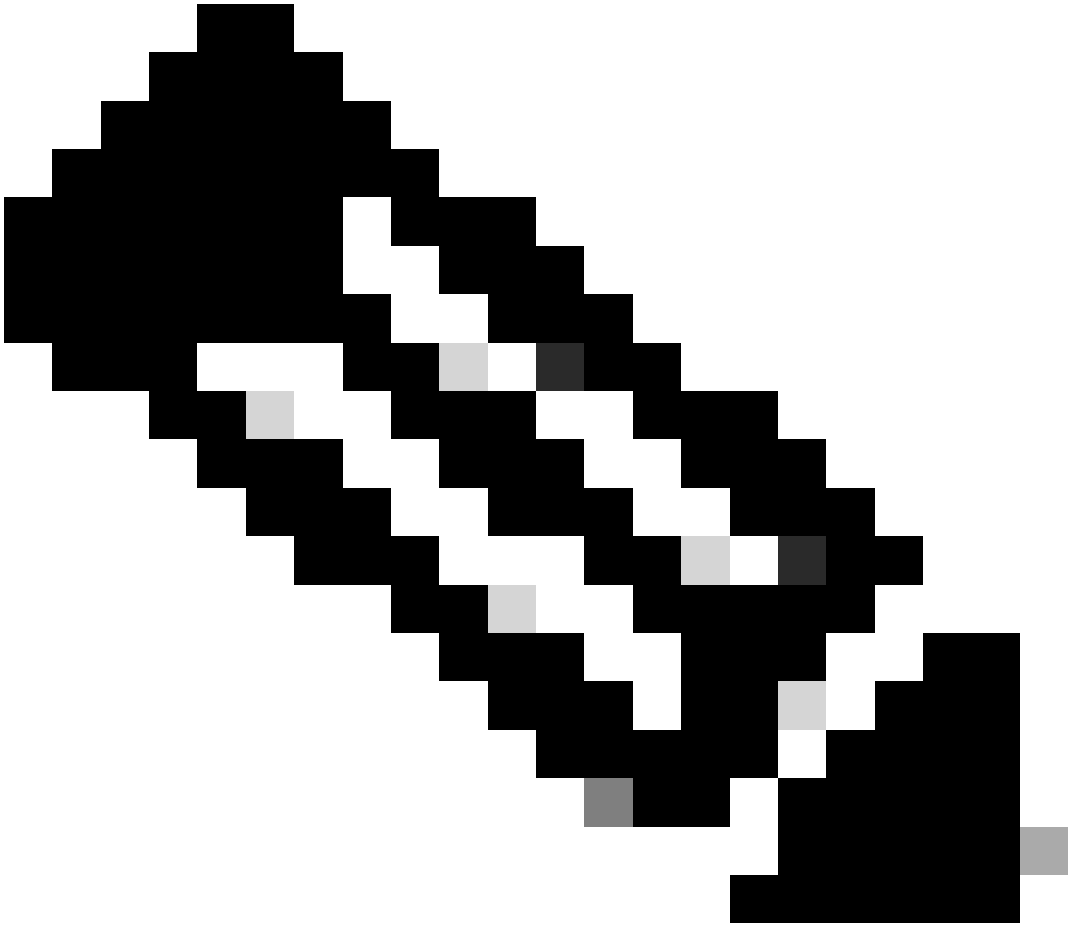
iBGP

Utilizzare iBGP se un AS desidera fungere da sistema di transito verso un altro AS. È possibile eseguire la stessa operazione se si apprende tramite eBGP, si ridistribuisce in IGP e quindi si ridistribuisce in un'altra AS. Ma iBGP offre una maggiore flessibilità e modalità più efficienti

per lo scambio di informazioni all'interno di una AS. Ad esempio, iBGP assicura modi per controllare il migliore punto di uscita fuori dall'AS con l'uso delle preferenze locali. L'attributo sectionLocal Preference fornisce ulteriori informazioni sulle preferenze locali.




```
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
```



Nota: si ricordi che quando uno speaker BGP riceve un aggiornamento da altri altoparlanti BGP nel proprio AS (iBGP), lo speaker BGP che riceve l'aggiornamento non ridistribuisce tali informazioni ad altri speaker BGP nel proprio AS. Lo speaker BGP che riceve l'aggiornamento ridistribuisce le informazioni ad altri speaker BGP al di fuori del proprio AS. Pertanto, occorre mantenere una mesh completa tra gli speaker iBGP all'interno di un AS.

RTA e RTB eseguono iBGP. RTA e RTD eseguono anche iBGP. Gli aggiornamenti BGP che provengono da RTB a RTA trasmettono a RTE, che si trova all'esterno dell'AS. Gli aggiornamenti non vengono trasmessi a RTD, che si trova all'interno dell'AS. Pertanto, effettuare un peering iBGP tra RTB e RTD per non interrompere il flusso degli aggiornamenti.

Algoritmo decisionale del BGP

Dopo che BGP riceve aggiornamenti su destinazioni diverse da sistemi autonomi diversi, il protocollo deve scegliere i percorsi per raggiungere

una destinazione specifica. BGP sceglie un solo percorso per raggiungere una destinazione specifica.

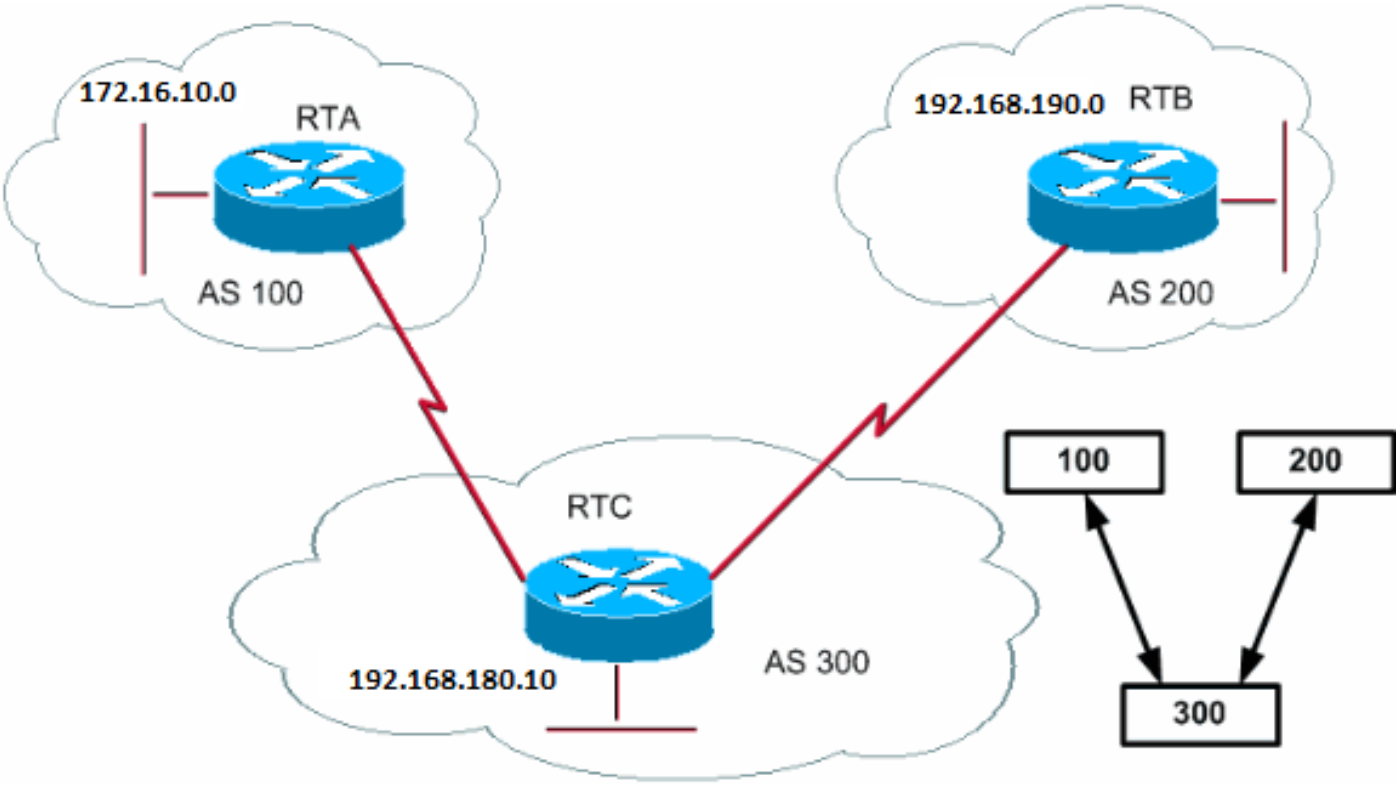
Il protocollo BGP basa la decisione su attributes diversi attributi, ad esempio hop successivo, pesi amministrativi, preferenza locale, origine della route, lunghezza del percorso, codice origine, metrica e altri.

Il BGP propaga sempre il percorso migliore verso i neighbor. Per ulteriori informazioni, fare riferimento all'algoritmo di selezione del miglior percorso BGP.

Nella sezione successiva vengono illustrati questi attributi e il loro utilizzo.

Case study BGP 2

Attributo AS_PATH



Ogni volta che un aggiornamento di route passa attraverso un AS, il numero AS viene anteposto a tale aggiornamento. L'attributo AS_PATH è in realtà l'elenco dei numeri AS attraversati da una route per raggiungere una destinazione. Un AS_SET è un insieme matematico ordinato {} di tutti gli AS attraversati. La sezione CIDR Example 2 (as-set) di questo documento fornisce un esempio di AS_SET.

Nell'esempio di questa sezione, RTB pubblicizza network 192.168.190.0 in AS200. Quando la route attraversa il percorso AS300, RTC aggiunge il proprio numero AS alla rete. Quando 192.168.190.0 raggiunge RTA, la rete ha due numeri AS collegati: prima 200, poi 300. Per RTA, il percorso per raggiungere 192.168.190.0 è (300, 200).

La stessa procedura si applica alle versioni 172.16.10.0 e 192.168.180.10. L'RTB deve seguire un percorso (300, 100); l'RTB attraversa l'AS300 e quindi l'AS100 per raggiungere la posizione 172.16.10.0. Per raggiungere 192.168.190.0 e 172.168.10.0, l'RTC deve attraversare il percorso (200).

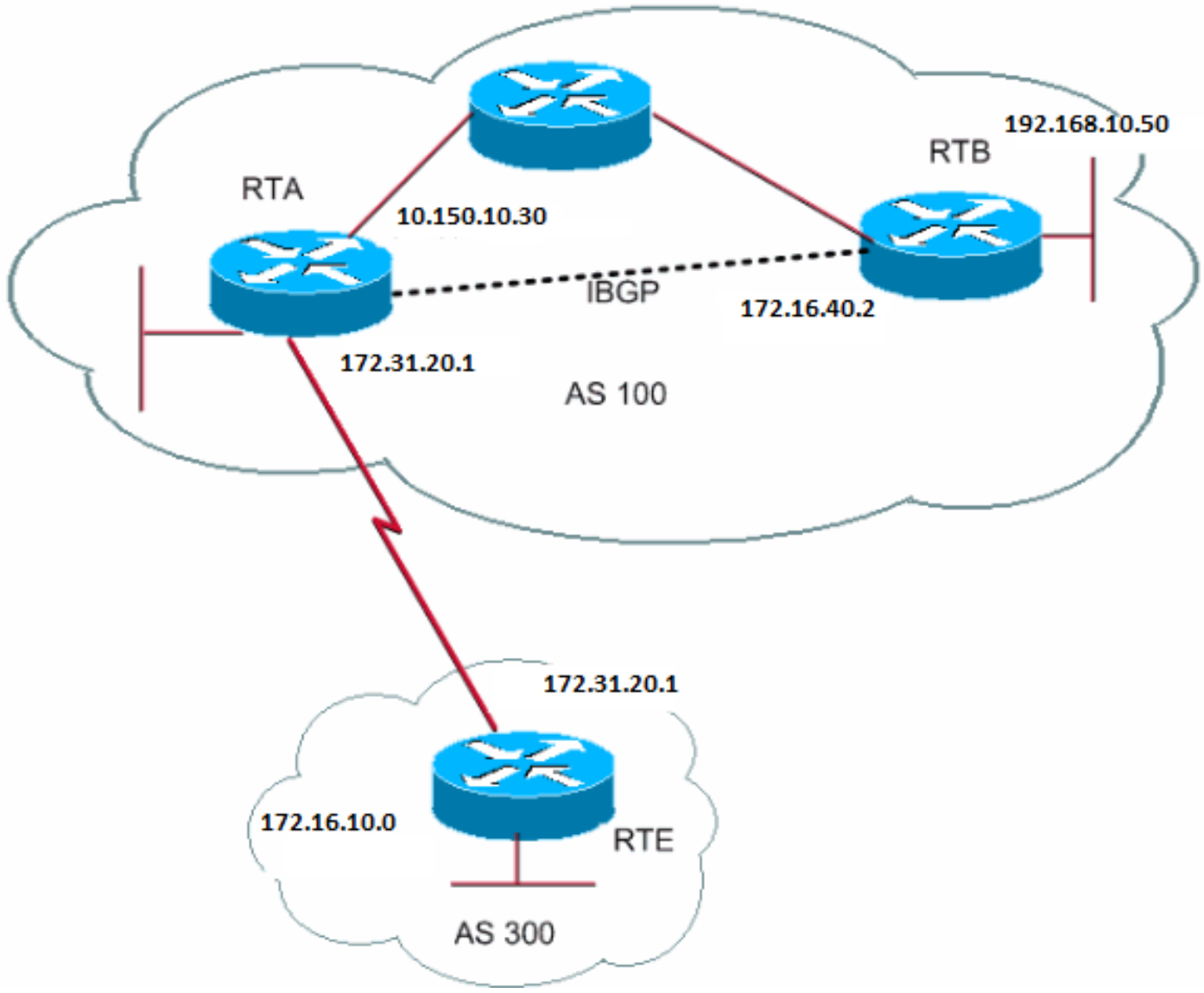
Attributo origine

L'origine è un attributo obbligatorio che definisce l'origine delle informazioni sul percorso. L'attributo origine può assumere tre valori:

- IGP: le NLRI (Network Layer Reachability Information) si trovano all'interno all'AS di origine. Ciò si verifica in genere quando si esegue il **bgp network**comando . L'aggiunta della tabella BGP indica IGP.

- EGP—NLRI è acquisito tramite EGP (exterior gateway protocol). Anein nella tabella BGP indica EGP.

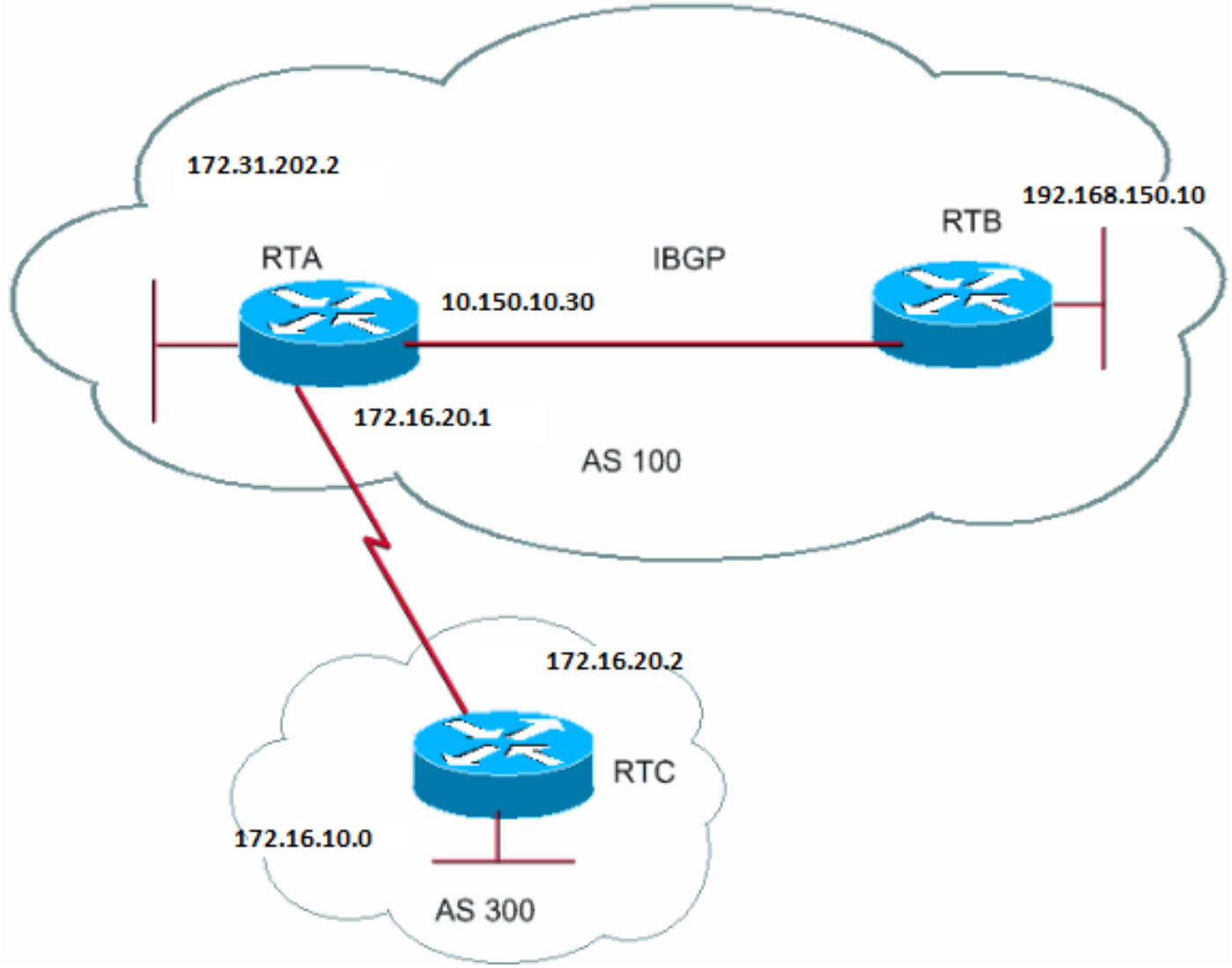
- INCOMPLETE: NLRI sconosciuto o acquisito con altri mezzi. INCOMPLETE di solito si ha quando si ridistribuiscono le route da altri protocolli di routing in BGP e l'origine della route è incompleta. Il punto interrogativo ?nella tabella BGP indica INCOMPLETE.



```
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 raggiunge 172.16.10.0 tramite 300 i. "300 i" indica che il percorso AS successivo è 300 e che l'origine della route è IGP. RTA raggiunge anche 192.168.10.150 tramite i. Questa "i" indica che la voce è nello stesso AS e che l'origine è IGP. RTE raggiunge 172.31.202.2 tramite 100 i. "100 i" significa che l'AS successivo è 100 e l'origine è IGP. RTE raggiunge anche 192.168.190.0 tramite 100?. Il valore "100 ?" indica che l'AS successivo è 100 e che l'origine è incompleta e proviene da un percorso statico.

Attributo next-hop BGP



Attributo next-hop BGP

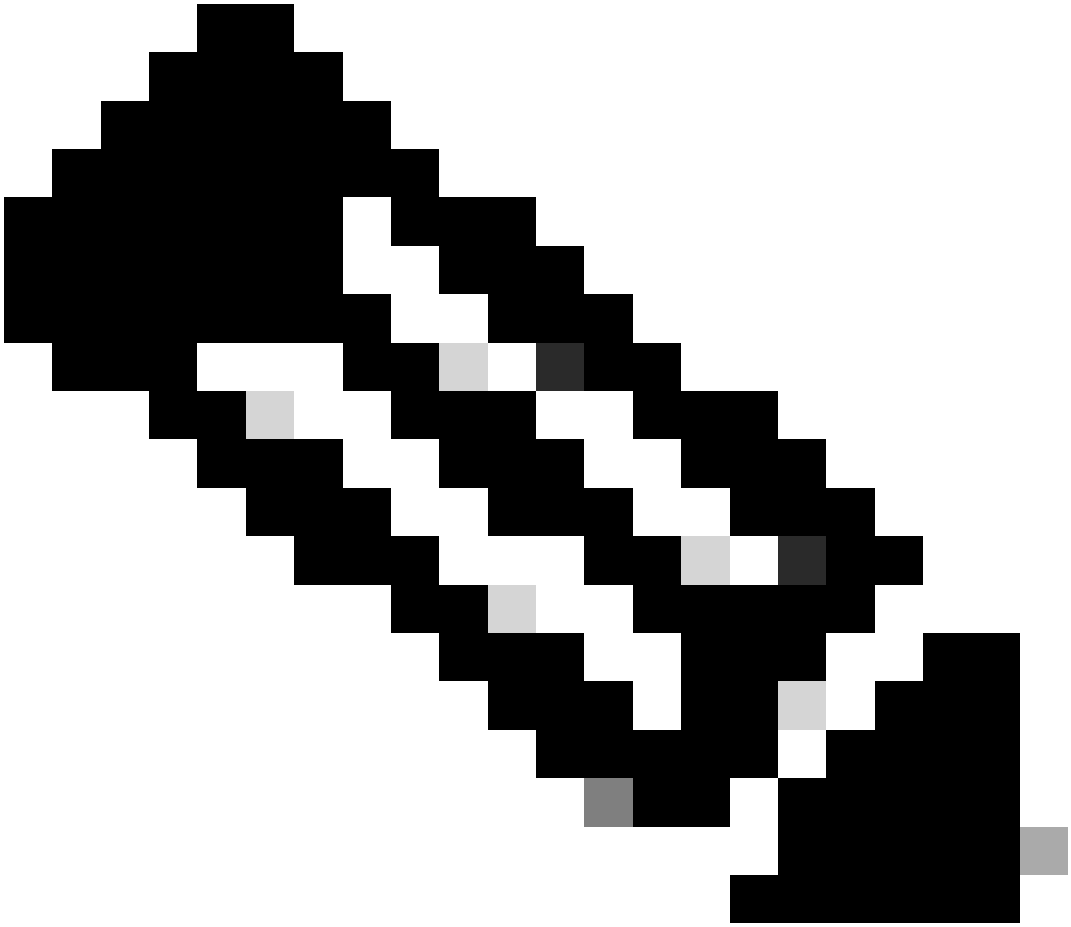
L'attributo next-hop BGP è l'indirizzo IP del next-hop da utilizzare per raggiungere una determinata destinazione.

Per eBGP, l'hop successivo è sempre l'indirizzo IP della risorsa adiacente specificata dal neighbor comando. Nell'esempio di questa sezione, RTC pubblicizza 172.16.10.0 su RTA con un hop successivo di 172.31.20.2. RTA annuncia 172.31.202.2 a RTC con un hop successivo di 172.31.20.1. Per iBGP, il protocollo stabilisce che l'hop successivo annunciato da eBGP deve essere trasferito in iBGP. A causa di questa regola, RTA annuncia 172.16.10.0 al suo iBGP peer RTB con un hop successivo di 172.31.20.2. In base all'RTB, l'hop successivo per raggiungere il valore 172.16.10.0 è 172.31.20.2 *e non* 10.150.10.30.

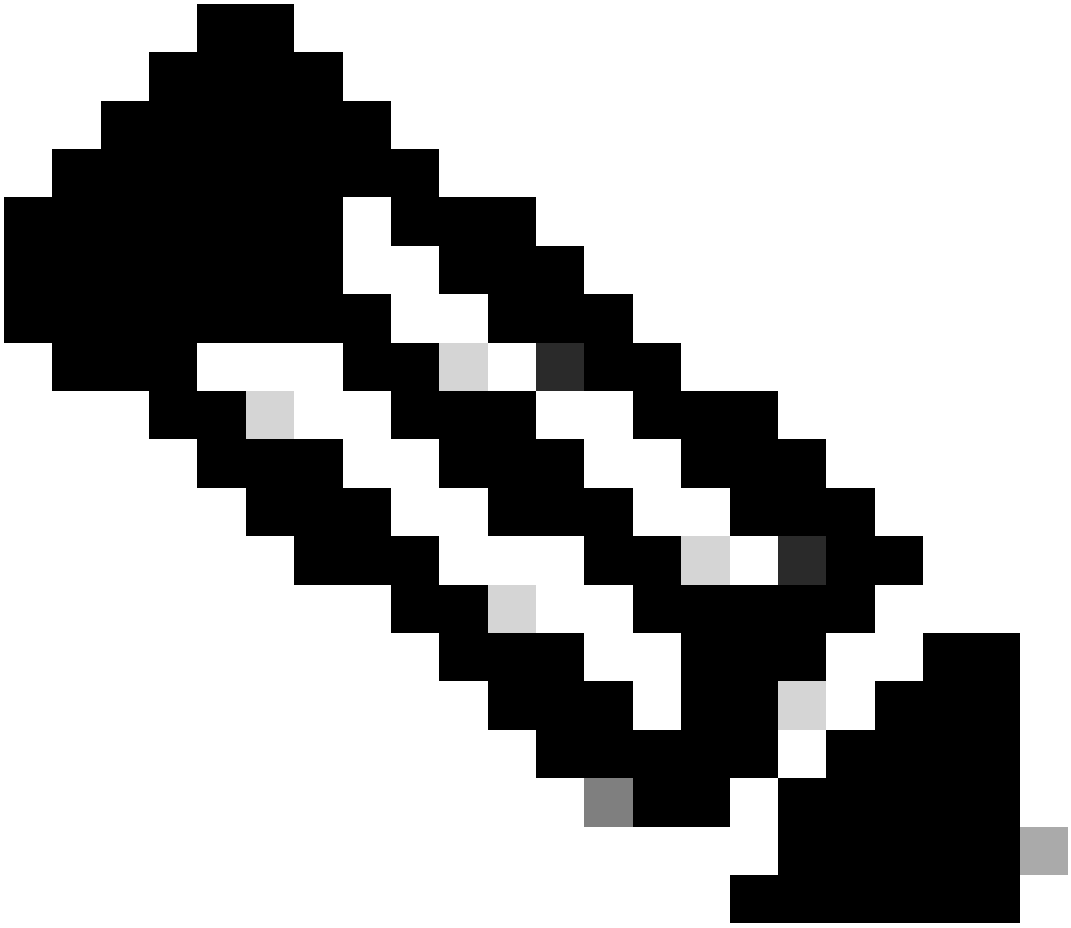
Assicurarsi che RTB possa raggiungere 172.31.20.2 tramite IGP. In caso contrario, RTB elimina i pacchetti con la destinazione 172.16.10.0 perché l'indirizzo del next-hop non è accessibile. Ad esempio, se RTB esegue iGRP, è possibile eseguire anche iGRP su RTA network 172.16.10.0. Si desidera rendere l'iGRP passivo sul collegamento a RTC in modo che BGP venga scambiato solo.

```
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
```



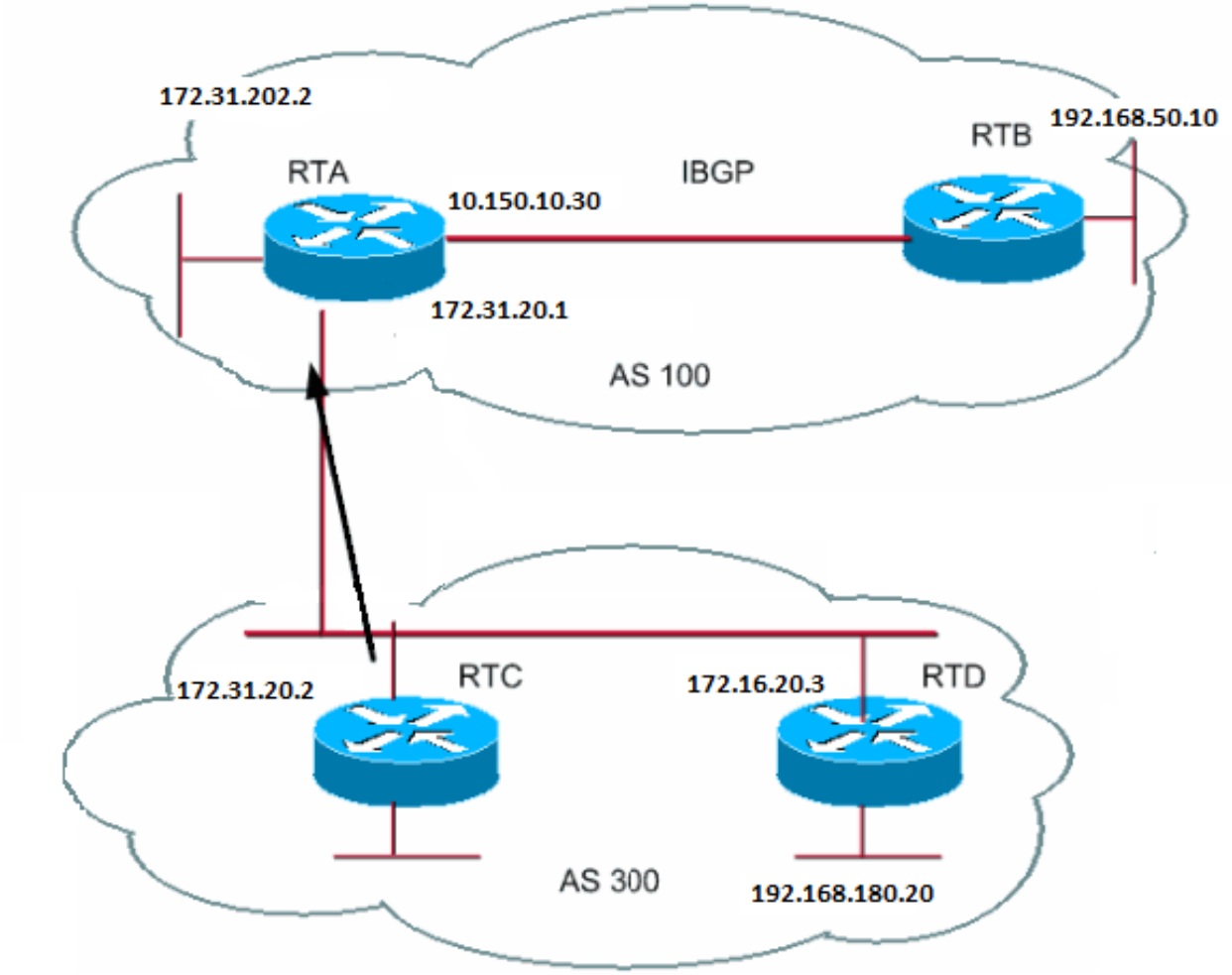
Nota: RTC pubblicizza 172.16.10.0 su RTA con un next-hop di 172.31.20.2.



Nota: RTA annuncia 172.16.10.0 a RTB con un hop successivo pari a 172.31.20.2. L'hop successivo eBGP viene trasferito in iBGP.

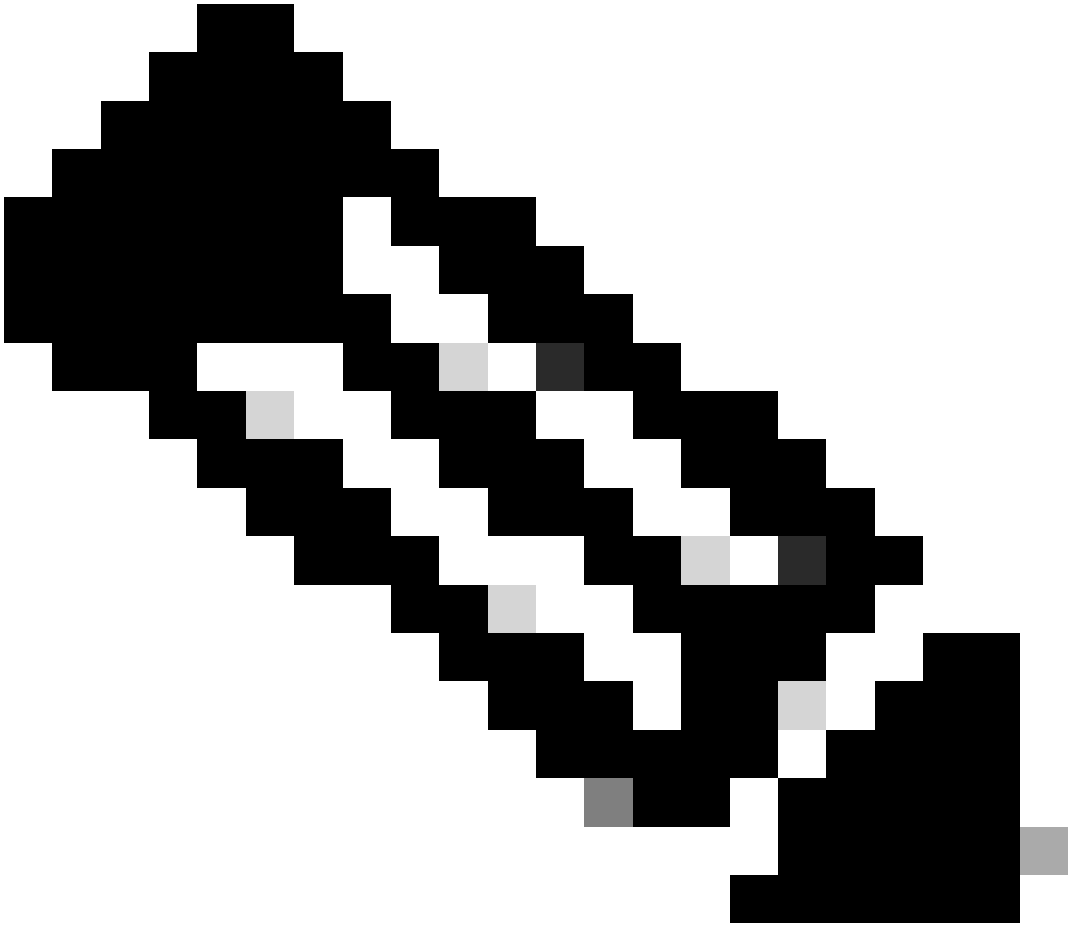
Presta particolare attenzione quando si hanno a che fare con reti multiaccesso e reti multiaccesso (NBMA) non broadcast. Per ulteriori informazioni, fare riferimento alle sezioni BGP Next Hop (Multiaccess Networks) e BGP Next Hop (NBMA).

Next-hop BGP (reti ad accessi multipli)



Questo esempio mostra il comportamento del comando next-hop su una rete ad accessi multipli come Ethernet.

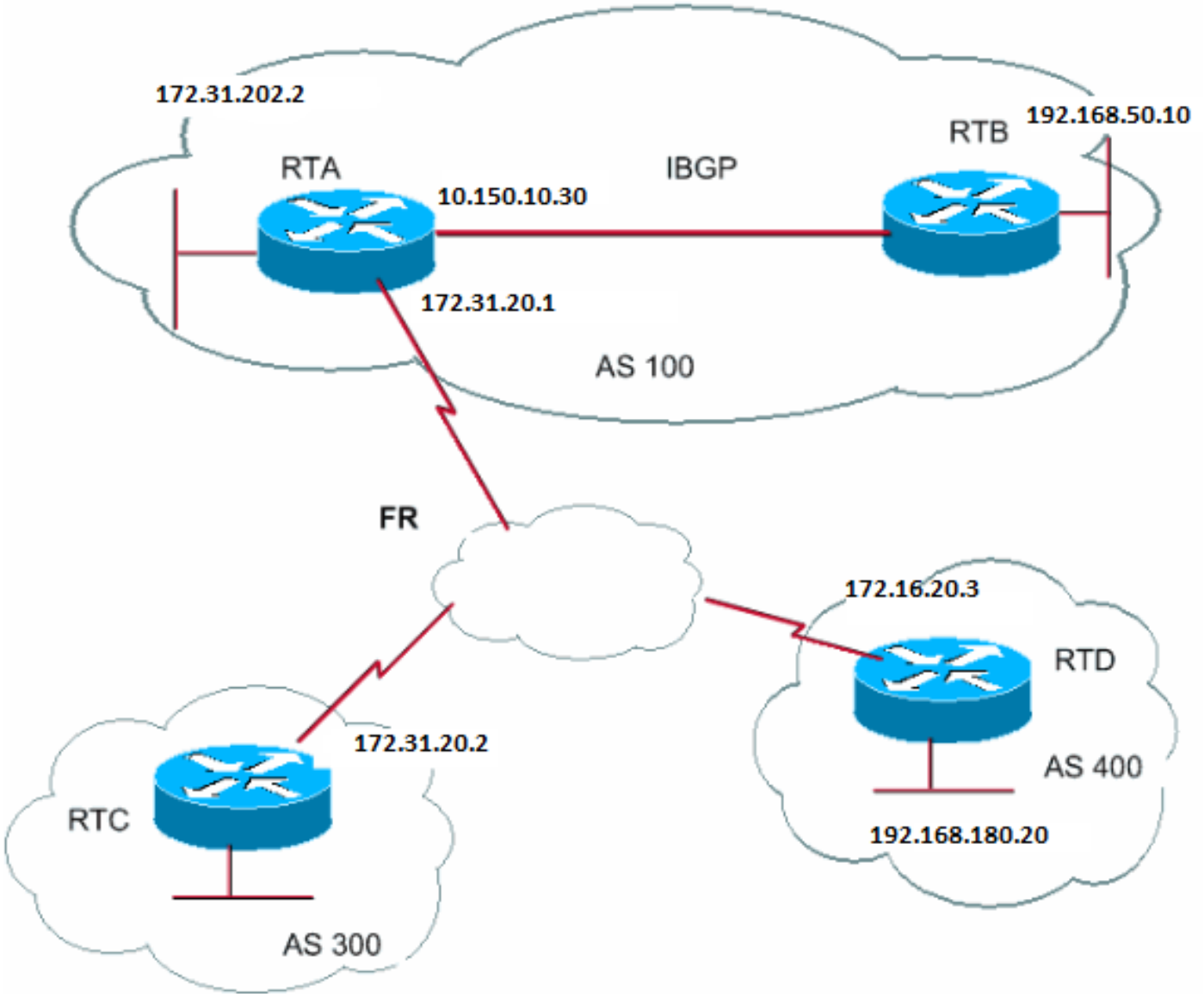
Si supponga che RTC e RTD in AS300 eseguano OSPF. RTC esegue BGP con RTA. L'RTC può raggiungere la rete 192.168.180.20 tramite la rete 172.16.20.3. Quando RTC invia un aggiornamento BGP a RTA per quanto riguarda 192.168.180.20, l'hop successivo è 172.16.20.3. RTC non usa il proprio indirizzo IP, 172.31.20.2. RTC utilizza questo indirizzo perché la rete tra RTA, RTC e RTD è una rete ad accesso multiplo. L'uso RTA di RTD come next-hop per raggiungere 192.168.180.20 è più sensato rispetto all'hop extra tramite RTC.



Nota: RTC pubblicizza 192.168.180.20 su RTA con next-hop 172.16.20.3.

Se il supporto comune per RTA, RTC e RTD non ha accessi multipli ma è di tipo NBMA, si verificano ulteriori complicazioni.

Next-hop BGP (NBMA)



Il supporto comune viene visualizzato come una nuvola nel diagramma. Se il supporto comune è un frame relay o qualsiasi cloud NBMA, è come se si avesse una connessione tramite Ethernet. RTC pubblicizza 192.168.180.20 su RTA con un next-hop di 172.16.20.3.

Il problema è che RTA non ha un circuito virtuale permanente (PVC) per RTD e non può raggiungere il next-hop. In questo caso, il routing non riesce.

Il next-hop-selfcomando risolve il problema.

Comando next-hop-self

Nelle situazioni con l'hop successivo, come nell'esempio di BGP Next Hop (NBMA), è possibile usare il next-hop-self comando. La sintassi è:

<#root>

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

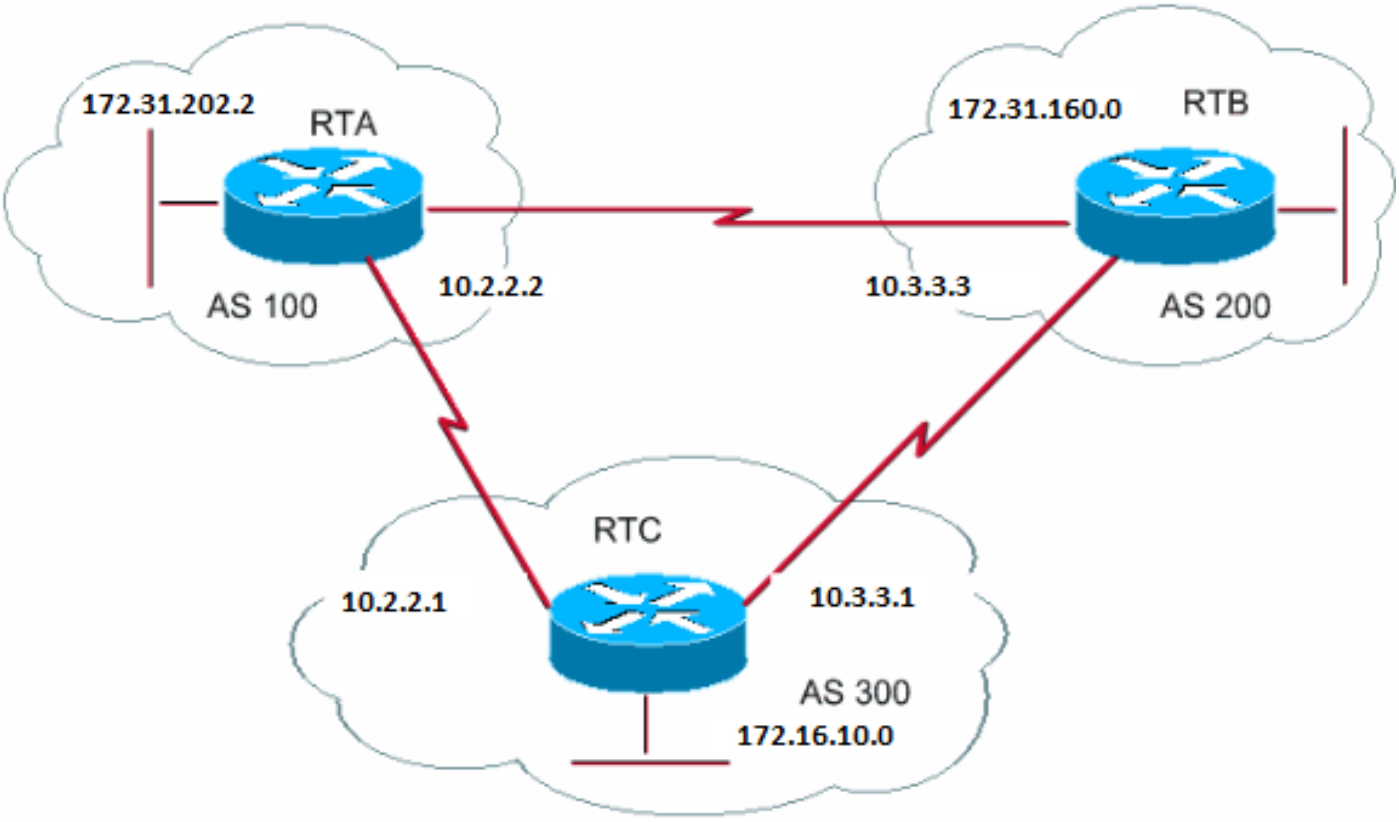
Il next-hop-selfcomando consente di forzare BGP a utilizzare un indirizzo IP specifico come hop successivo.

Per l'esempio riportato in Next-hop BGP (NBMA), questa configurazione risolve il problema:

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

RTC pubblicizza 192.168.180.20 con un next-hop di 172.31.20.2.

Backdoor BGP



Nel diagramma precedente, RTA e RTC eseguono eBGP. RTB e RTC eseguono eBGP. RTA e RTB eseguono una sorta di IGP, RIP, IGRP o un altro protocollo. Per definizione, gli aggiornamenti eBGP hanno una distanza di 20, che è inferiore alle distanze IGP. Le distanze predefinite

sono:

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

RTA riceve gli aggiornamenti su 172.31.160.0 tramite due protocolli di routing:

- eBGP con una distanza di 20
- IGP con una distanza maggiore di 20

Per impostazione predefinita, BGP ha le seguenti distanze:

- Distanza esterna: 20
- Distanza interna: 200
-

Distanza locale: 200

Tuttavia, è possibile utilizzare il distance comando per modificare le distanze predefinite:

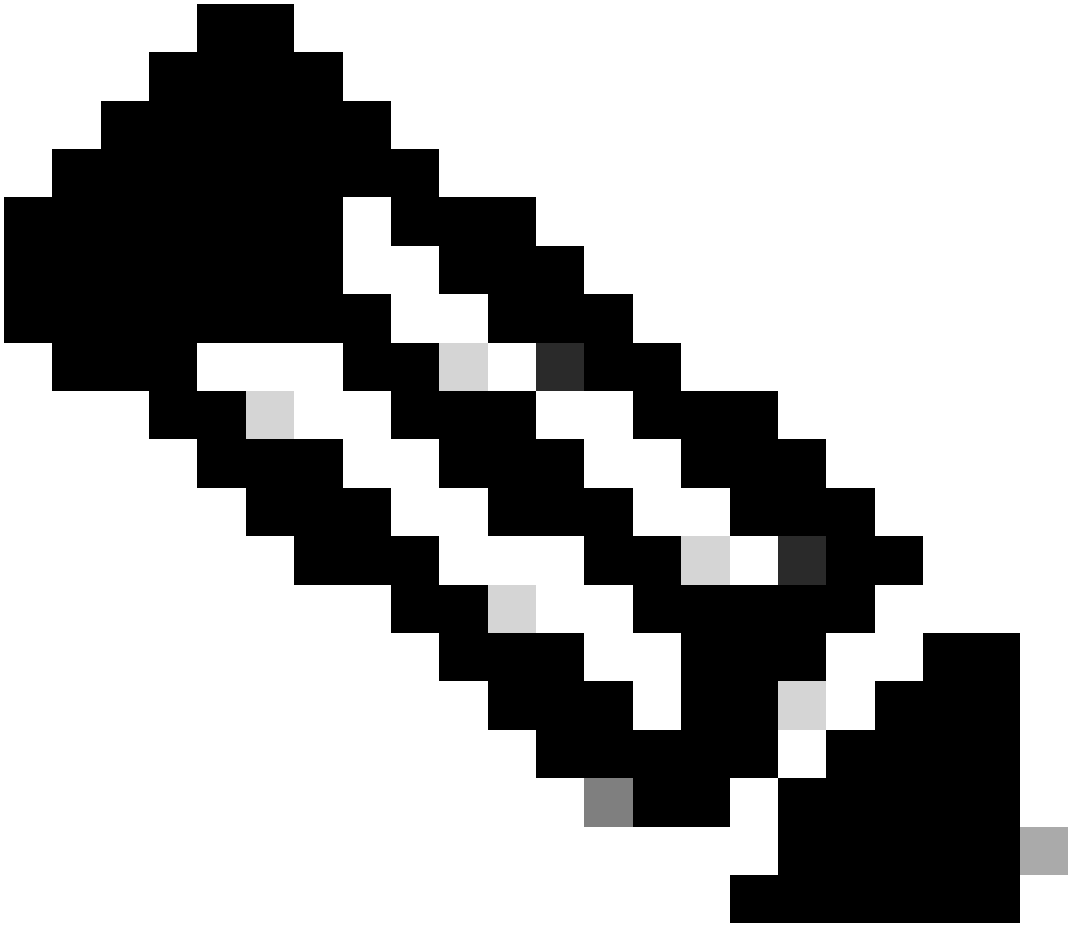
<#root>

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

RTA sceglie eBGP tramite RTC a causa della minore distanza.

Se si desidera che RTA acquisisca informazioni su 172.31.160.0 tramite RTB (IGP), sono disponibili due opzioni:

- Modificare la distanza esterna di eBGP o IGP.



Nota: questa modifica non è consigliata.

- Utilizzo di BGP backdoor.

BGP backdoor rende IGP la route preferita.

Eseguire il comando **networkaddressbackdoor**.

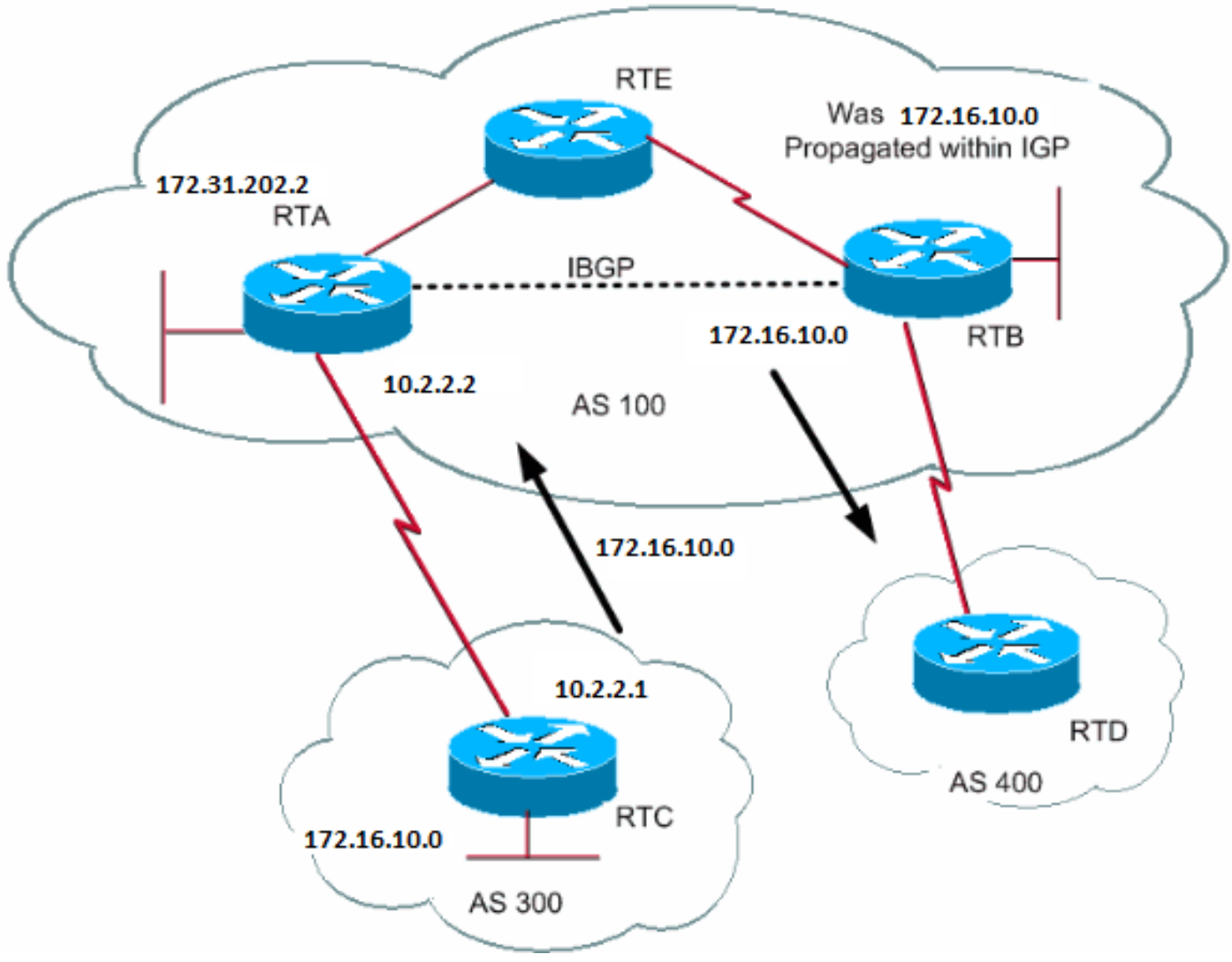
La rete configurata è la rete che si desidera raggiungere tramite IGP. Per il BGP, questa rete riceve lo stesso trattamento di una rete assegnata localmente, ad eccezione degli aggiornamenti BGP che non la pubblicizzano.

```
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
```

La rete 172.31.160.0 viene considerata come una voce locale, ma non viene pubblicizzata come una normale voce di rete.

RTA apprende 172.31.160.0 da RTB tramite EIGRP con distanza 90. RTA impara anche l'indirizzo da RTC via eBGP con distanza 20. Normalmente eBGP è la preferenza, ma a causa del comando **network backdoor**, la preferenza è EIGRP.

Sincronizzazione



Prima di analizzare la sincronizzazione, osservare questo scenario. RTC in AS300 invia aggiornamenti su 172.16.10.0. RTA e RTB eseguono

iBGP, quindi RTB ottiene l'aggiornamento ed è in grado di raggiungere la versione 172.16.10.0 tramite l'hop successivo 10.2.2.1. Tenere presente che l'hop successivo viene eseguito tramite iBGP. Per raggiungere il next-hop, RTB deve inviare il traffico a RTE.

Si supponga che RTA non abbia ridistribuito la rete 172.16.10.0 in IGP. A questo punto, RTE non ha idea dell'esistenza di 172.16.10.0.

Se RTB inizia a pubblicizzare su AS400 un valore che può raggiungere 172.16.10.0, il traffico proveniente da RTD a RTB con destinazione 172.16.10.0 arriva e scende a RTE.

La sincronizzazione stabilisce che, se il SA trasmette il traffico da un altro SA a un terzo AS, BGP non deve annunciare una route prima che tutti i router del SA siano a conoscenza della route tramite IGP. Il BGP attende fino a quando l'IGP non ha propagato la route all'interno dell'AS. Quindi, il BGP pubblicizza la route ai peer esterni.

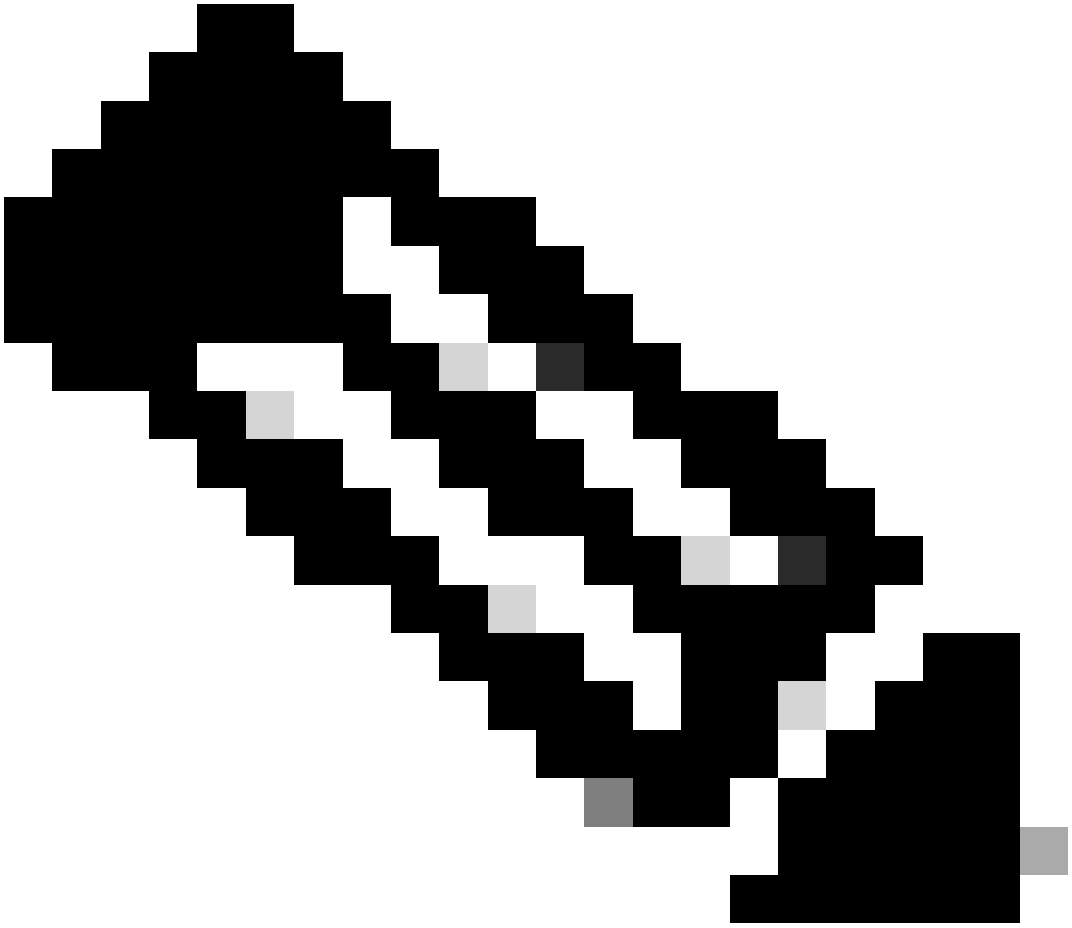
Nell'esempio riportato in questa sezione, RTB attende informazioni su 172.16.10.0 tramite IGP. Quindi, RTB inizia a inviare l'aggiornamento a RTD. Se si aggiunge una route statica in RTB che punta a 172.16.10.0, RTB può ritenere che le informazioni siano state propagate da IGP. Verificare che gli altri router possano raggiungere la versione 172.16.10.0.

Sincronizzazione disabilitata

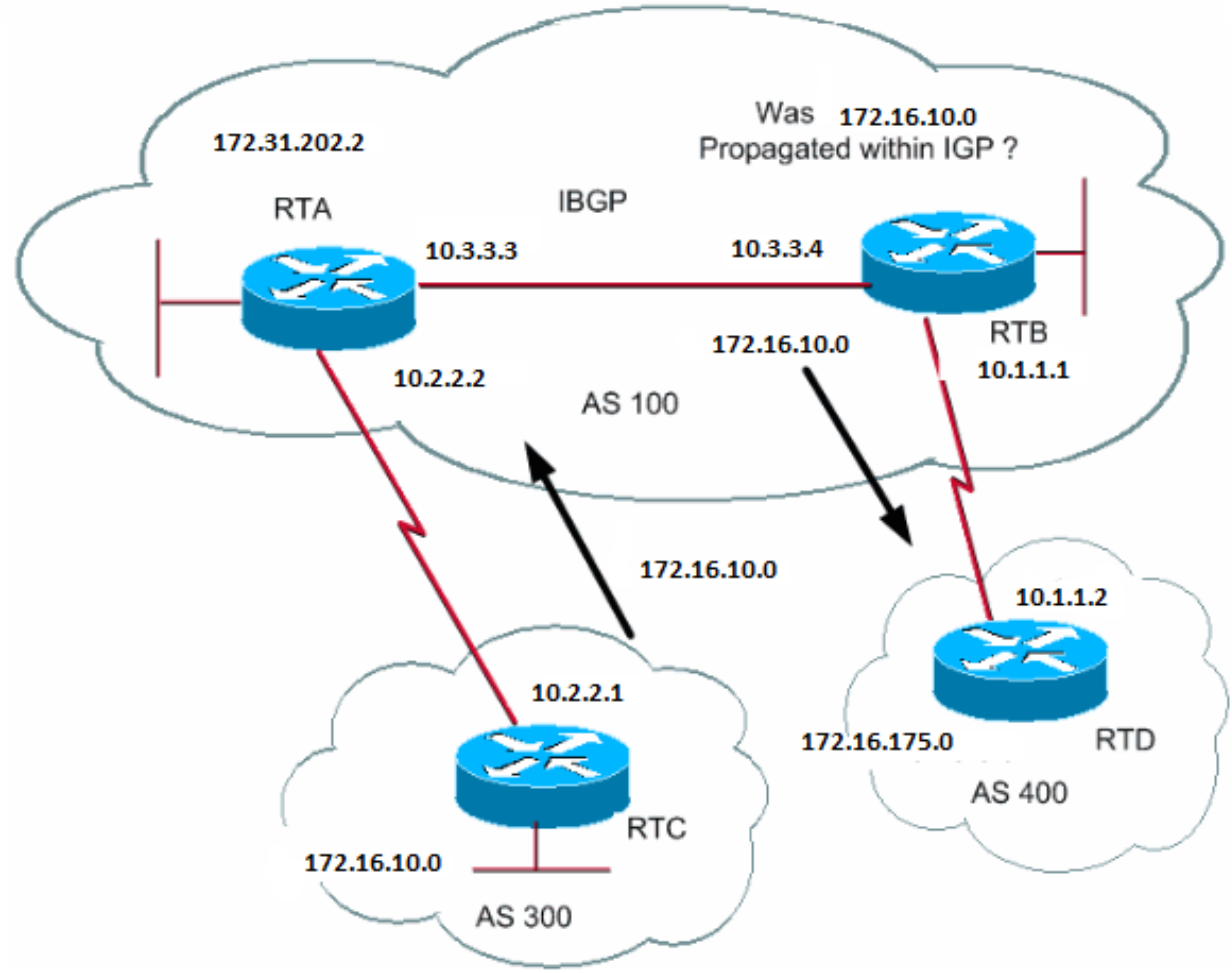
In alcuni casi, non è necessaria la sincronizzazione. Se non si trasferisce il traffico da un AS diverso attraverso il proprio AS, è possibile disabilitare la sincronizzazione. È inoltre possibile disabilitare la sincronizzazione se tutti i router dell'AS eseguono BGP. La disabilitazione di questa funzione consente di trasportare meno route in IGP e consentire a BGP di convergere più rapidamente.

La disabilitazione della sincronizzazione non è automatica. Se tutti i router nell'AS eseguono BGP e non si esegue affatto IGP, il router non ha modo di saperlo. Il router attende indefinitamente un aggiornamento IGP su una determinata route prima che il router invii la route ai peer esterni. In questo caso è necessario disabilitare manualmente la sincronizzazione in modo che il routing funzioni correttamente:

```
router bgp 100
 no synchronization
```

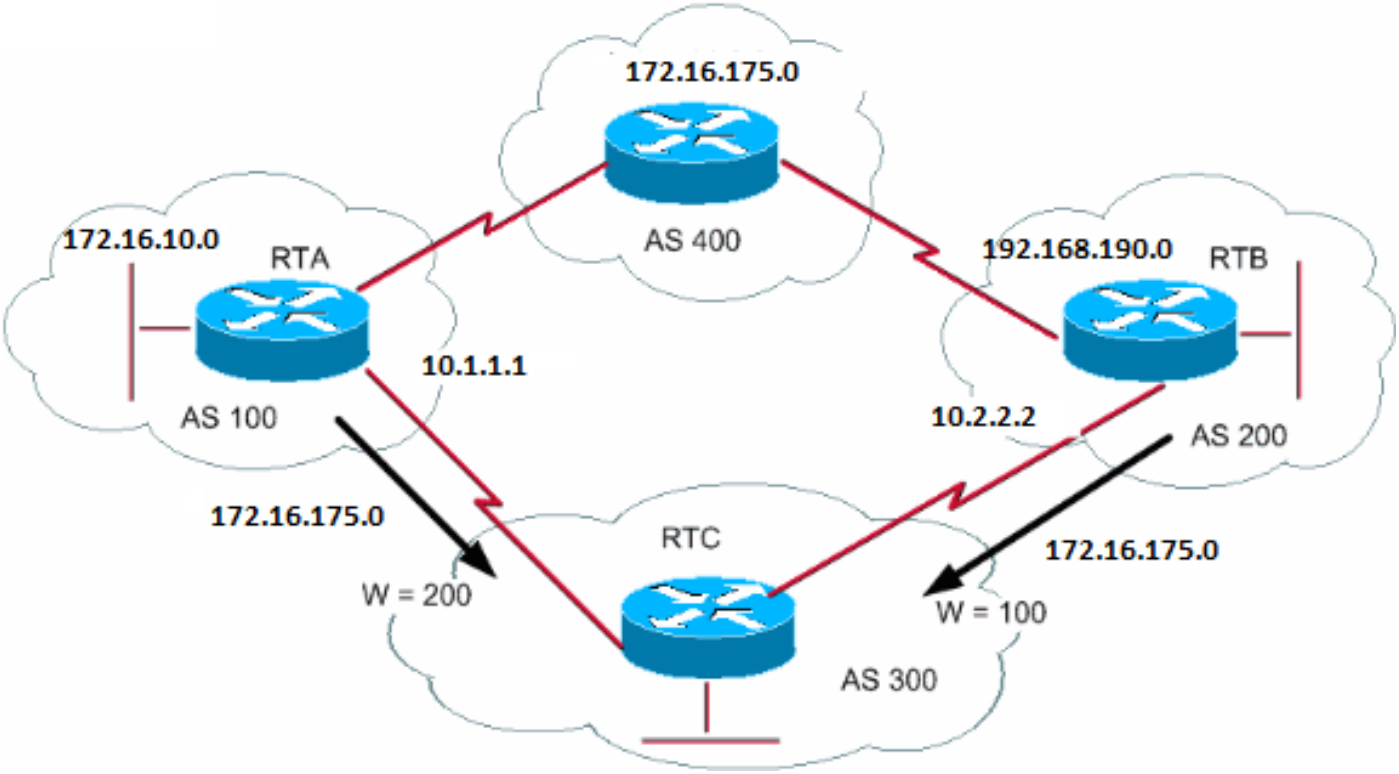


Nota: accertarsi di usare il comando clear ip bgp address per ripristinare la sessione.




```
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
```

Attributo peso



L'attributo peso è un attributo definito da Cisco. Questo attributo utilizza la ponderazione per selezionare il percorso migliore. Il peso viene assegnato localmente al router. Il valore ha senso solo per il router specifico. Il valore non viene propagato o trasmesso tramite nessuno degli aggiornamenti della route. Un peso può essere un numero compreso tra 0 e 65.535. Per impostazione predefinita, i percorsi originati dal router hanno un peso di 32.768 e gli altri percorsi hanno un peso di 0.

Le route con un peso più elevato hanno la preferenza quando esistono più route verso la stessa destinazione. Osservare l'esempio in questa sezione. RTA ha appreso informazioni sulla rete 172.16.0.0 da AS4. RTA propaga l'aggiornamento a RTC. RTB ha anche appreso la conoscenza della rete 172.16.0.0 da AS4. RTB propaga l'aggiornamento a RTC. RTC ora ha due modi per raggiungere 172.16.0.0 e deve decidere quale strada seguire. Se si imposta il peso degli aggiornamenti per RTC provenienti da RTA in modo che sia maggiore del peso degli aggiornamenti provenienti da RTB, RTC verrà forzato a utilizzare RTA come hop successivo per raggiungere il valore 172.16.0.0. Più metodi raggiungono questo peso impostato:

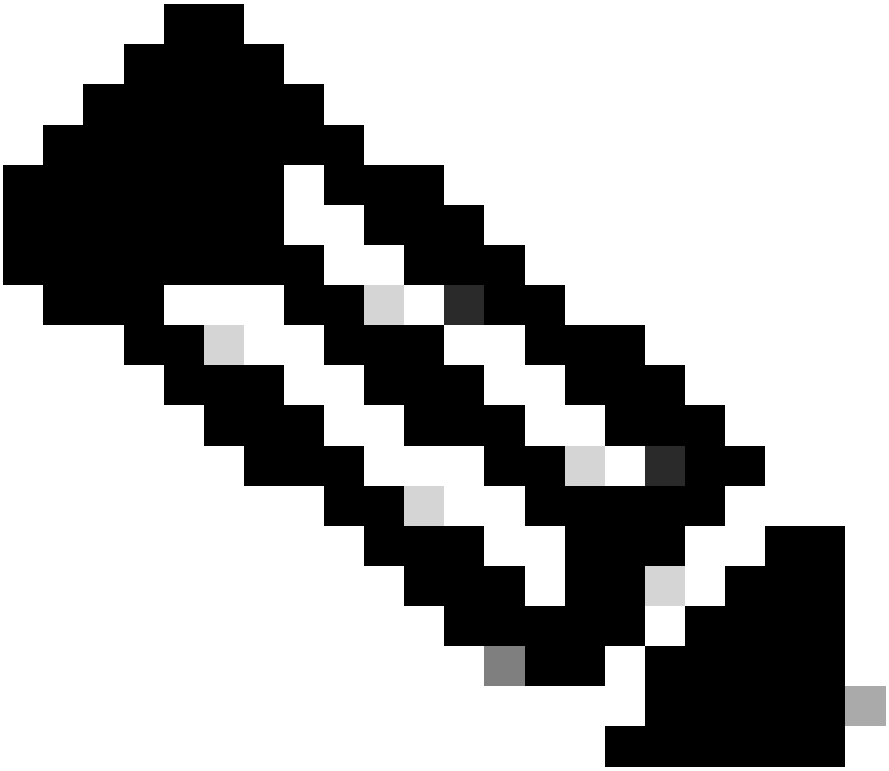
- Utilizzare il comando neighbor.

  vicino {ip-address|peer-group} peso <weight>

- Utilizzare gli elenchi di accesso AS_PATH.

- **ip as-path access-list <access-list-number>{allow | deny} <espressione-regolare-as>**

- **router adiacente <indirizzo-ip>filter-list <access-list-number>peso <weight>**



Nota: in alcuni scenari possono essere presenti pochi comandi non disponibili in alcune versioni del software.

- Utilizzare le route map.

```
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, che ha un peso superiore, è il next-hop preferito.

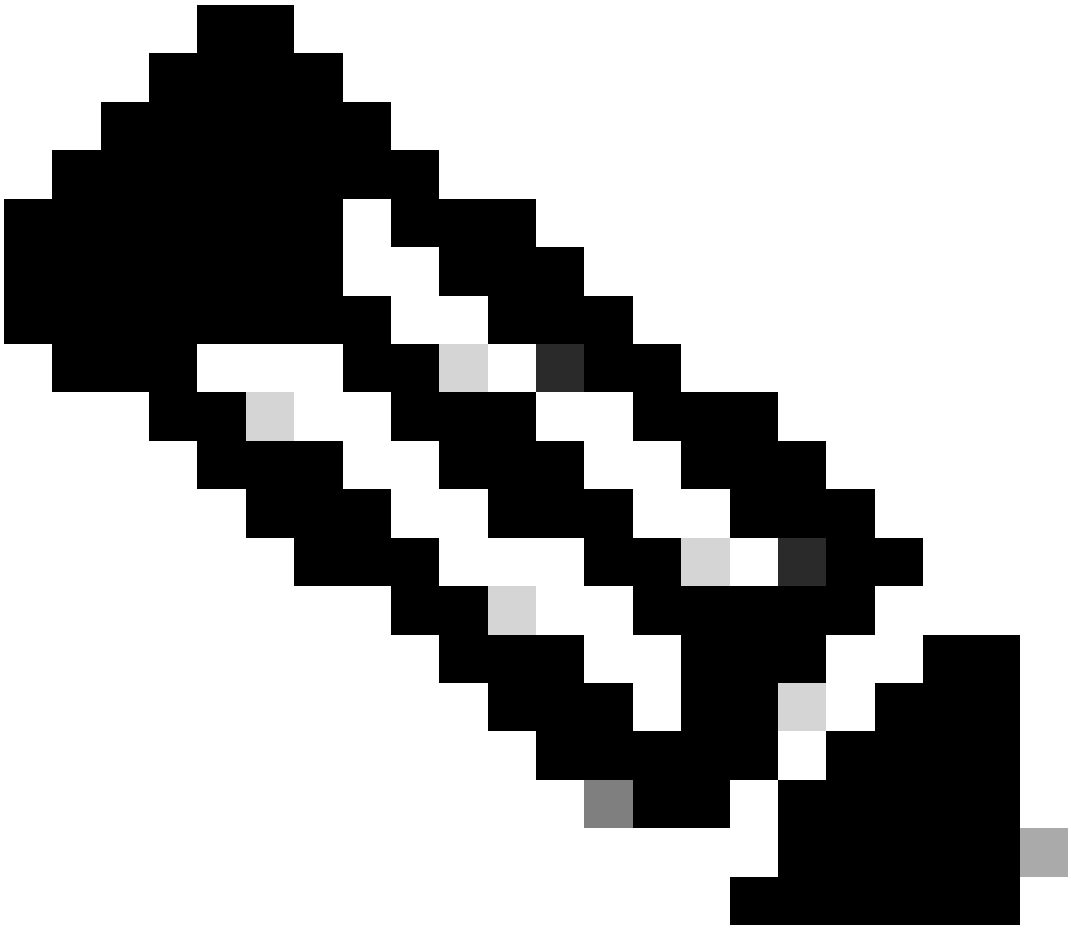
È possibile ottenere lo stesso risultato con IP AS_PATH e gli elenchi dei filtri.

```
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$
...
```

È inoltre possibile ottenere lo stesso risultato utilizzando le route map.

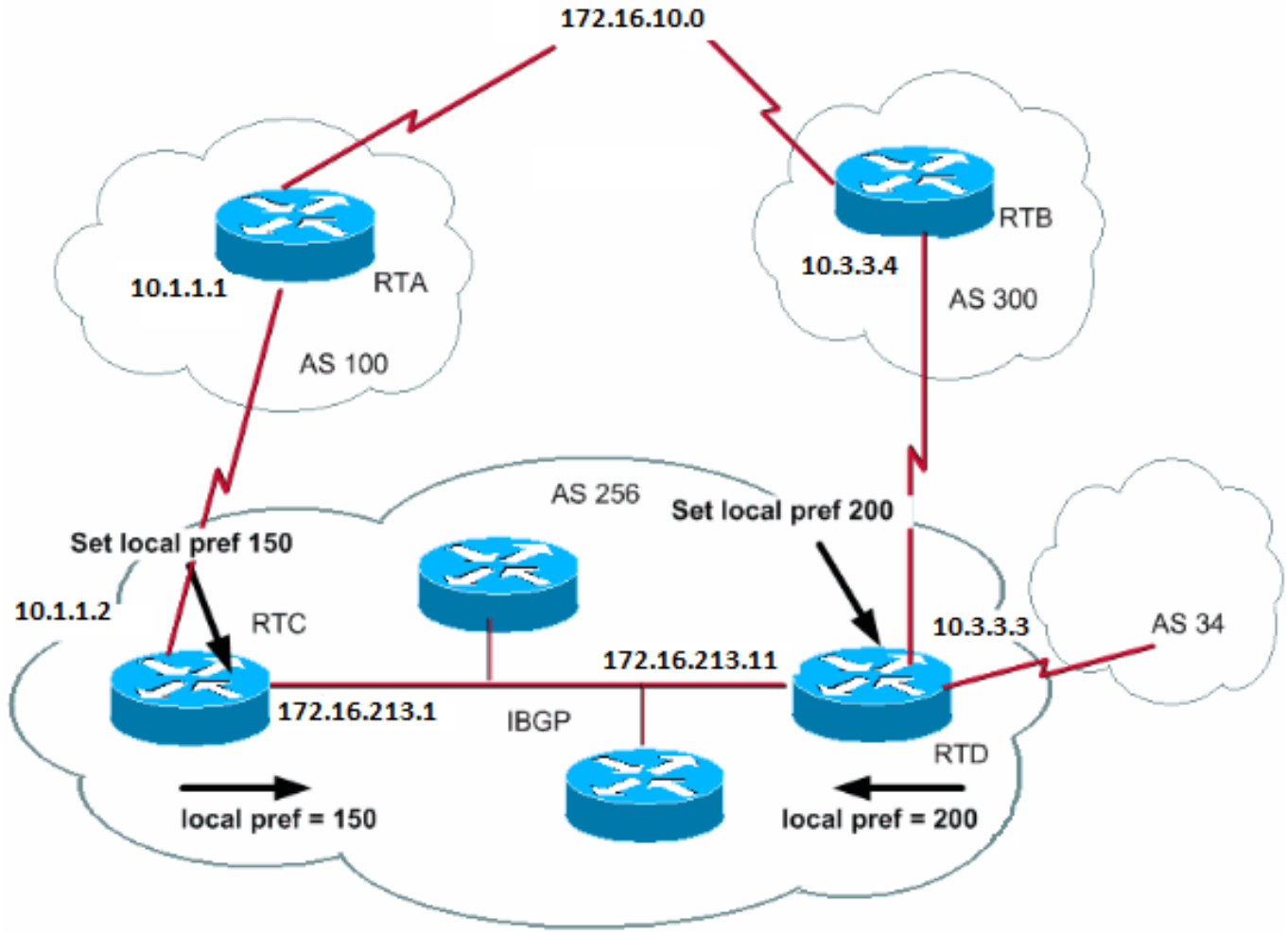
```
RTC#
router bgp 300
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 route-map setweightin in
 neighbor 10.2.2.2 remote-as 200
 neighbor 10.2.2.2 route-map setweightin in
...
ip as-path access-list 5 permit ^100$
```

```
...

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.
```



Nota: è possibile modificare il peso per preferire il percorso MPLS VPN BGP con il percorso IGP come backup.

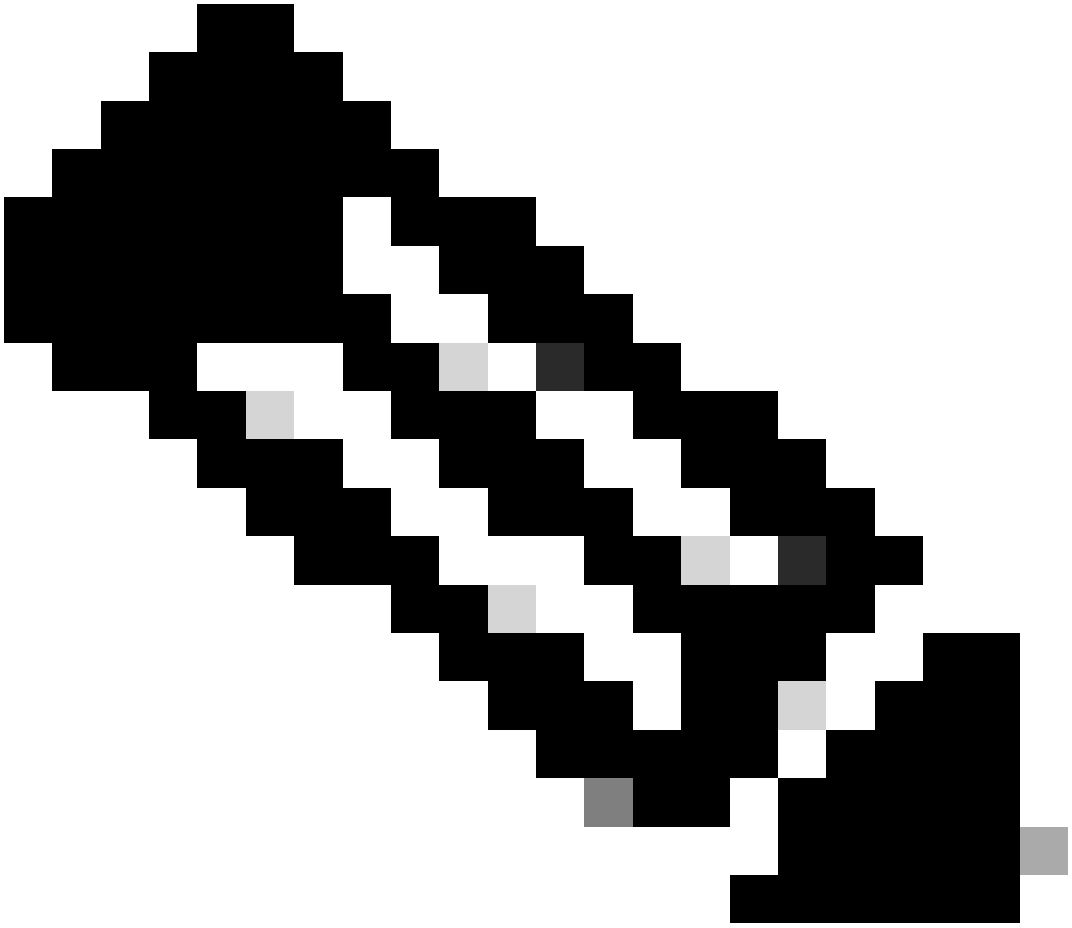
Attributo preferenza locale



La preferenza locale è un'indicazione all'AS del percorso preferibile per uscire dall'AS e raggiungere una determinata rete. Un percorso con preferenza locale superiore è maggiormente preferibile. Il valore predefinito della preferenza locale è 100.

A differenza dell'attributo peso, che è rilevante solo per il router locale, la preferenza locale è un attributo scambiato dai router nello stesso AS.

La preferenza locale si imposta con il comando bgp default local-preference value . È inoltre possibile impostare le preferenze locali con le route map, come illustrato nell'esempio riportato in questa sezione:



Nota: per poter prendere in considerazione le modifiche, è necessario eseguire un soft reset (ossia cancellare il processo bgp sul router). Per cancellare il processo bgp, usare il comando dove softclear ip bgp [soft][in/out] indica un soft reset e non interrompe la sessione e [in/out] specifica la configurazione in entrata o in uscita. Se in/out non è specificato, le sessioni in entrata e in uscita vengono reimpostate.

Il comando bgp default local-preference imposta la preferenza locale sugli aggiornamenti del router che vanno ai peer nello stesso AS. Nel diagramma di questa sezione, AS256 riceve gli aggiornamenti su 172.16.10.0 da due diversi lati dell'organizzazione. La preferenza locale consente di determinare il modo in cui uscire da AS256 per raggiungere la rete. Si supponga che RTD sia la preferenza del punto di uscita. Questa configurazione imposta la preferenza locale per gli aggiornamenti da AS300 a 200 e per gli aggiornamenti da AS100 a 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
```

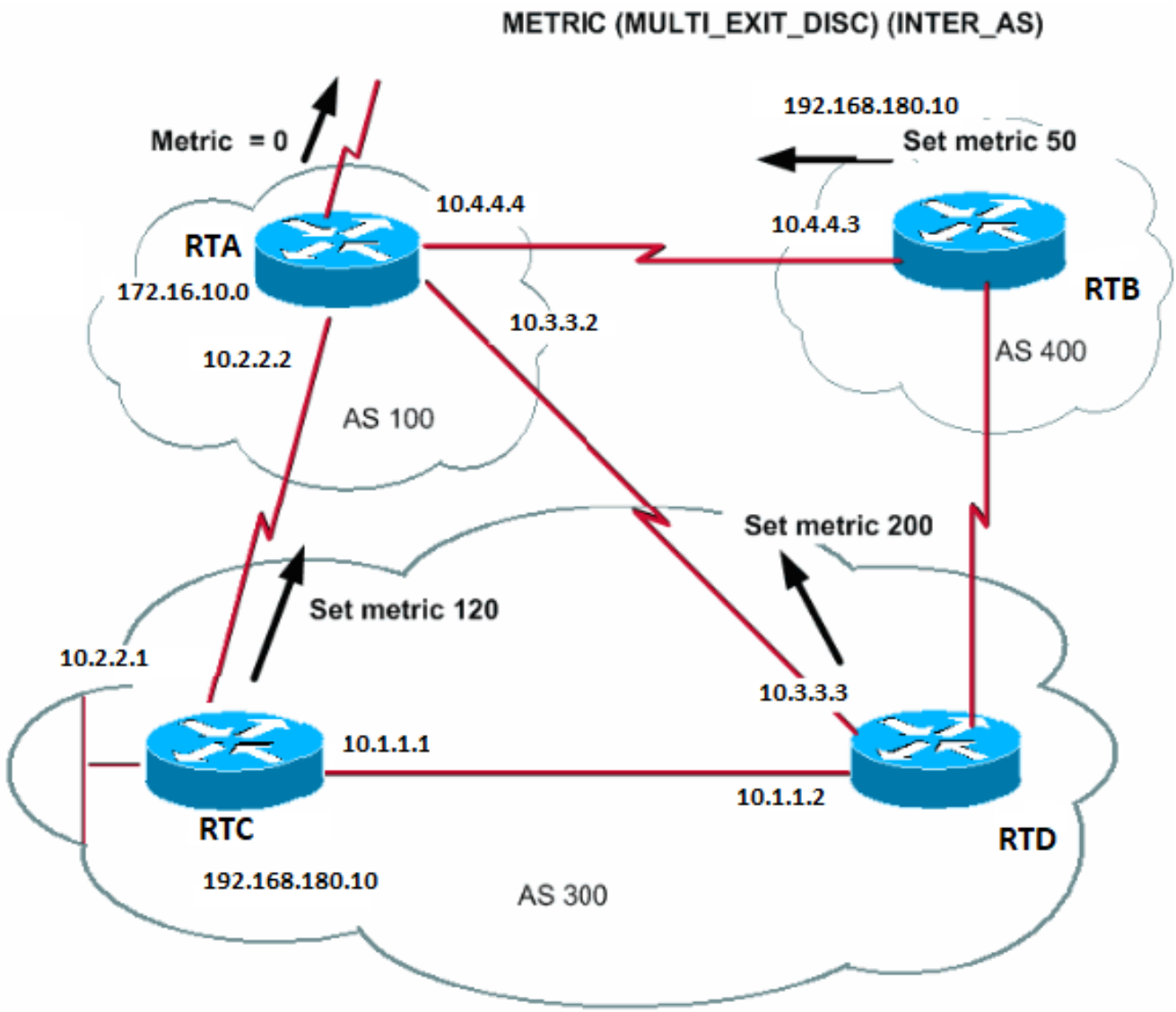
In questa configurazione, RTC imposta la preferenza locale di tutti gli aggiornamenti su 150. La stessa RTD imposta la preferenza locale di tutti gli aggiornamenti su 200. All'interno di AS256 si verifica uno scambio di preferenze locali. Pertanto, sia RTC che RTD si rendono conto che la rete 172.16.10.0 ha una preferenza locale maggiore quando gli aggiornamenti provengono da AS300 anziché da AS100. Tutto il traffico in AS256 che ha quella rete come destinazione trasmette con RTD come punto di uscita.

L'uso di route map offre maggiore flessibilità. Nell'esempio in questa sezione, tutti gli aggiornamenti ricevuti da RTD vengono contrassegnati con la preferenza locale 200 quando gli aggiornamenti raggiungono RTD. Anche gli aggiornamenti che provengono da AS34 sono contrassegnati con la preferenza locale di 200. Questo tag può non essere necessario. Per questo motivo, è possibile utilizzare le route map per specificare gli aggiornamenti specifici che devono essere contrassegnati con una preferenza locale specifica. Di seguito è riportato un esempio:

```
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
```

Con questa configurazione, qualsiasi aggiornamento proveniente da AS300 ha una preferenza locale di 200. Tutti gli altri aggiornamenti, ad esempio quelli provenienti da AS34, hanno un valore di 150.

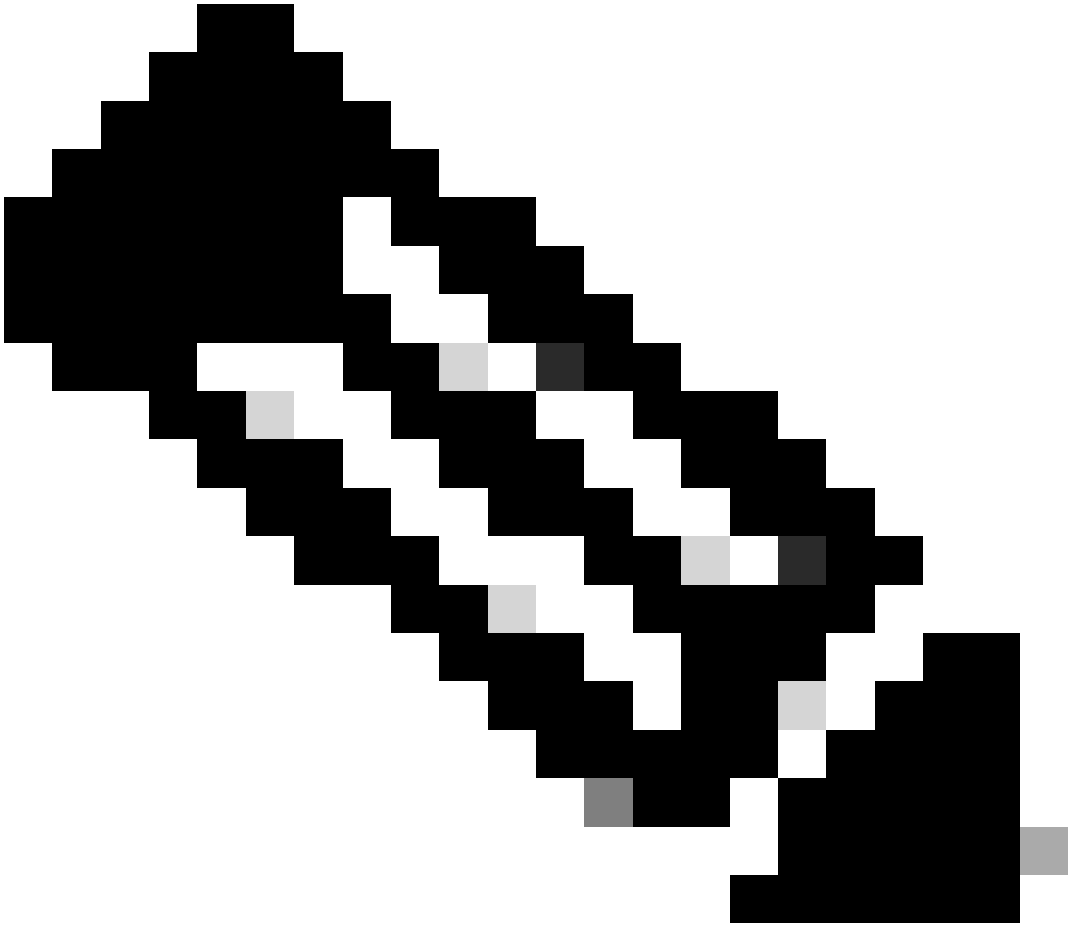
Attributo metrica



L'attributo metrica è anche detto MULTI_EXIT_DISCRIMINATOR, MED (BGP4) o INTER_AS (BGP3). L'attributo è un suggerimento per i neighbor esterni circa la preferenza di percorso in un AS. L'attributo fornisce un modo dinamico per influenzare un altro AS nel modo di raggiungere una determinata route quando in tale AS vi sono più punti di ingresso. Si preferisce un valore di metrica inferiore.

A differenza delle preferenze locali, la metrica viene scambiata tra AS. Una metrica viene trasferita in un AS ma non lascia l'AS. Quando un aggiornamento entra nell'AS con una determinata metrica, tale metrica viene utilizzata per prendere decisioni all'interno dell'AS. Quando lo stesso aggiornamento viene passato a un terzo AS, la metrica viene restituita a 0. Il diagramma in questa sezione mostra il set di metriche. Il valore predefinito della metrica è 0.

A meno che non riceva altre indicazioni, il router confronta le metriche per i percorsi dai neighbor nello stesso AS. Affinché il router possa confrontare le metriche dei neighbor provenienti da AS diversi, è necessario utilizzare il comando di configurazione speciale bgp always-compare-med sul router.



Nota: esistono due comandi di configurazione BGP che possono influenzare la selezione del percorso basata sul MED (multi-exit discriminator). I comandi sono bgp deterministic-med e bgp always-compare-med . Il comando bgp deterministic-med garantisce il confronto della variabile MED al momento di scegliere l'indirizzamento, quando peer diversi pubblicizzano nello stesso AS. Il comando bgp always-compare-med assicura il confronto del MED per i percorsi dei neighbor in diversi AS. Il comando bgp always-compare-med è utile quando più provider di servizi o aziende concordano una policy uniforme per l'impostazione del MED. Fare riferimento a In che modo il comando bgp deterministic-med differisce dal comando bgp always-compare-med per capire come questi comandi influenzano la selezione del percorso BGP.

Nel diagramma di questa sezione, AS100 ottiene informazioni sulla rete 192.168.180.10 tramite tre router diversi: RTC, RTD e RTB. RTC e RTD sono in AS300, mentre RTB è in AS400.

In questo esempio, il confronto dei percorsi AS su RTA viene ignorato dal comando bgp bestpath as-path ignore . È configurato per forzare BGP a ricadere sull'attributo successivo per il confronto dell'indirizzamento (in questo caso metrica o MED). Se il comando viene omesso, il BGP può

installare la route 192.168.180.10 dal router RTC come quella con il percorso AS più breve.

Si supponga di aver impostato la metrica proveniente da RTC su 120, la metrica proveniente da RTD su 200 e la metrica proveniente da RTB su 50. Per impostazione predefinita, un router confronta le metriche provenienti da router adiacenti nello stesso AS. Pertanto, RTA può solo confrontare la metrica che proviene da RTC con la metrica che proviene da RTD. RTA sceglie RTC come hop successivo migliore perché 120 è inferiore a 200. Quando RTA ottiene un aggiornamento da RTB con metrica 50, non può confrontare la metrica con 120 perché RTC e RTB si trovano in AS diversi. RTA deve scegliere in base ad altri attributi.

Per forzare RTA a confrontare le metriche, è necessario utilizzare il comando bgp always-compare-med su RTA. Queste configurazioni illustrano questo processo:

```
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
```

Con queste configurazioni, RTA sceglie RTC come next-hop, in considerazione del fatto che tutti gli altri attributi sono uguali. Per includere RTB nel confronto delle metriche, è necessario configurare RTA in questo modo:

```
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 questo caso, RTA sceglie RTB come next-hop migliore per raggiungere la rete 192.168.180.10.

Se si usa il comando **default-metricnumber**, è possibile impostare le metriche anche durante la ridistribuzione delle route in BGP.

Si supponga che, nell'esempio di questa sezione, RTB immetta una rete tramite static in AS100. La configurazione è la seguente:

```
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.
```

Attributo community

L'attributo community è un attributo transitorio e facoltativo compreso tra 0 e 4.294.967.200. L'attributo community consente di raggruppare le destinazioni di una determinata comunità e di applicare le decisioni di instradamento che corrispondono a tali comunità. Decisioni di routing sono ad esempio: accettazione, preferenza e ridistribuzione.

È possibile utilizzare le route map per impostare gli attributi community. Il comando set**della route map ha la seguente sintassi:**

<#root>

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

Alcune community predefinite e note da utilizzare in questo comando sono:

- **no-export**: non pubblicizzare ai peer eBGP. Mantiene questa route all'interno di un AS.

- **no-advertise**: non pubblicizzare questa route ad altri peer, interni o esterni.

- **Internet**: pubblicizza questa route alla community Internet. Qualsiasi router appartiene a questa community.

- **local-as**: da utilizzare in scenari di confederazione per impedire la trasmissione di pacchetti all'esterno dell'AS locale.

Seguono due esempi di route map che impostano la community:

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

o

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

Se non si imposta la parola chiave additive, 200 sostituisce qualsiasi vecchia community già esistente. Se si utilizza la parola chiave additive, si verifica un'aggiunta di 200 alla community. Anche se si imposta l'attributo community, per impostazione predefinita questo attributo non viene trasmesso ai neighbor. Per inviare l'attributo a un neighbor, è necessario utilizzare questo comando:

<#root>

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

Di seguito è riportato un esempio:

```
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
```

Nel software Cisco IOS versione 12.0 e successive, è possibile configurare le community in tre formati diversi: decimale, esadecimale e AA:NN. Per impostazione predefinita, il software Cisco IOS utilizza il vecchio formato decimale. Per configurare e visualizzare il pacchetto in A:NN, usare il comando **ip bgp-community new**-global configuration format. La prima parte del formato AA:NN rappresenta il numero AS, mentre la seconda parte rappresenta un numero a 2 byte.

Di seguito è riportato un esempio:

Senza il comando **ip bgp-community new-format** nella configurazione globale, il comando **show ip bgp 10.6.0.0 visualizza il valore dell'attributo community in formato decimale.** In questo esempio, il valore dell'attributo community è 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

A questo punto, eseguire il comando ip bgp-community new-format a livello globale su questo 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

Con **il** comando **ip bgp-community new-**formatglobal configuration, il valore della community viene visualizzato in formato AA:NN.

Nell'output del comando **show ip bgp 10.6.0.0**, il valore visualizzato è 100:20:

<#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
```

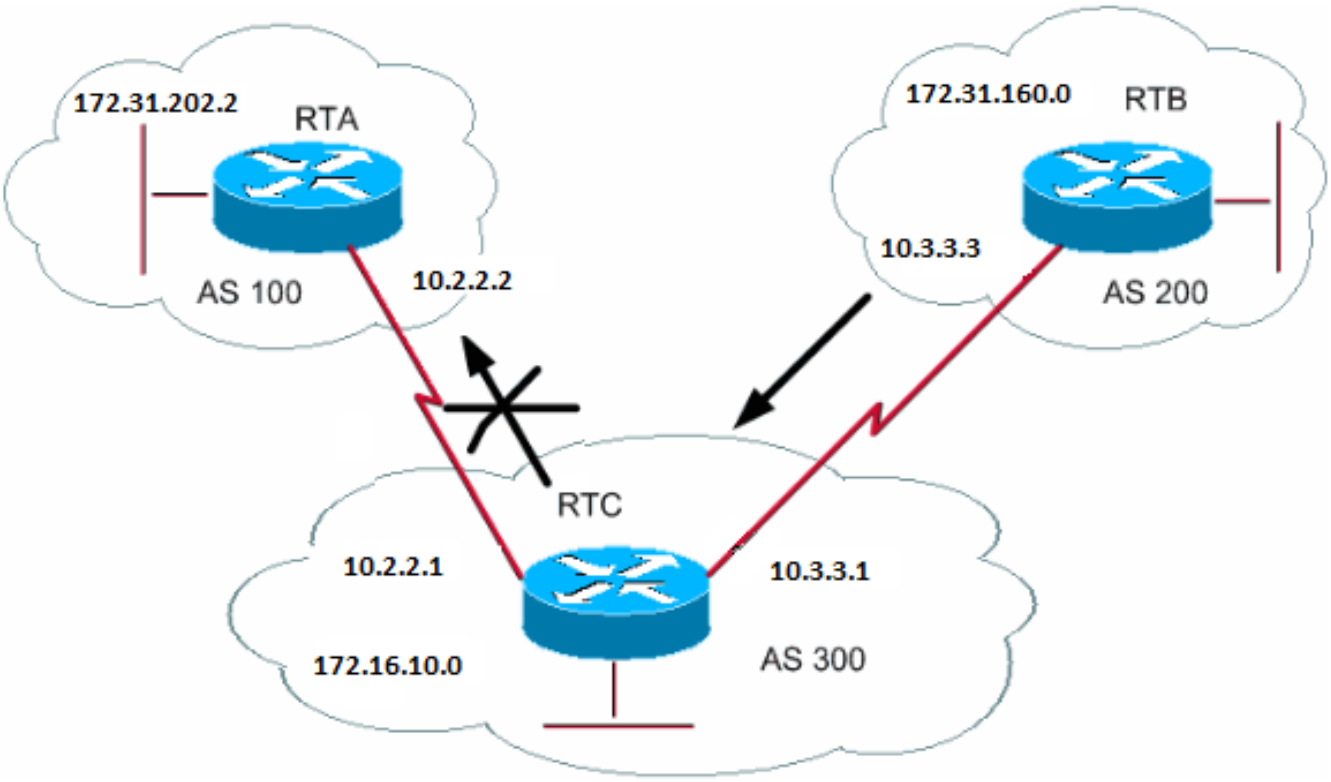
Community: 100:20

Case study BGP 3

Filtro BGP

Diversi metodi di filtro consentono di controllare l'invio e la ricezione degli aggiornamenti BGP. È possibile filtrare gli aggiornamenti BGP in base alle informazioni sul percorso o alle informazioni sul percorso o alle community. Tutti i metodi ottengono gli stessi risultati. La scelta di un metodo rispetto a un altro dipende dalla configurazione della rete specifica.

Filtro ciclo di lavorazione



Per limitare le informazioni di routing acquisite o pubblicizzate dal router, è possibile filtrare il BGP utilizzando gli aggiornamenti di routing da o verso un particolare neighbor. Si definisce un elenco degli accessi e lo si applica agli aggiornamenti da o verso un neighbor. Utilizzare questo comando nella modalità di configurazione del router:

<#root>

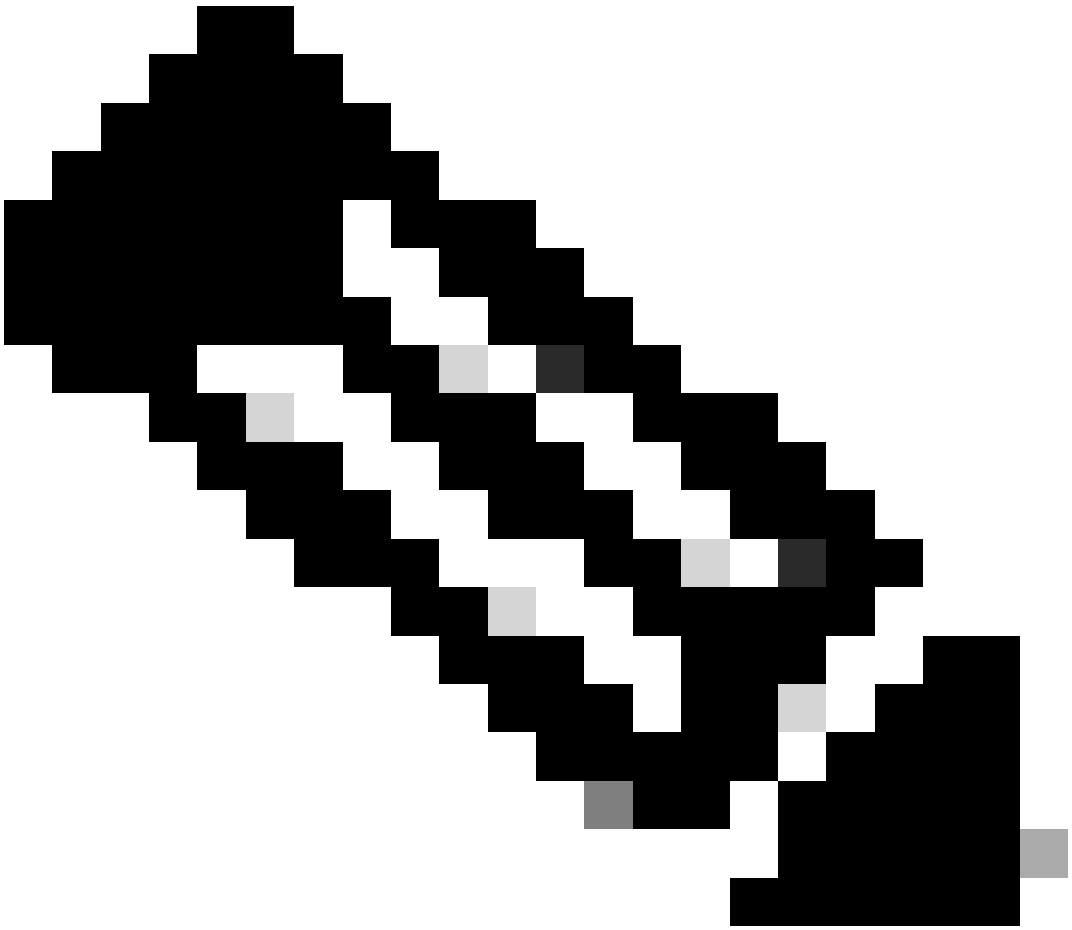
neighbor {ip-address | peer-group-name} distribute-list access-list-number {in | out}

In questo esempio, RTB genera la rete 172.31.160.0 e invia l'aggiornamento a RTC. Se RTC desidera interrompere la propagazione degli aggiornamenti di AS100, è necessario definire un elenco di accessi per filtrare tali aggiornamenti e applicare l'elenco di accessi durante la comunicazione con RTA:

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

access-list 1 deny 172.31.160.0 0.0.255.255

access-list 1 permit 0.0.0.0 255.255.255.255

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

L'uso degli elenchi degli accessi è un po' complicato quando si ha a che fare con supernet che possono causare alcuni conflitti.

Si supponga che, come nell'esempio in questa sezione, RTB abbia subnet diverse di 160.10.xx L'obiettivo è filtrare gli aggiornamenti e pubblicizzare solo 192.168.160.0/8.



Nota: la notazione /8 indica che si utilizzano 8 bit di subnet mask, che iniziano dall'estrema sinistra dell'indirizzo IP. Questo indirizzo è equivalente a 192.168.160.0 255.0.0.0.

Il comando access-list 1 permit 192.168.160.0 0.255.255.25 5permette 192.168.160.0/8, 192.168.160.0/9, e così via. Per limitare l'aggiornamento solo a 192.168.160.0/8, è necessario utilizzare un elenco degli accessi esteso con questo formato:

<#root>

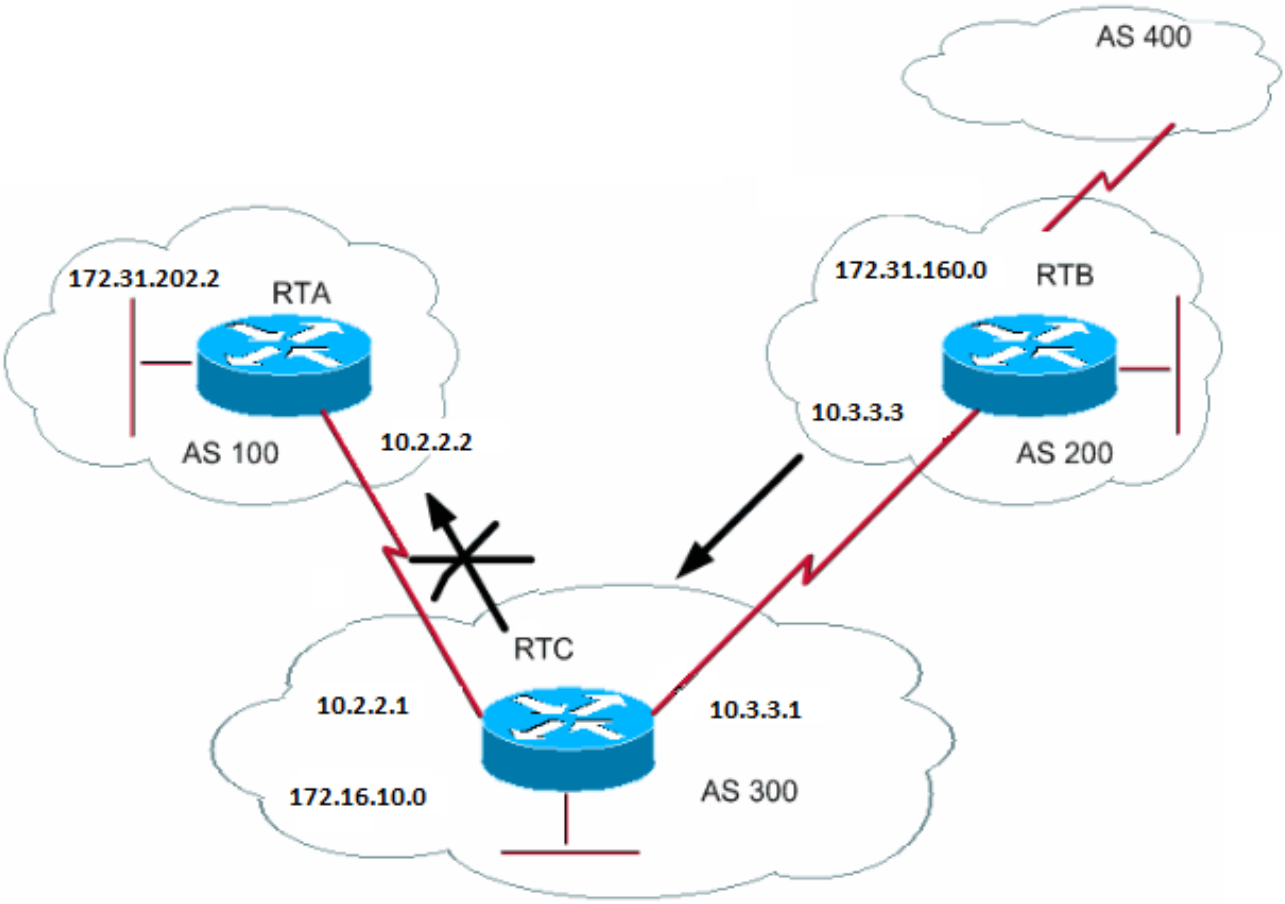
```
access-list 101 permit ip 192.168.160.0 0.255.255.255 255.0.0.0 0.0.0.0.
```

Questo elenco autorizza solo 192.168.160.0/8.

Per configurazioni di esempio su come filtrare le reti dai peer BGP, fare riferimento al blocco uno o più reti da un peer BGP. Il metodo utilizza il comando **distribute-list** con elenchi di controllo di accesso (ACL) standard ed estesi, oltre alla possibilità di filtrare l'elenco di prefissi.

Filtro percorso

Potete anche filtrare i tracciati.



È possibile specificare un elenco degli accessi per gli aggiornamenti in entrata e in uscita utilizzando le informazioni sui percorsi AS di BGP. Nel diagramma di questa sezione è possibile bloccare gli aggiornamenti relativi a 172.31.160.0 in modo che non vengano visualizzati in AS100. Per bloccare gli aggiornamenti, definire un elenco degli accessi in RTC che impedisca la trasmissione a AS100 di qualsiasi aggiornamento originato da AS200. Utilizzare i seguenti comandi:

<#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}

In questo esempio viene interrotto l'invio da RTC a RTA di aggiornamenti su 172.31.160.0:

```
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 .*
```

Il access-list 1comando di questo esempio forza il rifiuto di qualsiasi aggiornamento con informazioni sul percorso che inizia con 200 e termina con 200. L'espressione`^200$`nel comando è un'"espressione regolare", in cui`^`significa "inizia con" e`$`significa "termina con". Poiché RTB invia gli aggiornamenti alla versione 172.31.160.0 con le informazioni sul percorso che iniziano con 200 e terminano con 200, gli aggiornamenti corrispondono all'elenco degli accessi. L'elenco degli accessi nega questi aggiornamenti.

L'espressione`.*`è un'altra espressione regolare in cui`.`significa "qualsiasi carattere" e`*`significa "la ripetizione di quel carattere". Pertanto .* rappresenta qualsiasi informazione sul percorso, necessaria per consentire la trasmissione di tutti gli altri aggiornamenti.

Cosa succede se, invece di utilizzare`^200$`, si utilizza`^200`? Con un AS400, come nel diagramma di questa sezione, gli aggiornamenti originati da AS400 hanno informazioni sul percorso del modulo (200, 400). In questo percorso le informazioni sono 200 e 400. Questi aggiornamenti corrispondono all'elenco degli accessi`^200`, in quanto le informazioni sul percorso iniziano con 200. L'elenco degli accessi impedisce la trasmissione di questi aggiornamenti a RTA, che non è il requisito.

Per verificare se l'espressione regolare implementata è corretta, usare il comando **show ip bgp regexpregular-expression**. Questo comando mostra tutti i percorsi che corrispondono alla configurazione dell'espressione regolare.

Espressione regolare AS

Questa sezione illustra la creazione di un'espressione regolare.

Un'espressione regolare è un modello per la corrispondenza con una riga di input. Quando si crea un'espressione regolare, si specifica una riga a cui deve corrispondere l'input. Nel caso di BGP, si specifica una riga costituita da informazioni sul percorso a cui deve corrispondere un input.

Nell'esempio della sezione **Filtro percorso**, è stata specificata la stringa`^200$`. Si desidera che le informazioni sul percorso incluse negli aggiornamenti corrispondano alla stringa per poter essere scelte.

Un'espressione regolare comprende:

- **Intervallo**

  Un intervallo è una sequenza di caratteri tra parentesi quadre. Un esempio è [abcd].

- **Atomo**

  Un atomo è un singolo carattere. Seguono alcuni esempi:

`.`

- Il `carattere .`corrisponde a qualsiasi carattere singolo.

^

- ^ corrisponde all'inizio della riga di input.

$

- $ corrisponde alla fine della riga di input.

\

- Il carattere\corrisponde al carattere.

_

- Il_carattere corrisponde a una virgola (`,`), una parentesi graffa aperta (`{`), una parentesi graffa chiusa (`}`), l'inizio della stringa di input, la fine della stringa di input o uno spazio.

•

Pezzo

Un pezzo è uno di questi simboli, che viene dopo un atomo:

*

- * corrisponde a 0 o più sequenze dell'atomo.

+

- + corrisponde a 1 o più sequenze dell'atomo

?

- Il carattere?corrisponde all'atomo o alla stringa null.

- **Filiale**

  Una filiale è costituita da 0 o più pezzi concatenati.

Seguono alcuni esempi di espressioni regolari:

a*

- Questa espressione indica qualsiasi occorrenza della lettera "a", incluso nessuna.

```
a+
```

- Questa espressione indica che deve essere presente almeno un'occorrenza della lettera "a".

```
ab?a
```

- Questa espressione corrisponde a "aa" o "aba".

```
_100_
```

- Questa espressione indica tramite AS100.

```
_100$
```

- Questa espressione indica un'origine di AS100.

```
^100 .*
```

- Questa espressione indica la trasmissione da AS100.

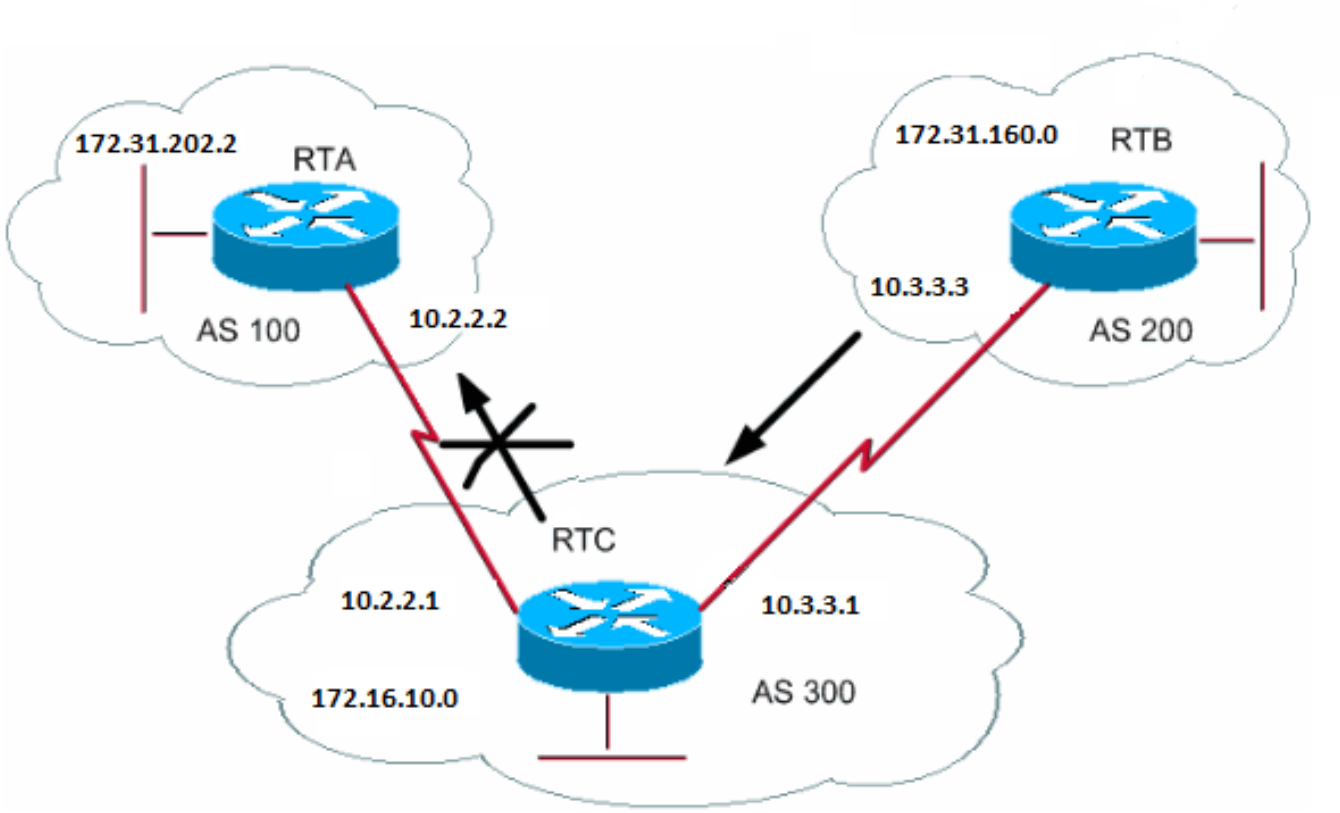
```
^$
```

- Questa espressione indica l'origine da questo AS.

Per configurazioni di esempio relative al filtro delle espressioni regolari, fare riferimento a Usa espressioni regolari in BGP.

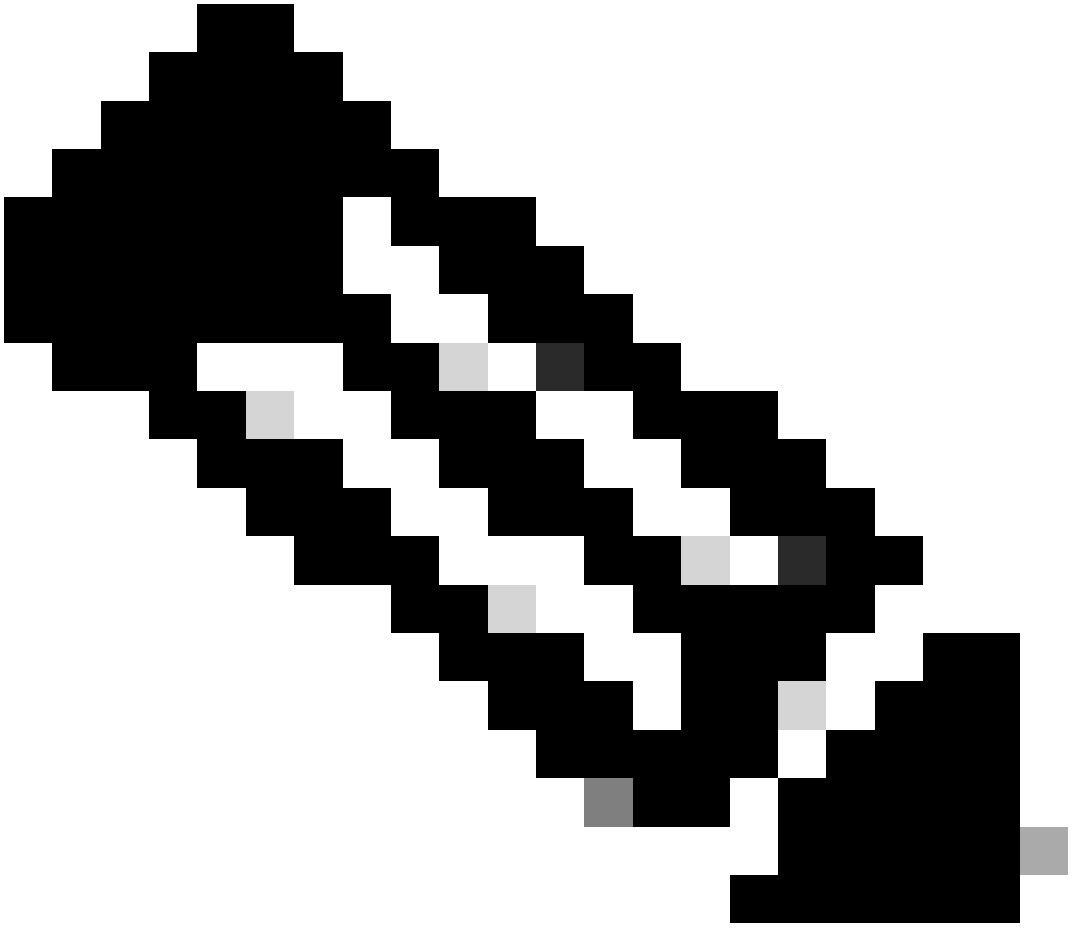
Filtro community BGP

Questo documento illustra il filtro delle route e il filtro dei percorsi AS. Un altro metodo è il filtro community. La sezione Attributo community descrive la community e in questa sezione vengono forniti alcuni esempi di utilizzo della community.

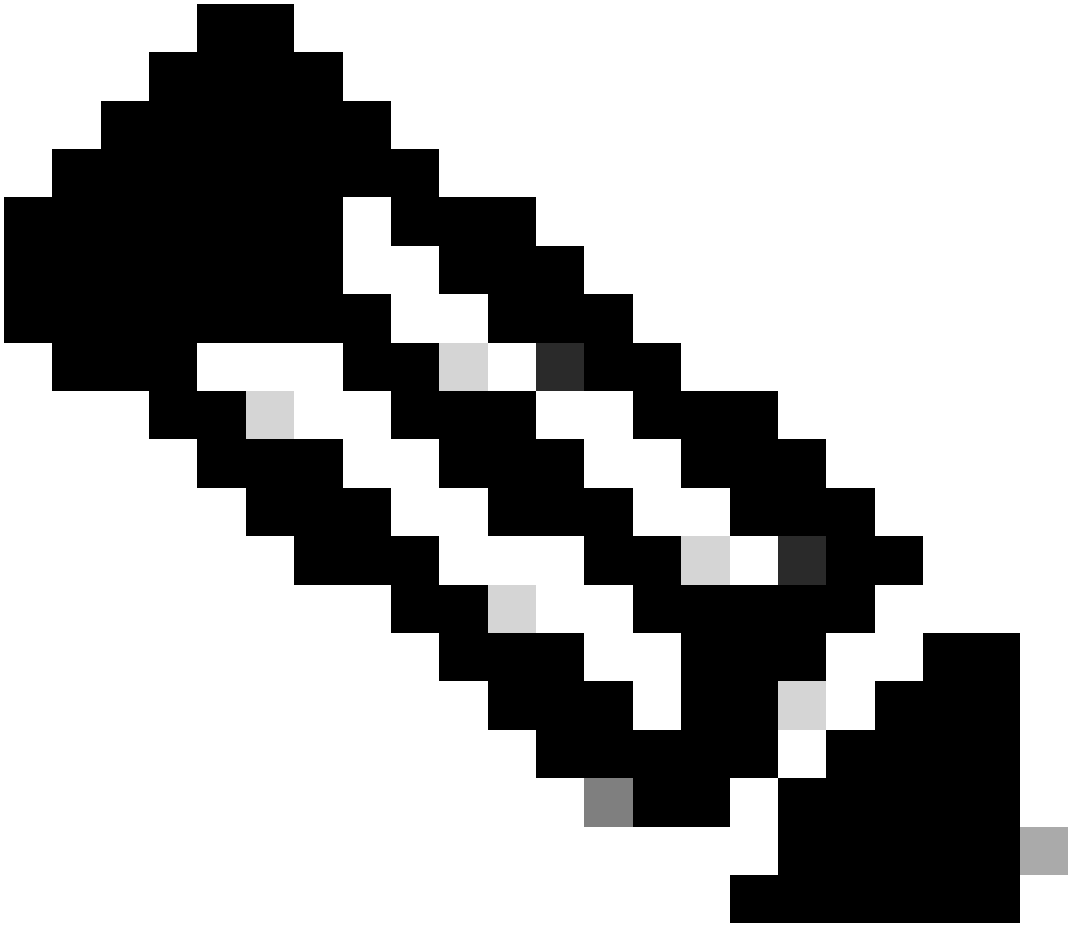


In questo esempio, RTB deve impostare l'attributo community sulle route BGP pubblicizzate da RTB in modo che RTC non le propaghi ai peer esterni. Utilizzare l'attributo no-exportcommunity.

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

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

access-list 1 permit 0.0.0.0 255.255.255.255
```



Nota: in questo esempio viene utilizzato il comandoroute-map setcommunity per impostare la community in modo che non venga esportata.



Nota: il **neighbor send-community** comando è necessario per inviare questo attributo a RTC.

Quando RTC ottiene gli aggiornamenti con l'attributo NO_EXPORT, RTC non propaga gli aggiornamenti all'RTA peer esterno.

In questo esempio, RTB ha impostato l'attributo community su**100 200 additive** . Questa azione aggiunge il valore 100 200 a qualsiasi valore comunitario corrente prima della trasmissione a 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
```

Un elenco di community è un gruppo di community utilizzate in una clausola match di una route map. L'elenco delle community consente di filtrare o impostare gli attributi in base a diversi elenchi di numeri di community.

<#root>

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

Ad esempio, è possibile definire questa route map, 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.
```

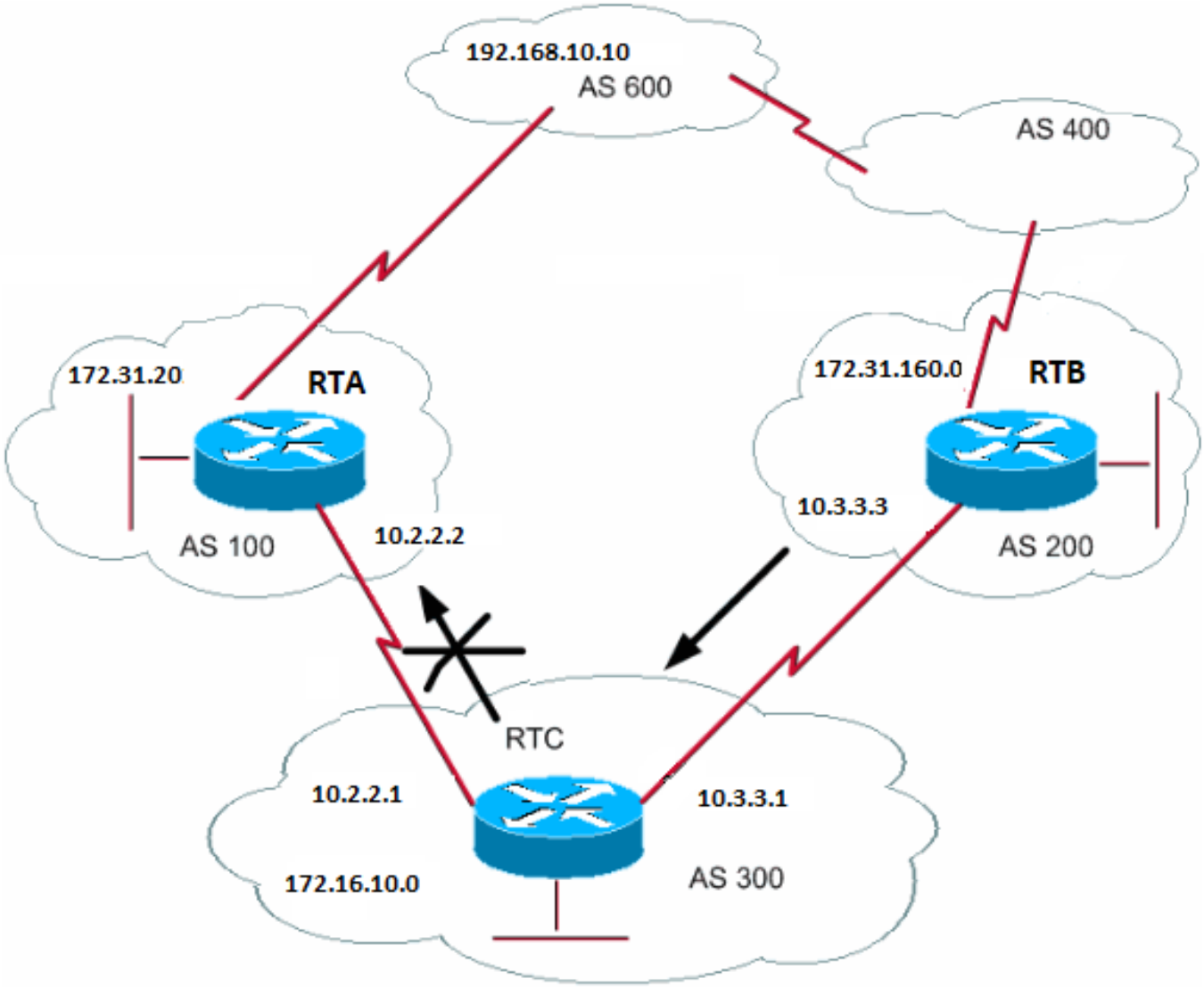
È possibile utilizzare l'elenco delle community per filtrare o impostare alcuni parametri, come peso e metrica, in determinati aggiornamenti in base al valore della community. Nel secondo esempio di questa sezione, RTB ha inviato aggiornamenti a RTC con una community di 100 200. Se RTC desidera impostare il peso con questi valori come base, è possibile eseguire questa operazione:

```
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 questo esempio, qualsiasi route con 100 nell'attributo community corrisponde all'elenco 1. Il peso di questa route è impostato su 20. Qualsiasi route che abbia solo 200 corrispondenze della community nell'elenco 2 e che pesi 20. La parola chiave **precise** afferma che la Comunità è composta solo da 200 persone e nient 'altro. L'ultimo elenco di exact si assicura che non vengano rilasciati altri aggiornamenti. Si ricordi che qualsiasi cosa che non corrisponde viene eliminata, per impostazione predefinita. La parola chiave internet indica tutte le route perché tutte le route sono membri della community Internet.

Per ulteriori informazioni, fare riferimento a Configurazione e controllo di una rete di provider upstream con valori della community BGP.

BGP neighbor e route map



È possibile utilizzare il comando neighbor insieme alle route map per filtrare o impostare i parametri sugli aggiornamenti in entrata e in uscita.

Le route map associate all'istruzione neighbor non hanno effetto sugli aggiornamenti in arrivo quando si esegue la corrispondenza in base all'indirizzo IP:

<#root>

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

Si supponga che, come nel diagramma di questa sezione, si desideri che RTC acquisisca informazioni da AS200 sulle reti locali per AS200 e nient'altro. Inoltre, si desidera impostare su 20 il peso delle route accettate. Utilizzare una combinazione di elenchi degli accessi **adiacenti** e **come percorso**:

```
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$
```

Per tutti gli aggiornamenti che hanno origine da AS200, le informazioni sul percorso iniziano con 200 e terminano con 200. Questi aggiornamenti sono consentiti. Eventuali altri aggiornamenti vengono eliminati.

Si supponga di desiderare:

- L'accettazione degli aggiornamenti originati da AS200 e con un peso di 20
- L'eliminazione degli aggiornamenti originati da AS400
- Un peso 10 per gli altri aggiornamenti

```
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 .*
```

Questa istruzione imposta un peso di 20 per gli aggiornamenti locali di AS200. L'istruzione imposta inoltre un peso di 10 per gli aggiornamenti precedenti a AS400 e rifiuta gli aggiornamenti provenienti da AS400.

Utilizzo del comando set as-path prepend

In alcune situazioni, è necessario modificare le informazioni sul percorso per modificare il processo decisionale di BGP. Il comando da utilizzare con una route map è:

<#root>

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

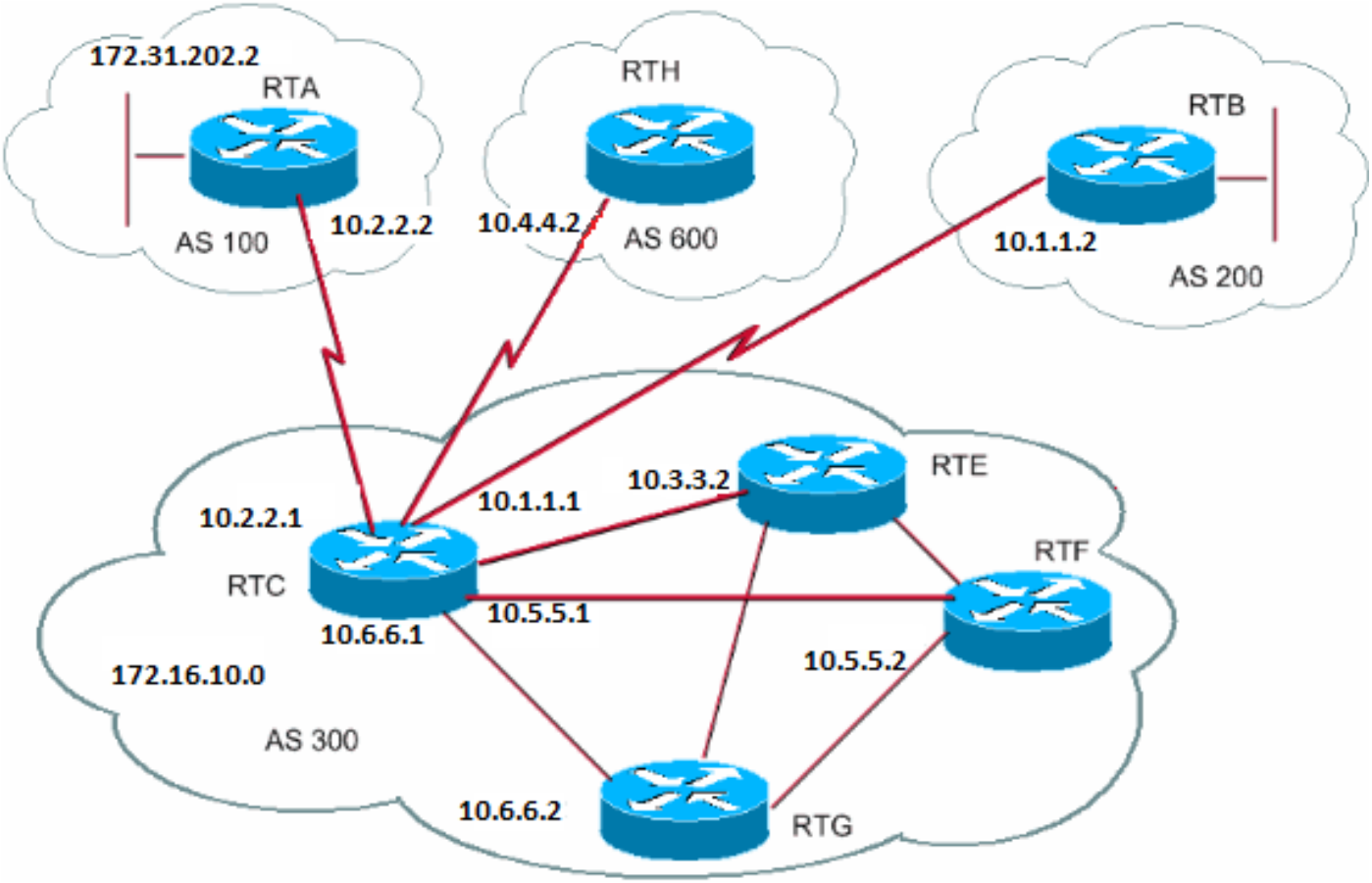
Si supponga che, nel diagramma della sezione BGP Neighbors and Route Maps, RTC pubblicizzi la propria rete 172.16.10.0 a due SA differenti, AS100 e AS200. Quando le informazioni vengono propagate a AS600, i router di AS600 dispongono di informazioni sulla raggiungibilità della rete pari a 172.16.10.0 tramite due percorsi diversi. La prima route è via AS100 con percorso (100, 300) e la seconda è via AS400 con percorso (400, 200, 300). Se tutti gli altri attributi sono uguali, AS600 sceglie il percorso più breve e sceglie la route tramite AS100.

AS300 ottiene tutto il traffico tramite AS100. Se si desidera influenzare questa decisione dalla fine di AS300, è possibile fare in modo che il percorso attraverso AS100 appaia più lungo del percorso che attraversa AS400. A tale scopo, anteporre i numeri AS alle informazioni sul percorso corrente annunciate in AS100. È prassi comune ripetere il proprio numero AS nel modo seguente:

```
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
```

A causa di questa configurazione, AS600 riceve gli aggiornamenti relativi alla versione 172.16.10.0 tramite AS100 con le informazioni sul percorso seguenti: (100, 300, 300, 300). Queste informazioni sul percorso sono più lunghe di (400, 200, 300) che AS600 ha ricevuto da AS400.

Gruppi di peer BGP



Un gruppo di peer BGP è un gruppo di BGP neighbor con le stesse policy di aggiornamento. Le route map, gli elenchi di distribuzione e gli elenchi di filtri in genere impostano le policy di aggiornamento. Non vengono definiti gli stessi criteri per ogni router adiacente separato, ma viene definito un nome di gruppo peer e vengono assegnati questi criteri al gruppo peer.

I membri del gruppo di peer ereditano tutte le opzioni di configurazione del gruppo di peer. È inoltre possibile configurare i membri per sovrascrivere queste opzioni se le opzioni non influiscono sugli aggiornamenti in uscita. È possibile ignorare solo le opzioni impostate sull'ingresso.

Per definire un gruppo di peer, utilizzare questo comando:

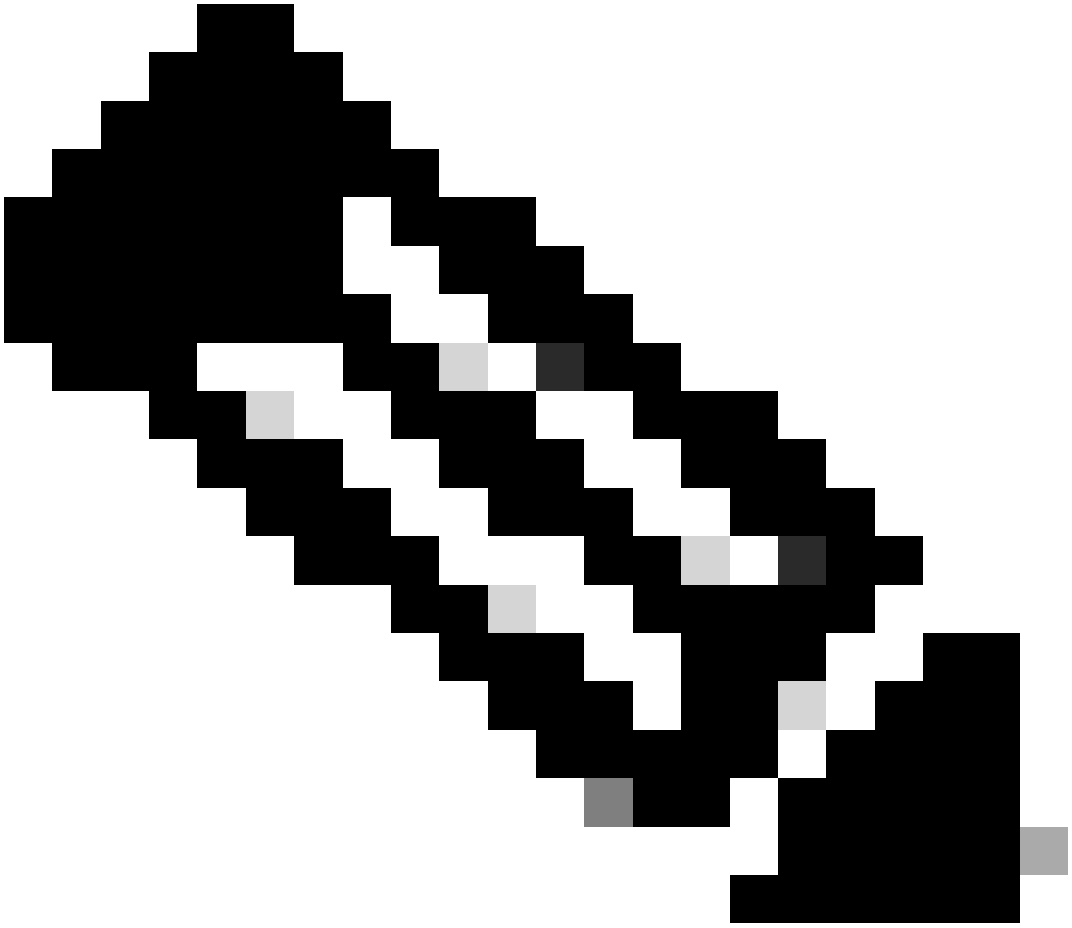
```
<#root>
```

neighbor peer-group-name peer-group

Questo esempio applica i gruppi di peer ai BGP neighbor interni ed esterni:

```
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
```

Questa configurazione definisce un gruppo di peer con il nome internalmap. La configurazione definisce alcuni criteri per il gruppo, ad esempio una mappa di route **SETMETRIC** per impostare la metrica su 5 e due elenchi di filtri diversi, 1 e 2. La configurazione applica il gruppo peer a tutti i router adiacenti interni, RTE, RTF e RTG. Inoltre, la configurazione definisce un elenco di filtri 3 separato per l'RTE neighbor. Questo elenco di filtri sostituisce l'elenco di filtri 2 all'interno del gruppo di peer.

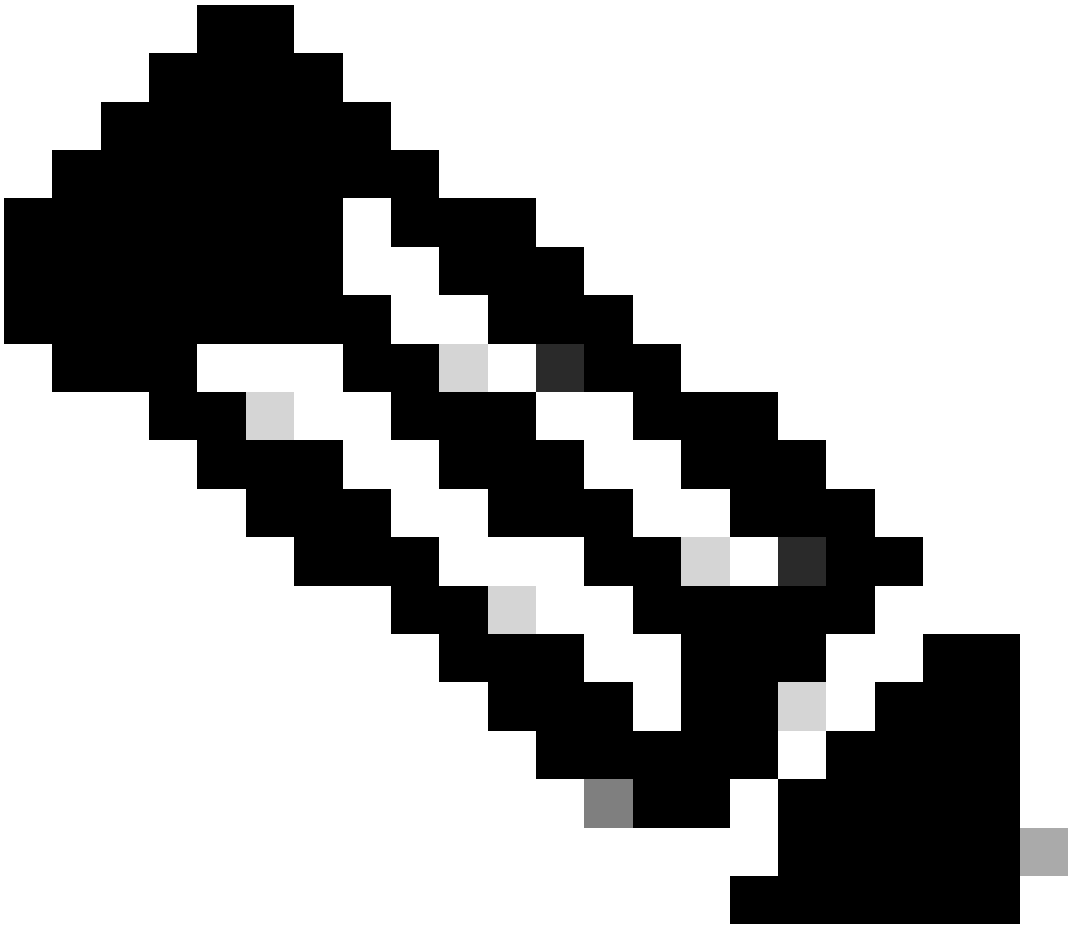


Nota: è possibile sostituire solo le opzioni che influiscono sugli aggiornamenti in entrata.

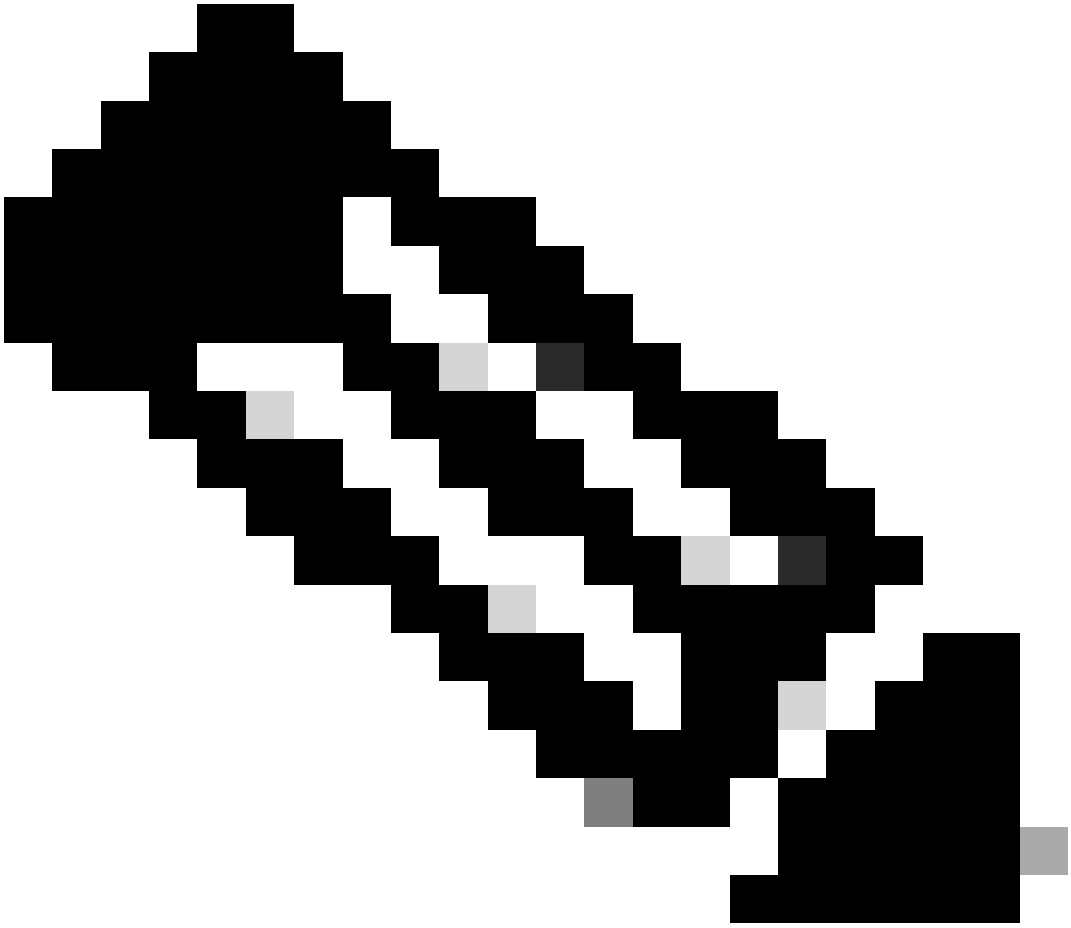
Ora vediamo come è possibile utilizzare i gruppi di peer con neighbor esterni. Con lo stesso diagramma di questa sezione, si configura RTC con una externalmap di gruppo di peer e si applica il gruppo di peer ai neighbor esterni.

```
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
```



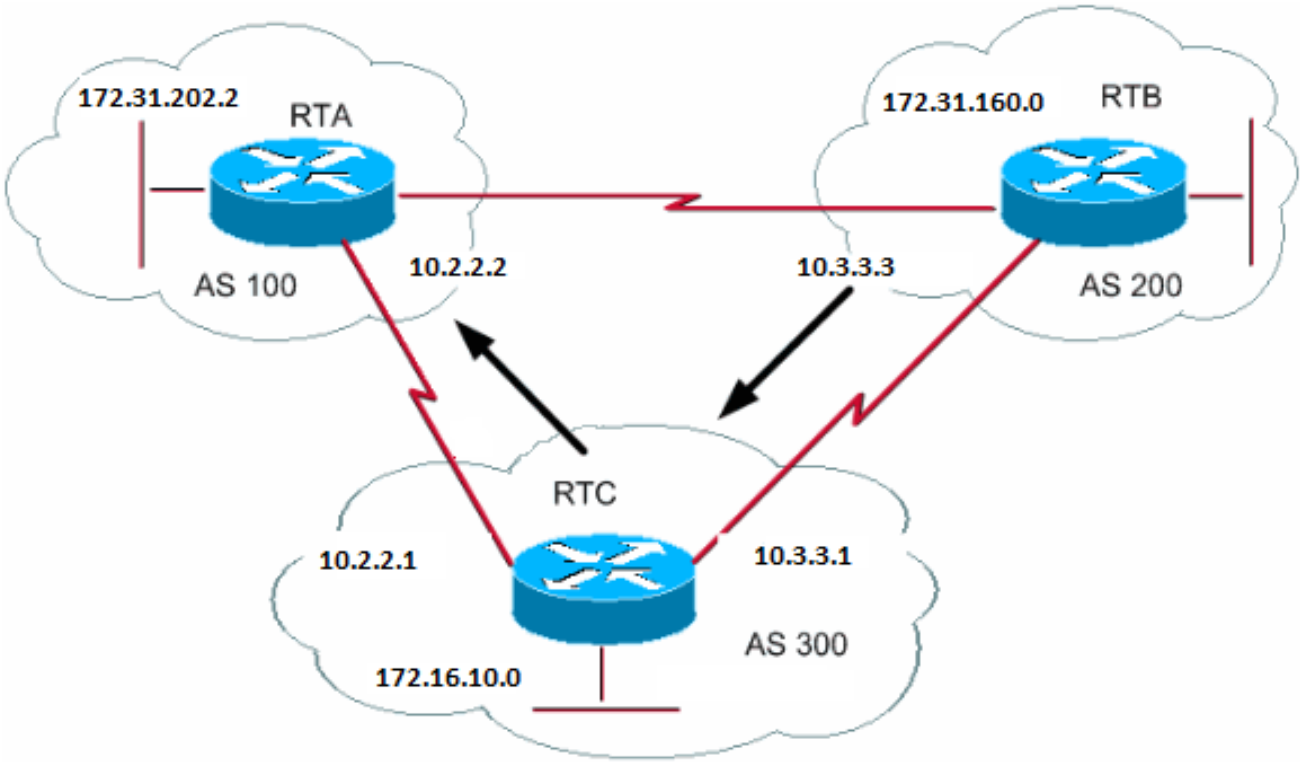
Nota: in queste configurazioni, si definiscono le istruzioni remote-as all'esterno del gruppo di peer perché è necessario definire diversi AS esterni. Inoltre, assegnando l'elenco dei filtri 3 vengono sovrascritti gli aggiornamenti in entrata dell'10.1.1.2 neighbor. Per ulteriori informazioni sui gruppi di peer, fare riferimento a Gruppi di peer BGP.



Nota: nel software Cisco IOS versione 12.0(24)S, Cisco ha introdotto la funzionalità BGP Dynamic Update Peer Group. La funzione è disponibile anche nelle versioni successive del software Cisco IOS. La funzione introduce un nuovo algoritmo che calcola e ottimizza dinamicamente i gruppi di aggiornamento dei neighbor che condividono le stesse policy in uscita. Questi neighbor possono condividere gli stessi messaggi di aggiornamento. Nelle versioni precedenti del software Cisco IOS, il gruppo di messaggi di aggiornamento BGP era basato su configurazioni di gruppi di peer. Questo metodo serve per raggruppare gli aggiornamenti delle policy in uscita limitate e configurazioni di sessione specifiche. La funzione Gruppo di peer per aggiornamento dinamico BGP separa la replica del gruppo di aggiornamento dalla configurazione del gruppo di peer. Questa separazione migliora il tempo di convergenza e la flessibilità della configurazione dei neighbor. Per ulteriori dettagli, fare riferimento a Gruppi di peer per aggiornamento dinamico BGP.

Case study BGP 4

CIDR e indirizzi aggregati



Uno dei principali miglioramenti di BGP4 rispetto a BGP3 è il CIDR (classless interdomain routing). Il CIDR o supernetting è un nuovo modo di esaminare gli indirizzi IP. Con CIDR non esiste alcun concetto di classe, come ad esempio la classe A, B o C. Ad esempio, la rete 192.168.213.0 una volta era una rete di classe C non valida. La rete è una supernet legale, 192.168.213.0/16. Il valore 16 rappresenta il numero di bit nella subnet mask, quando si conta dall'estremità sinistra dell'indirizzo IP. Questa rappresentazione è simile a 192.168.213.0 255.255.0.0.

Utilizzare gli aggregati per ridurre al minimo le dimensioni delle tabelle di routing. L'aggregazione è il processo che combina le caratteristiche di diverse route in modo da consentire di pubblicizzare una singola route. Nell'esempio, RTB genera la rete 172.31.160.0. Configurare RTC in modo da propagare una supernet della route 192.168.160.0 all'RTA:

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

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

RTC propaga l'indirizzo aggregato 192.168.160.0 a RTA.

Comandi aggregati

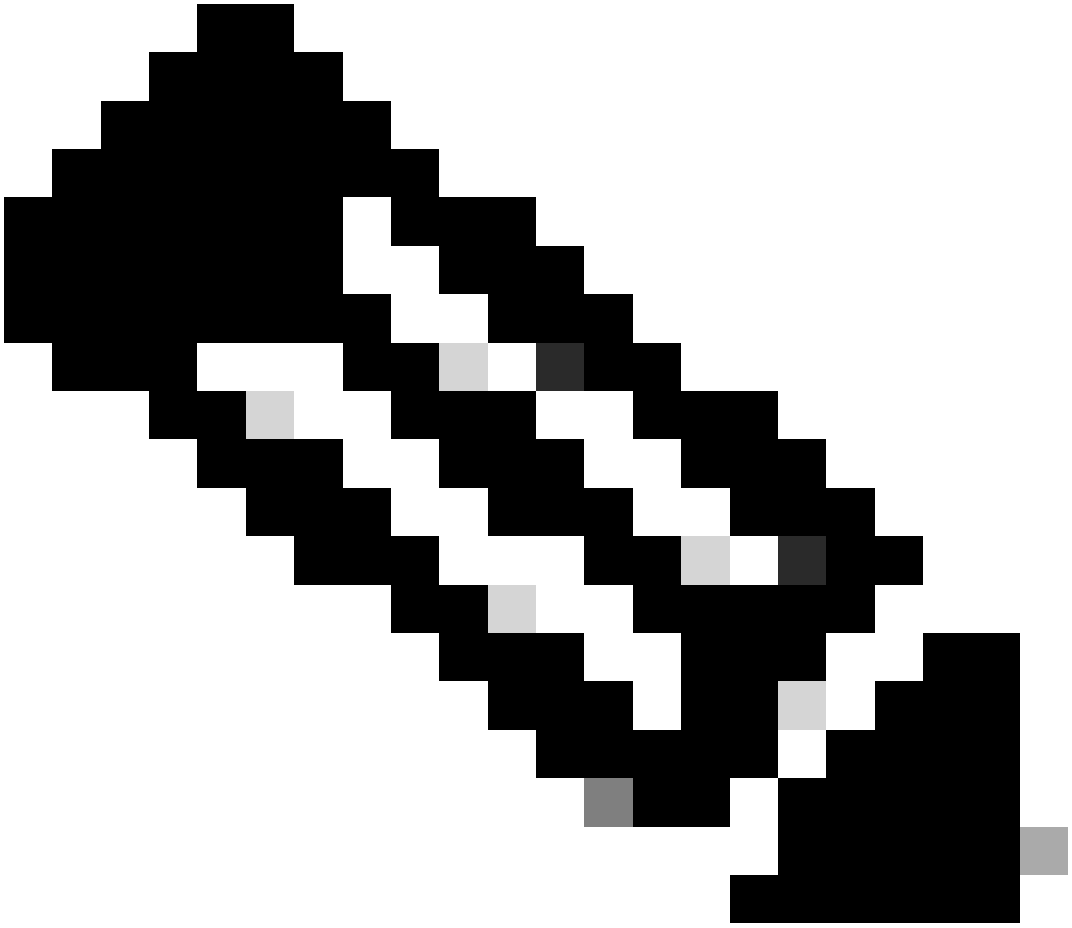
Esiste un'ampia gamma di comandi aggregati. È necessario comprendere il funzionamento di ognuno per avere il comportamento di aggregazione desiderato.

Il primo comando è quello dell'esempio riportato nella sezione CIDR e indirizzi aggregati:

<#root>

aggregate-address address-mask

Questo comando pubblicizza la route del prefisso e tutte le route più specifiche. Il comando **aggregate-address 192.168.160.0** propaga una rete aggiuntiva 192.168.160.0 ma non impedisce la propagazione di 172.31.160.0 a RTA. Il risultato è la propagazione di entrambe le reti 192.168.160.0 e 172.31.160.0 su RTA, ovvero la pubblicizzazione del prefisso e della route più specifica.



Nota: non è possibile aggregare un indirizzo se non si dispone di una route più specifica di tale indirizzo nella tabella di routing BGP.

Ad esempio, RTB non può generare un'aggregazione per 192.168.160.0 se RTB non ha una voce più specifica di 192.168.160.0 nella tabella BGP. È possibile inserire una route più specifica nella tabella BGP. L'immissioone della route può avvenire tramite:

- Aggiornamenti in arrivo da altri AS
-

Ridistribuzione di un prodotto IGP o statico in BGP

- Comando network, ad esempio, network 172.31.160.0

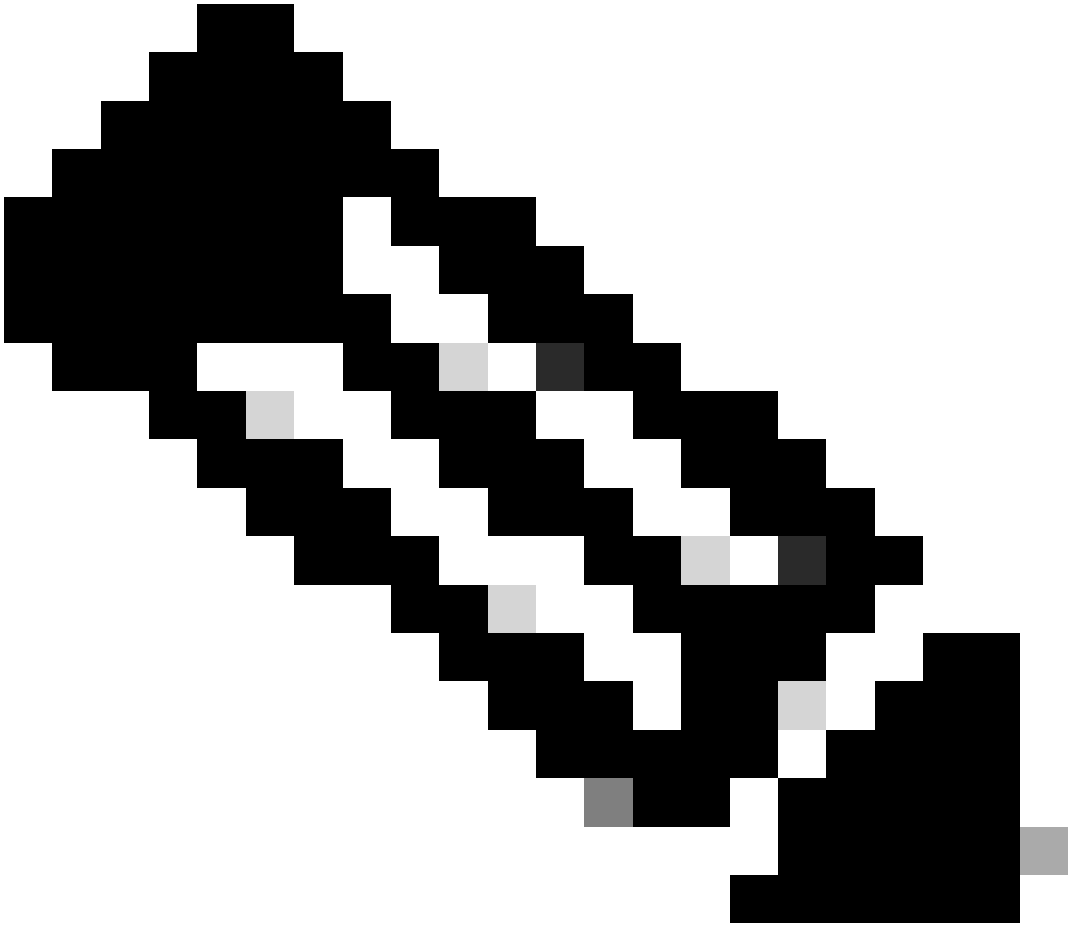
Se si desidera che RTC propaghi solo la rete 192.168.160.0 e non la route più specifica, immettere questo comando:

<#root>

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

Questo comando pubblicizza solo il prefisso. Il comando elimina tutte le route più specifiche.

Il comando **aggrega 192.168.160.0 255.0.0.0** propaga la rete 192.168.160.0 e sopprime la route più specifica 172.31.160.0.



Nota: se si aggrega una rete iniettata nel BGP tramite l'istruzione network, la voce network viene sempre inserita negli aggiornamenti BGP. Questa immissione si verifica anche se si utilizza il comando aggregate summary-only. L'esempio della sezione Esempio CIDR 1 illustra questa situazione.

<#root>

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

Questo comando pubblicizza il prefisso e le route più specifiche. Ma il comando include le informazioni as-set nelle informazioni sul percorso degli aggiornamenti di routing.

<#root>

aggregate 192.168.0.0 255.0.0.0 as-set

Nella sezione CIDR Esempio 2 (as-set) viene descritto questo comando.

Se si desidera eliminare le route più specifiche quando si esegue l'aggregazione, definire una route map e applicare la route map agli aggregati. L'azione consente di scegliere in modo selettivo le route più specifiche da eliminare.

<#root>

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

Questo comando pubblicizza il prefisso e le route più specifiche. Ma il comando elimina la pubblicizzazione basata sulla route map. Si supponga che, con il diagramma nella sezione CIDR e indirizzi aggregati, si desideri aggregare 192.168.160.0, sopprimere la route più specifica 192.168.160.20 e consentire la propagazione di 172.31.160.0. Utilizza questa mappa route:

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

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

Per definizione della suppress-map, vi è una soppressione dagli aggiornamenti di tutti i pacchetti consentiti dall'elenco degli accessi.

Quindi, applicare la route map all'istruzione aggregate .

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

Questa è un'altra variante:

<#root>

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

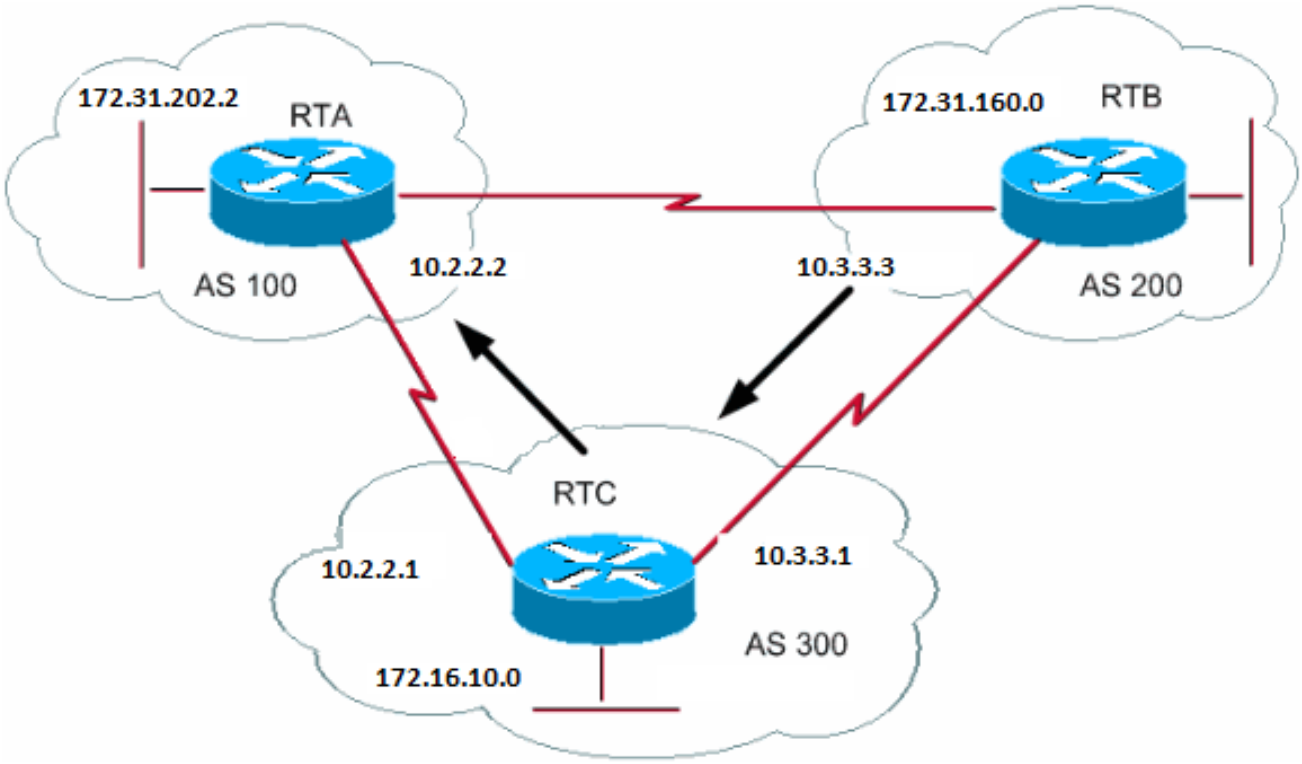
Questo comando consente di impostare gli attributi, ad esempio la metrica, al momento dell'invio degli aggregati. Per impostare l'origine degli aggregati su IGP, applicare questa route map al comando aggregate attribute-map

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

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

Per ulteriori informazioni, fare riferimento a Informazioni sull'aggregazione delle route in BGP.

Esempio CIDR 1



Richiesta: consentire a RTB di annunciare il prefisso 192.168.160.0 ed eliminare tutte le route più specifiche. Il problema di questa richiesta è che la rete 172.31.160.0 è locale rispetto a AS200, quindi AS200 è l'iniziatore di 172.31.160.0. Non è possibile fare in modo che RTB generi un prefisso per 192.168.160.0 senza la generazione di una voce per 172.31.160.0, anche se si utilizza il comando **aggregate summary-only**. RTB genera entrambe le reti perché RTB è l'iniziatore di 172.31.160.0. Ci sono due soluzioni a questo problema.

La prima soluzione consiste nell'utilizzare una route statica e ridistribuire in BGP. Il risultato è che RTB pubblicizza l'aggregato con un'origine incompleta (?).

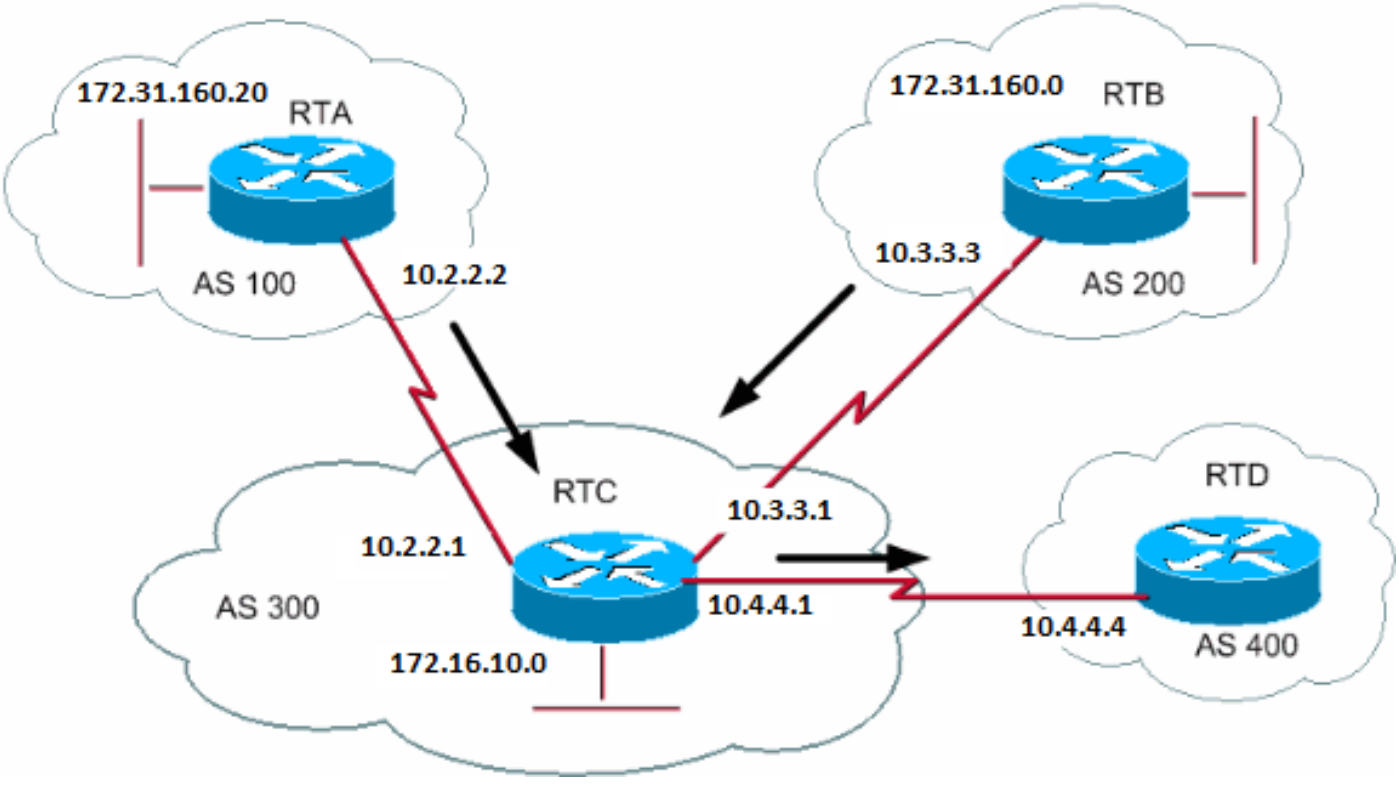
```
RTB#
router bgp 200
 neighbor 10.3.3.1 remote-as 300
 redistribute static

!--- 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
```

Nella seconda soluzione, oltre all'indirizzamento statico, si aggiunge una voce per il comandonetwork. Questa voce ha lo stesso effetto, tranne per il fatto che imposta l'origine dell'aggiornamento su 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
```

Esempio CIDR 2 (as-set)

Si usa l'istruzione as-set nell'aggregazione per ridurre la dimensione delle informazioni sul percorso. Con as-set, il numero AS viene elencato una sola volta, indipendentemente da quante volte il numero AS è apparso in più percorsi aggregati. Si utilizza il comando aggregate as-set in situazioni in cui l'aggregazione delle informazioni causa la perdita di informazioni in relazione all'attributo del percorso. In questo esempio, RTC ottiene gli aggiornamenti su 192.168.160.20 da RTA e gli aggiornamenti su 172.31.160.0 da RTB. Supponiamo che RTC voglia aggregare la rete 192.168.160.0/8 e inviarla a RTD. RTD non conosce l'origine di tale route. Se si aggiunge l'istruzione aggregate as-set, si impone a RTC di generare le informazioni sul percorso sotto forma di set {}. Questo set include tutte le informazioni sul percorso, indipendentemente da quale percorso è arrivato per primo.




```
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
```

Caso 1:

RTC non dispone di un'istruzione as-set . RTC invia un aggiornamento 192.168.160.0/8 a RTD con informazioni sul percorso (300), come se la route fosse originata da 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.
```

Caso 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}.
```

I due soggetti successivi, BGP Confederation e Route Reflector, sono destinati ai provider di servizi Internet (ISP) che vogliono un ulteriore controllo dell'esplosione del peering iBGP all'interno dei loro AS.

Confederazione BGP

L'implementazione della confederazione BGP riduce la mesh iBGP all'interno di un AS. Il trucco consiste nel dividere un AS in più AS e assegnare l'intero gruppo a una singola confederazione. Ogni AS da solo dispone di iBGP completamente mesh e di connessioni ad altri AS all'interno della confederazione. Anche se questi AS hanno peer eBGP con AS all'interno della confederazione, gli AS scambiano il routing come se usassero iBGP. In questo modo, la confederazione conserva le informazioni di hop, metriche e preferenze locali. Per il mondo esterno, la confederazione sembra essere un singolo AS.

Per configurare una confederazione BGP, utilizzare questo comando:

```
<#root>
```

bgp confederation identifier <autonomous-system>

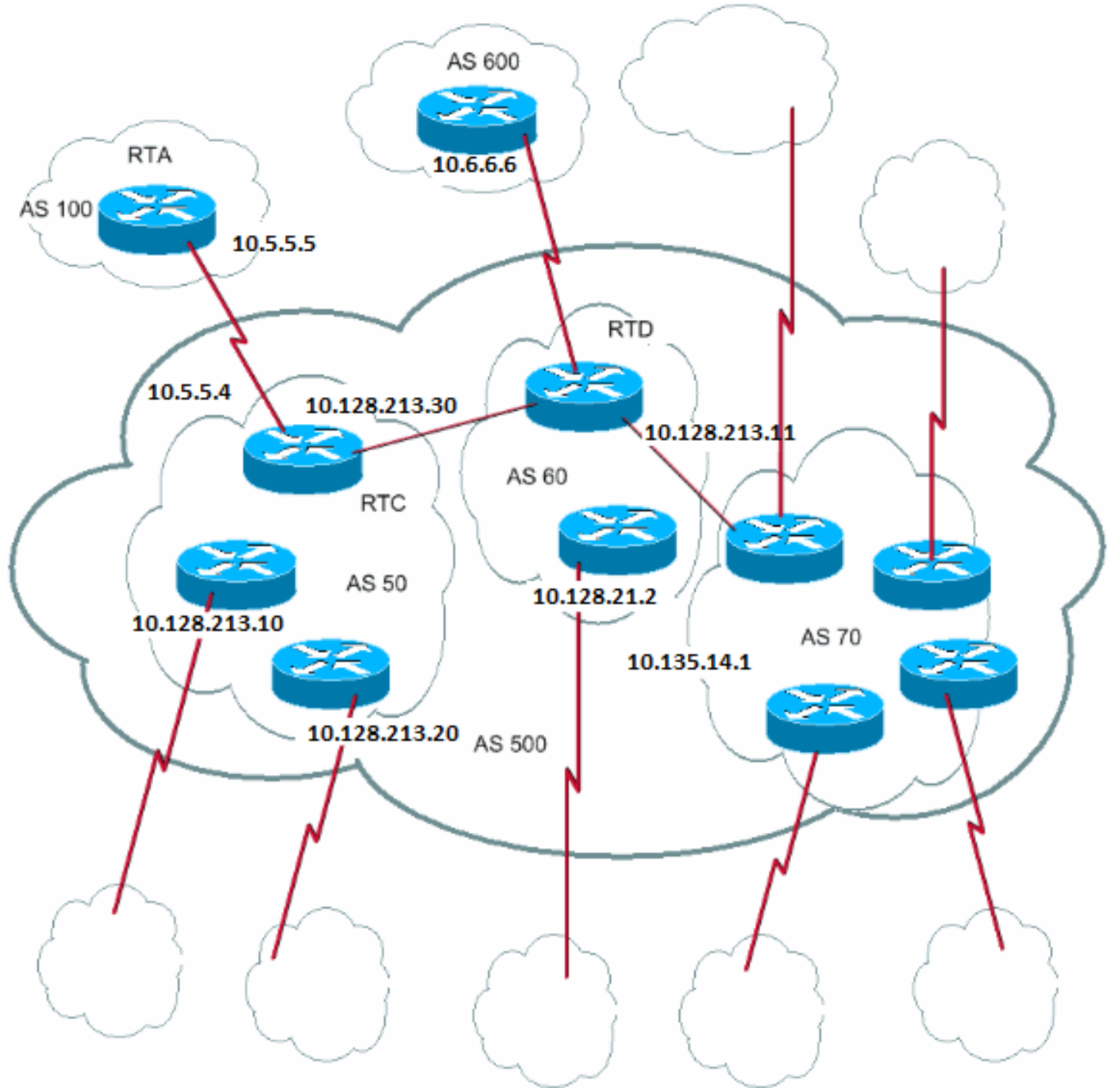
L'identificatore della confederazione è il numero AS del gruppo di confederazione.

L'attivazione di questo comando esegue il peering tra più AS all'interno della confederazione:

```
<#root>
```

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

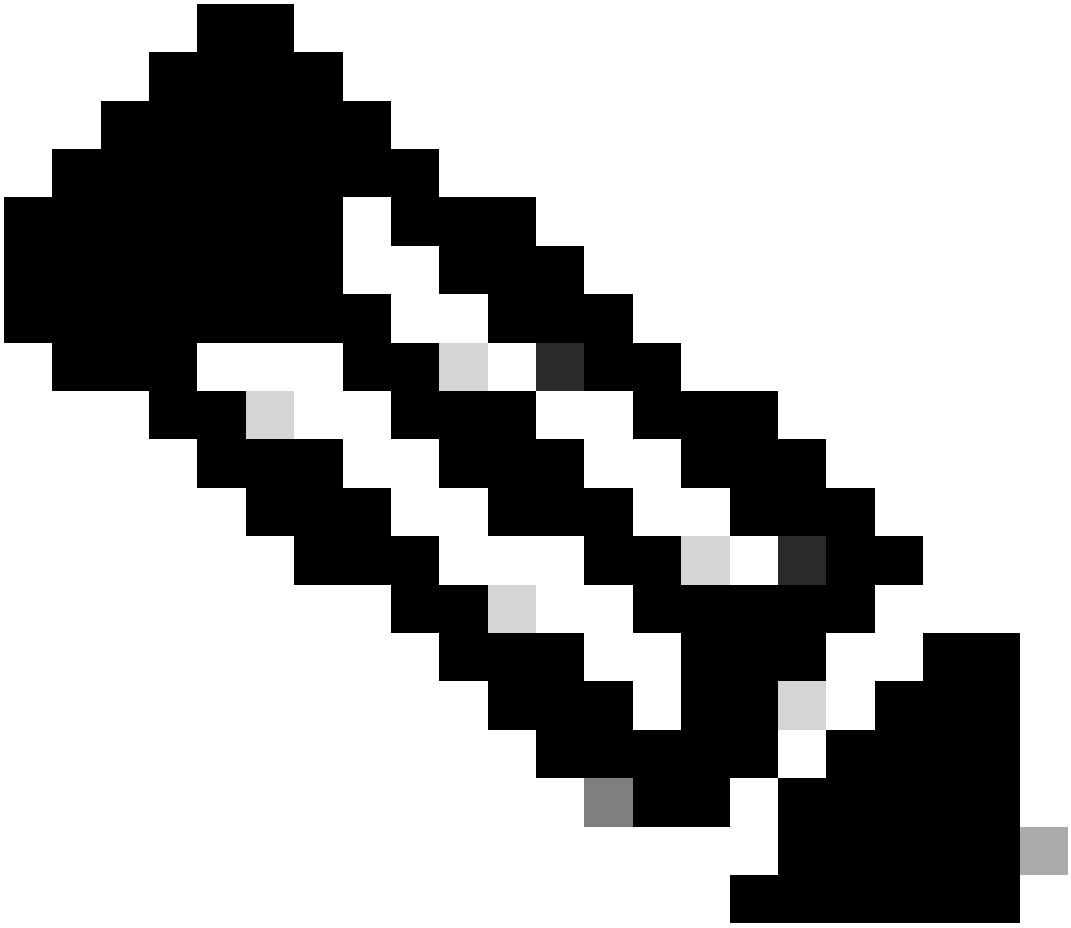
Segue un esempio di confederazione:



Si supponga di avere un AS500 composto da nove speaker BGP. Esistono anche altri speaker non BGP, ma si ha interesse solo per gli speaker BGP che hanno connessioni eBGP ad altri AS. Se si desidera creare una rete iBGP completa all'interno di AS500, sono necessarie nove connessioni peer per ciascun router. Sono necessari otto peer iBGP e un peer eBGP per AS esterni.

Se si utilizza la modalità confederazione, è possibile dividere AS500 in più AS50, AS60 e AS70. L'identificatore di confederazione AS è 500. Il mondo esterno vede una sola AS, AS500. Per ciascuno degli AS50, AS60 e AS70, si definisce una mesh completa di peer iBGP e si definisce l'elenco dei peer della confederazione con il comando **bgp confederation peers**.

Segue una configurazione di esempio dei router RTC, RTD e RTA:



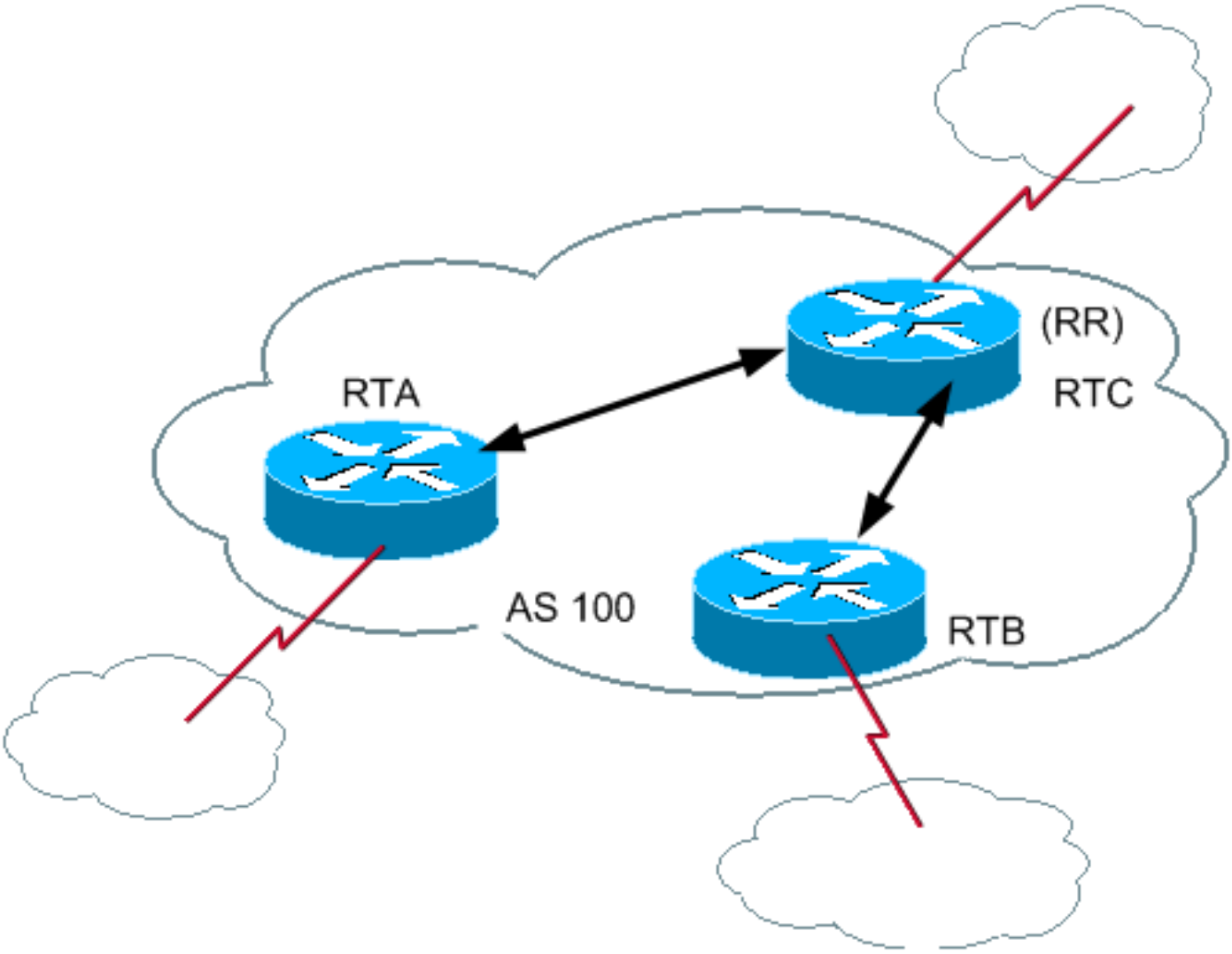
Nota: RTA non conosce AS50, AS60 o AS70. RTA conosce solo AS500.

```
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)
```

Route Reflector

Un'altra soluzione per l'esplosione del peering iBGP all'interno di un AS è Route Reflector (RR). Come dimostra la sezione iBGP, un altoparlante BGP non annuncia una route che il diffusore BGP ha appreso tramite un altro diffusore iBGP a un terzo diffusore iBGP. È possibile attenuare questa limitazione e fornire un controllo aggiuntivo, che consente a un router di pubblicizzare, o riflettere, i percorsi acquisiti da iBGP verso altri speaker iBGP. Questa route reflection riduce il numero di peer iBGP all'interno di un AS.



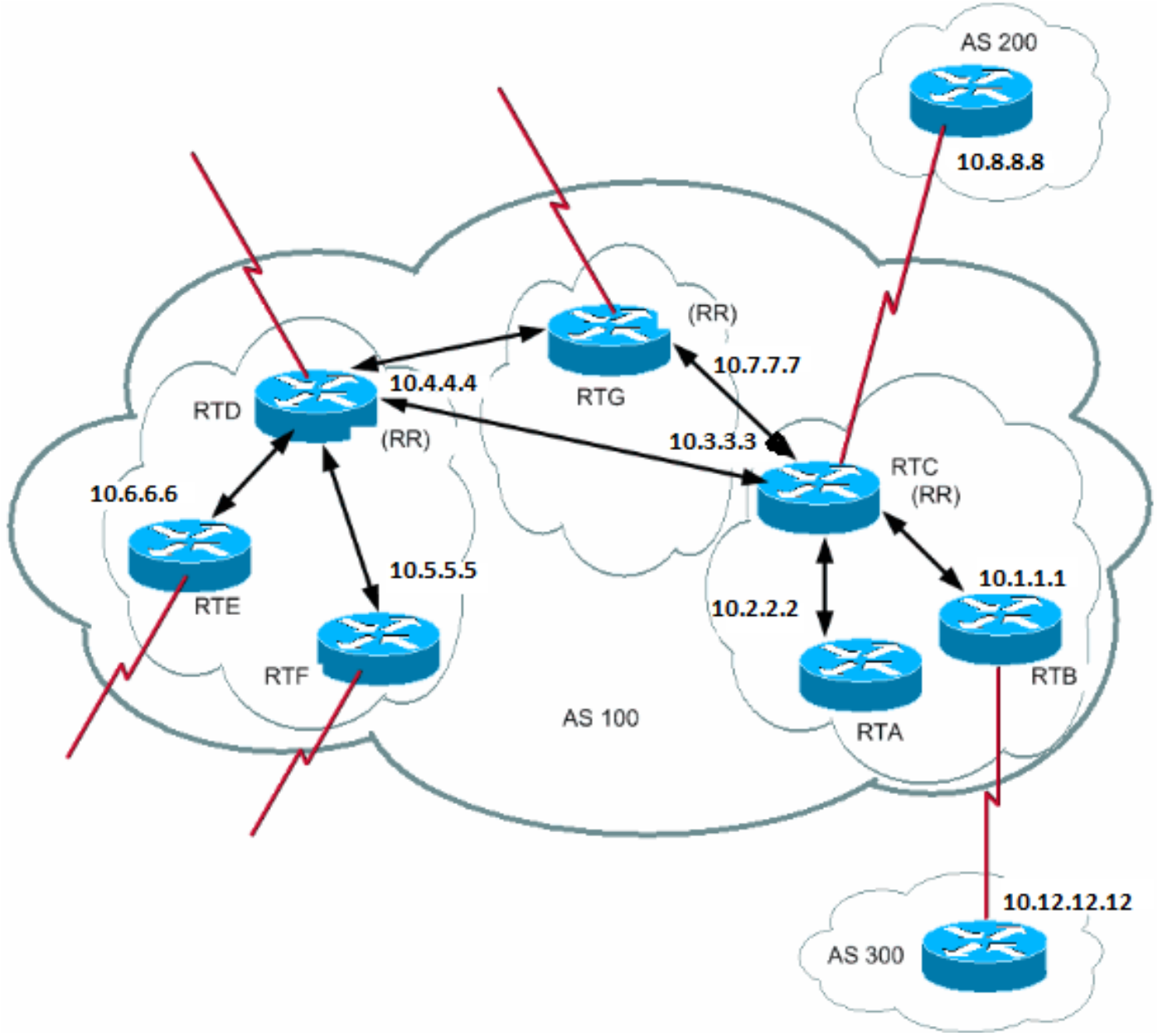
In casi normali, mantenere una mesh iBGP completa tra RTA, RTB e RTC entro AS100. Se si utilizza il concetto RR, RTC può essere selezionato come RR. In questo modo, RTC ha un peering iBGP parziale con RTA e RTB. Il peering tra RTA e RTB non è necessario perché RTC è un RR per gli aggiornamenti provenienti da RTA e RTB.

<#root>

neighbor <ip address> route-reflector-client

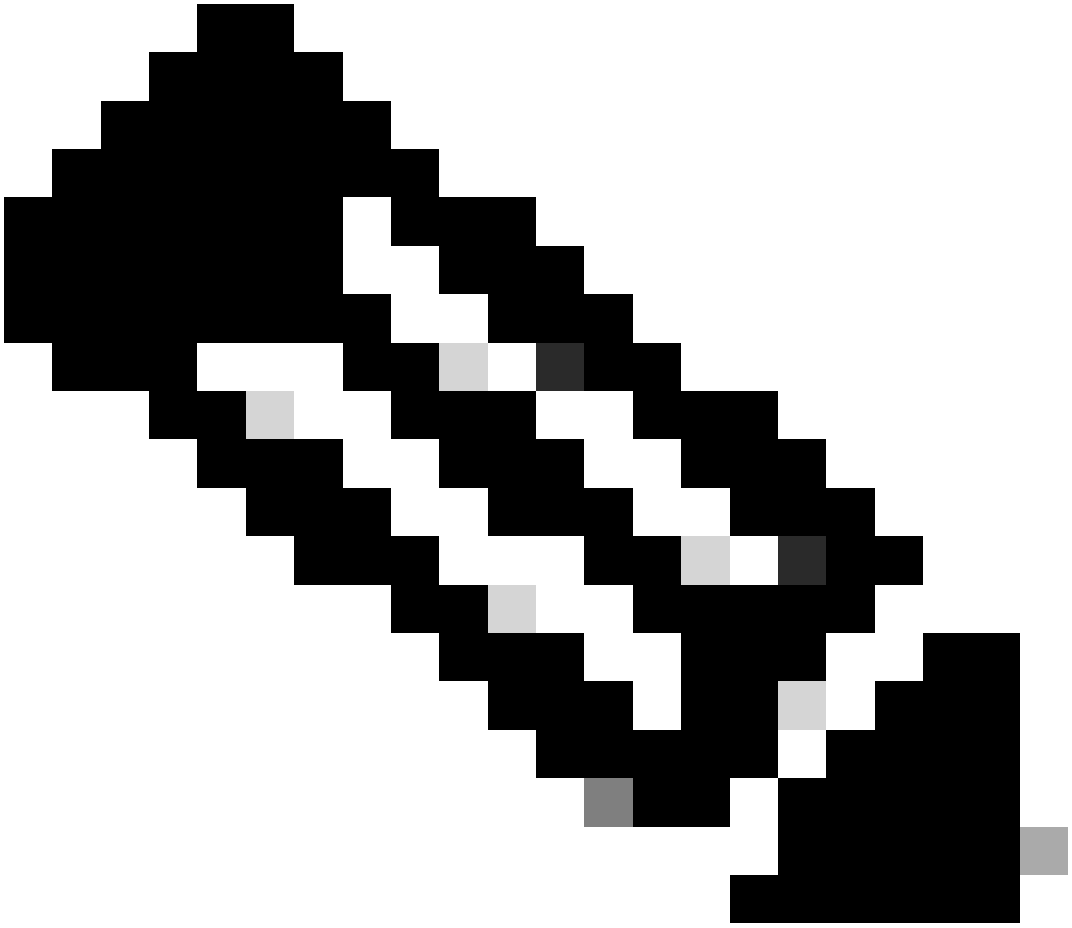
Il router con questo comando è l'RR e i neighbor a cui il comando punta sono i client dell'RR. Nell'esempio, il comando neighbor route-reflector-client della configurazione RTC punta agli indirizzi IP RTA e RTB. La combinazione di RR e client è un "cluster". In questo esempio, RTA, RTB e RTC formano un cluster con un singolo RR all'interno di AS100.

Altri peer iBGP del router che non sono client non sono client.



Un AS può avere più di un RR. In questa situazione, un RR tratta altri RR come qualsiasi altro altoparlante iBGP. Altri RR possono appartenere allo stesso cluster (gruppo di client) o ad altri cluster. In una configurazione semplice, è possibile dividere l'AS in più cluster. Ogni RR viene configurato con altri RR come peer non client in una topologia con mesh completa. I client non devono eseguire il peer con altoparlanti iBGP esterni al cluster di client.

Nel diagramma precedente, RTA, RTB e RTC formano un singolo cluster. RTC è l'RR. Per RTC, RTA e RTB sono client e qualsiasi altra cosa è non client. Si ricordi che il comando neighbor route-reflector-client punta ai client di un RR. Lo stesso RTD è l'RR per i client RTE e RTF. RTG è un RR in un terzo cluster.



Nota: RTD, RTC e RTG sono collegati completamente, a differenza dei router all'interno di un cluster.

Quando un RR riceve una route, vengono visualizzate le route RR come indicato nell'elenco. Tuttavia, questa attività dipende dal tipo di peer:

- Route da un peer non client: riflette su tutti i client all'interno del cluster.
- Route da un peer client: riflette su tutti i peer non client e anche sui peer client.

- Route da un peer eBGP: invia l'aggiornamento a tutti i peer client e non client.

Questa è la configurazione del BGP dei router RTC, RTD e RTB:

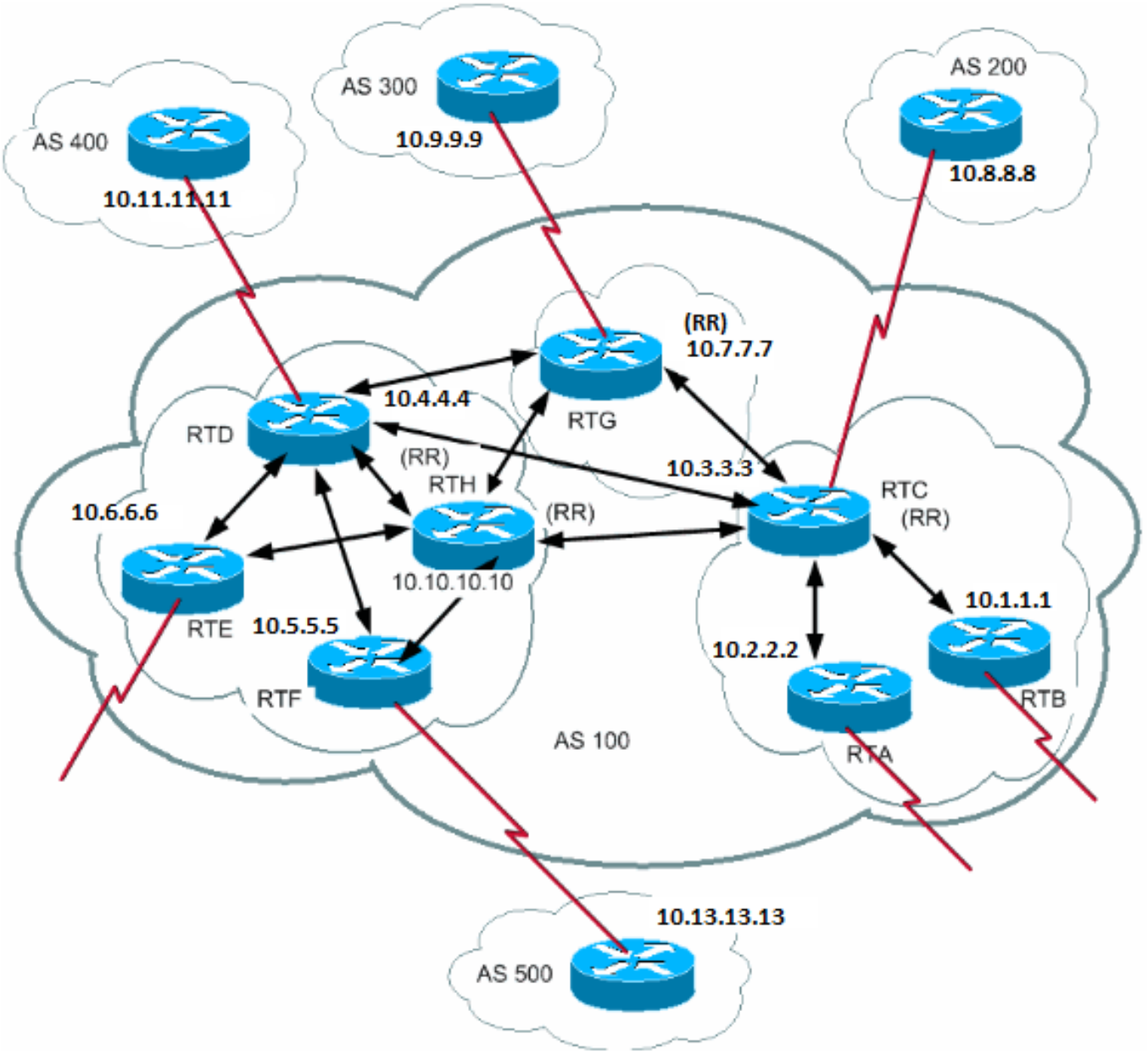
```
RTC#
router bgp 100
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 route-reflector-client
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.7.7.7 remote-as 100
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.8.8.8 remote-as 200

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
```

Poiché esiste una route reflection degli indirizzamenti acquisiti in iBGP, può verificarsi un loop di informazioni di routing. Lo schema RR presenta alcuni metodi per evitare questo loop:

- **originator-id**: si tratta di un attributo BGP opzionale, non transitivo, lungo 4 byte. Un RR crea questo attributo. L'attributo trasporta l'ID router (RID) del mittente della route nell'AS locale. Se, a causa di una configurazione errata, le informazioni di routing vengono restituite al mittente, le informazioni vengono ignorate.

- **cluster-list**: la sezione Più record di risorse all'interno di un cluster copre l'elenco dei cluster.

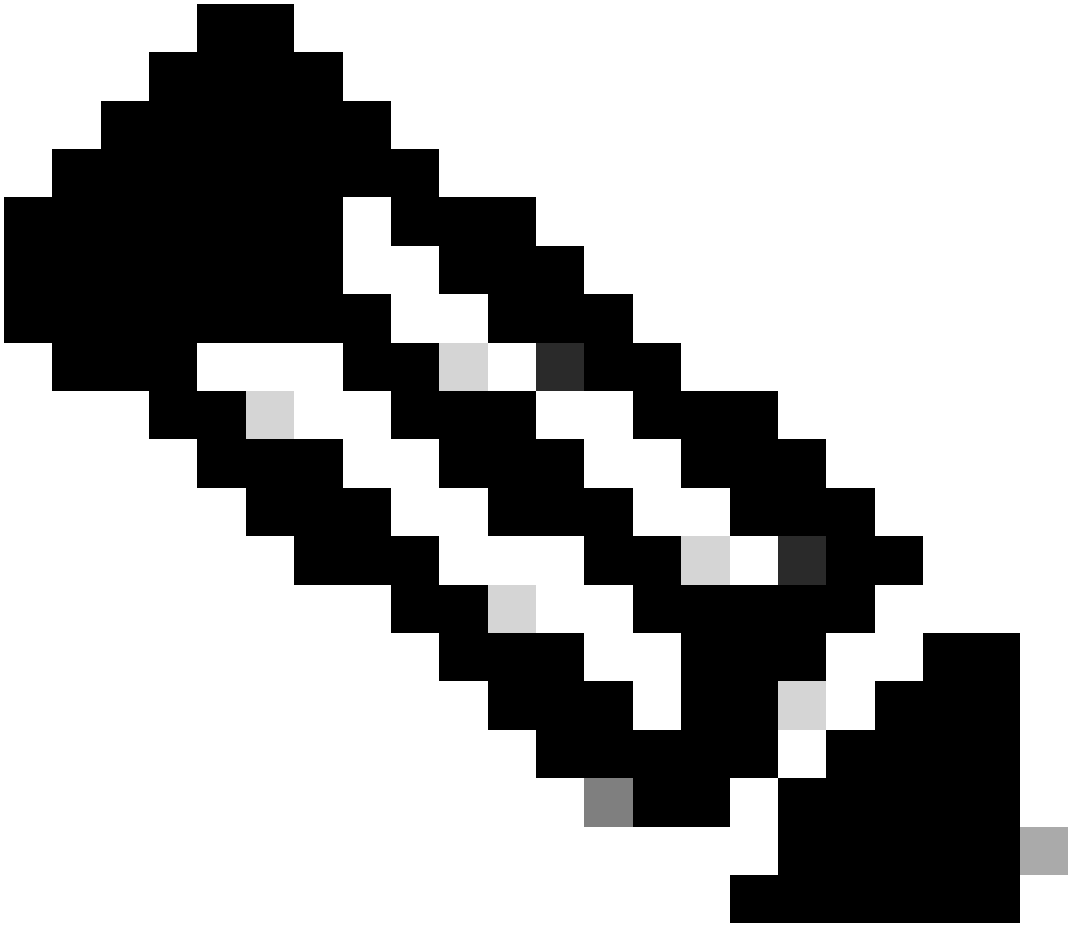
Più RR all'interno di un cluster



In genere, un cluster di client ha un singolo RR. In questo caso, l'ID del router dell'RR identifica il cluster. Per aumentare la ridondanza ed evitare singoli punti di errore, un cluster può avere più di un RR. È necessario configurare tutti gli RR nello stesso cluster con un ID cluster a 4 byte in modo che un RR possa riconoscere gli aggiornamenti dagli RR dello stesso cluster.

Un elenco di cluster è una sequenza di ID cluster superati dalla route. Quando un RR riflette una route dai client RR ai non client all'esterno del cluster, l'RR aggiunge l'ID del cluster locale all'elenco dei cluster. Se questo aggiornamento dispone di un elenco di cluster vuoto, l'RR ne crea uno. Con questo attributo, un RR può identificare se le informazioni di routing sono tornate allo stesso cluster a causa di una configurazione errata. Se l'ID cluster locale viene trovato nell'elenco dei cluster, la pubblicizzazione viene ignorata.

Nel diagramma di questa sezione, RTD, RTE, RTF e RTH appartengono a un cluster. Sia RTD che RTH sono RR per lo stesso cluster.

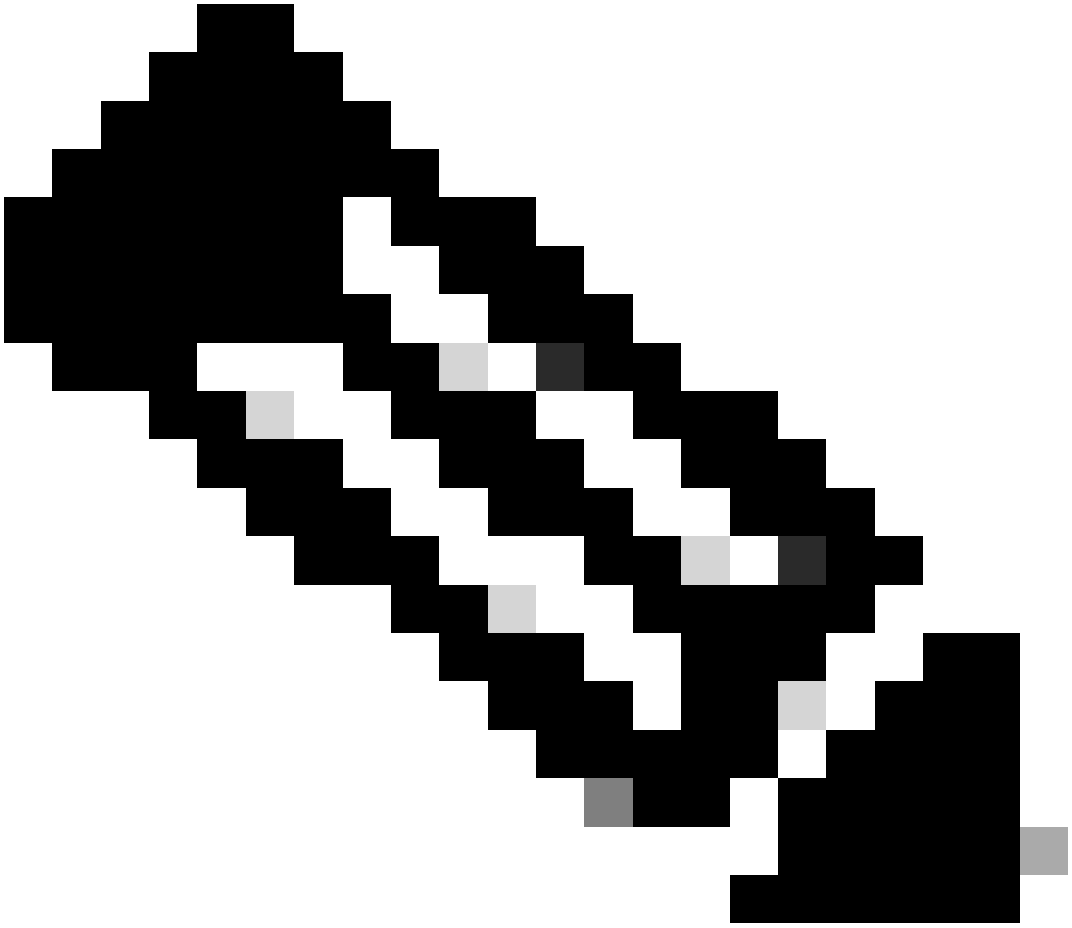


Nota: la ridondanza è dovuta al fatto che RTH ha eseguito il meshing del peering con tutti gli RR. Se RTD diminuisce, RTH sostituisce RTD.

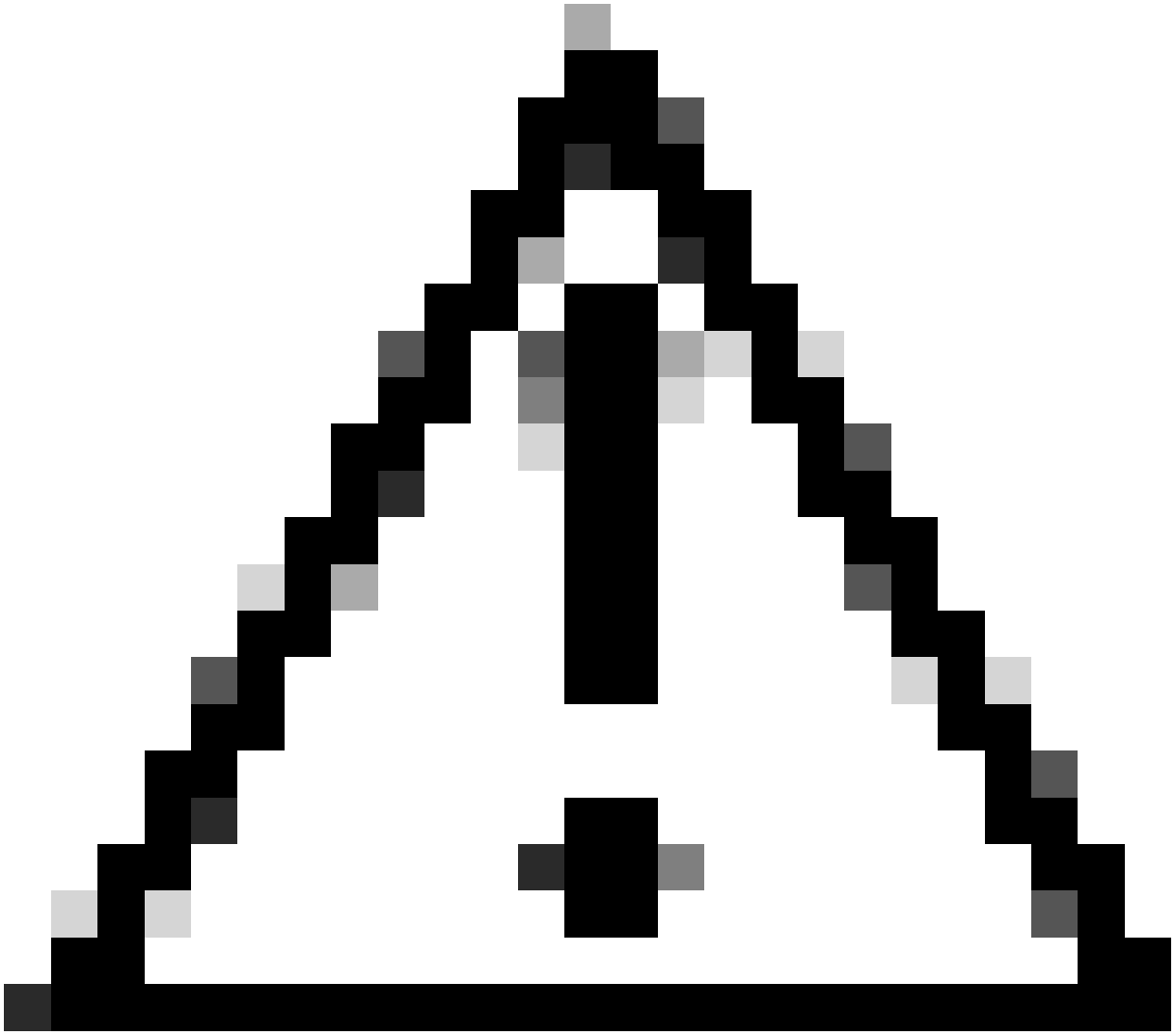
Questa è la configurazione di RTH, RTD, RTF e RTC:

```
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
```



Nota: il comando bgp cluster-id per RTC non è necessario, in quanto in quel cluster esiste un solo RR.



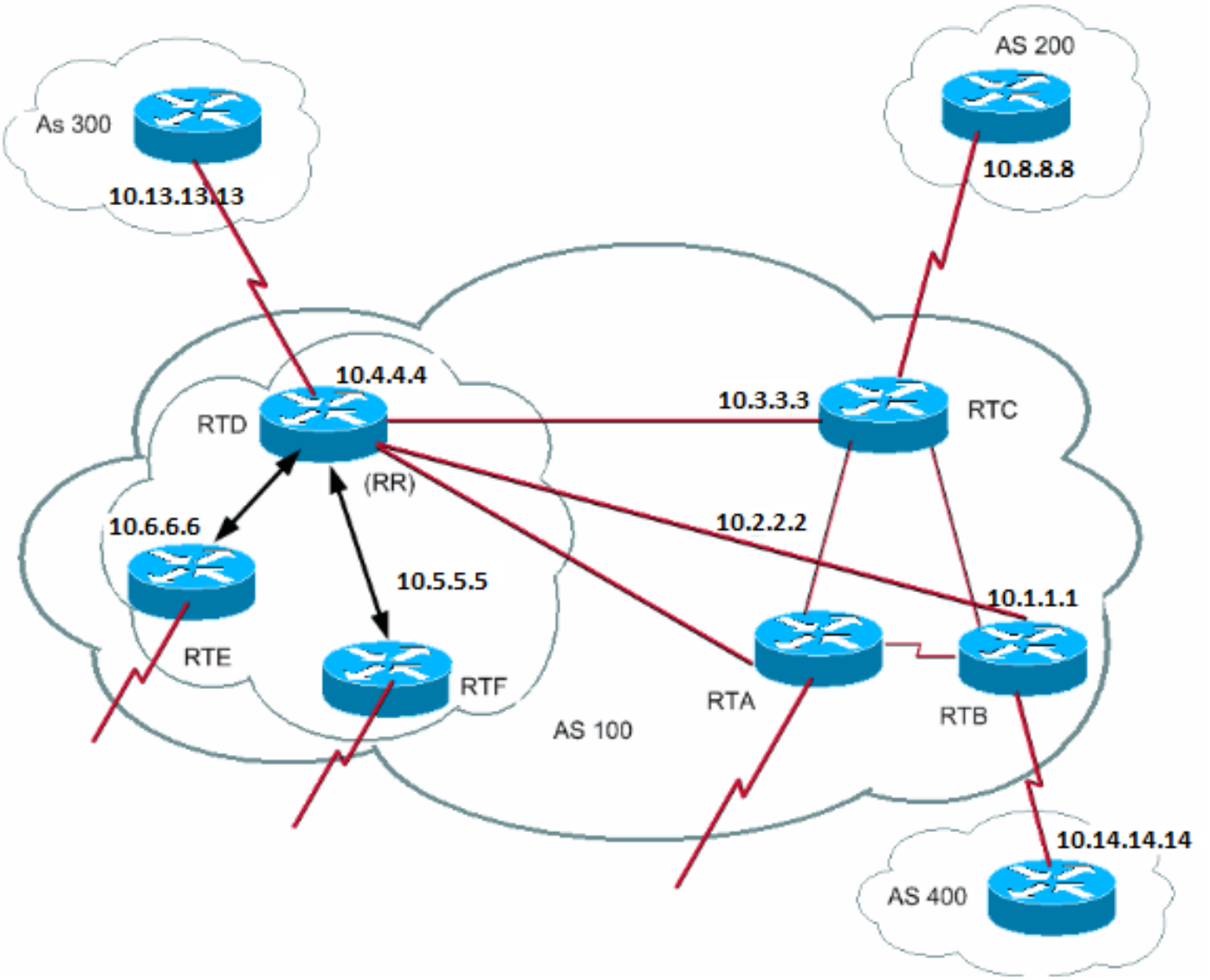
Attenzione: questa configurazione non utilizza gruppi peer. Non utilizzare i gruppi di peer se i client all'interno di un cluster non hanno peer iBGP diretti tra loro e i client si scambiano aggiornamenti tramite RR. Se si configurano gruppi di peer, un potenziale prelievo all'origine di una route su RR viene trasmesso a tutti i client all'interno del cluster. Questa trasmissione può causare problemi.

Il sottocomando **bgp client-to-client reflection** **del router è abilitato per impostazione predefinita su RR.** Se si disattiva la reflection BGP client-to-client sull'RR e si rende il peering BGP ridondante tra i client, è possibile utilizzare in modo sicuro i gruppi di peer. Per ulteriori informazioni, consultare Limitazioni dei gruppi di peer.

RR e speaker BGP convenzionali

Un AS può avere speaker BGP che non comprendono la funzione RR. Questo documento chiama questi router speaker BGP tradizionali. Lo

schema RR consente a questi speaker BGP convenzionali di coesistere. Questi router possono essere membri di un gruppo client o di un gruppo non client. L'esistenza di questi router consente una migrazione semplice e graduale dal modello iBGP corrente al modello RR. È possibile iniziare a creare cluster configurando un singolo router come RR e rendendo altri peer iBGP RR e client RR normali. Quindi, è possibile creare più cluster gradualmente.



In questo diagramma, RTD, RTE e RTF dispongono della funzione di route reflection. RTC, RTA e RTB sono router convenzionali. Non è possibile configurare questi router come RR. È possibile eseguire una normale mesh iBGP tra questi router e RTD. In seguito, quando si è pronti per l'aggiornamento, è possibile impostare RTC come RR con i client RTA e RTB. I client non devono necessariamente comprendere lo schema di riflessione delle route; solo le RR richiedono l'aggiornamento.

Questa è la configurazione di RTD e 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
```

Quando si è pronti per aggiornare RTC e rendere RTC un RR, rimuovere l'iBGP full mesh e fare in modo che RTA e RTB diventino client di RTC.

Evitare il loop di informazioni di routing

Finora nel documento sono stati menzionati due attributi che è possibile utilizzare per impedire potenziali cicli di informazioni: **originator-id** e **cluster-list**.

Un altro modo per controllare i loop consiste nell'imporre più restrizioni alla clausola set **delle route map in uscita.** La clausola set per le route map in uscita non influisce sulle route che si riferiscono ai peer iBGP.

È inoltre possibile applicare ulteriori restrizioni all'**hop successivo-self**, che è un'opzione di configurazione per router adiacenti. Quando si utilizza **next-hop-**selfo in RR, la clausola influisce solo sull'hop successivo delle route acquisite da eBGP, in quanto l'hop successivo delle route riflesse non deve essere modificato.

Flap dampening sulla route

Il software Cisco IOS versione 11.0 ha introdotto il route dampening. Il route dampening è un meccanismo per ridurre al minimo l'instabilità causata dal flapping. Il dampening riduce anche le oscillazioni sulla rete. Si definiscono i criteri per identificare le route con un comportamento anomalo. Una route che esegue il flapping ottiene una penalità pari a 1000 per ogni disattivazione/riattivazione. Non appena la penalità cumulativa raggiunge un limite di soppressione predefinito, si verifica la soppressione dell'annuncio route. La penalità diminuisce in modo esponenziale in base a un tempo di emivita preconfigurato. Una volta ridotta la penalità in base a un limite di riutilizzo predefinito, l'annuncio route non viene più eliminato.

Il dampening non si applica alle route esterne a un AS e acquisite tramite iBGP. In questo modo, il dampening della route evita una penalità più elevata per i peer iBGP in caso di route esterne all'AS.

La penalità si riduce con una granularità di 5 secondi. Le route non vengono eliminate con una granularità di 10 secondi. Il router conserva le informazioni di smorzamento finché la penalità non diventa inferiore alla metà del limite di riutilizzo. A quel punto, il router elimina le informazioni.

Inizialmente, il dampening è disattivato per impostazione predefinita. In caso di necessità, questa funzione può essere abilitata per impostazione predefinita in futuro. Questi comandi controllano il route dampening:

- **smorzamento bgp**: attiva lo smorzamento.

- **no smorzamento bgp** - Disattiva lo smorzamento.

- **bgp dampeninghalf-life-time**: modifica il tempo di emivita.

Un comando che imposta tutti i parametri contemporaneamente è:

- **smorzamento bgp half-life-timereusesoppressmaximum-suppress-time**

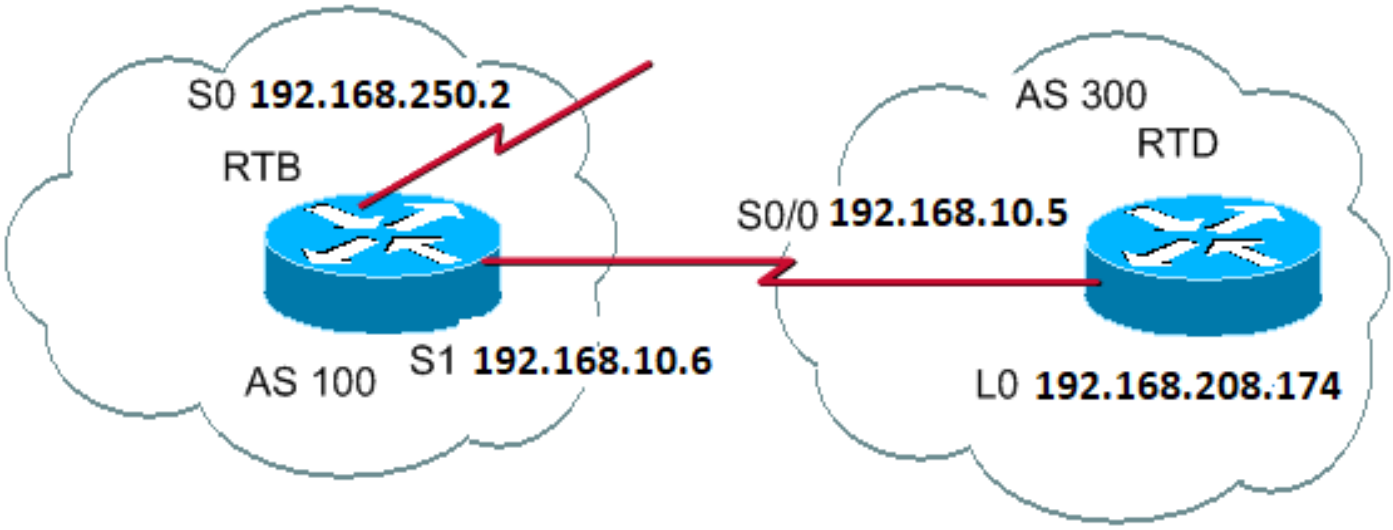
In questo elenco viene descritta la sintassi:

- **half-life-time**: l'intervallo è compreso tra 1 e 45 minuti e l'impostazione predefinita corrente è 15 minuti.

- **reuse-value**: l'intervallo è compreso tra 1 e 20.000 e il valore predefinito è 750.

- **suppress-value**: l'intervallo è compreso tra 1 e 20.000 e il valore predefinito è 2000.

- **max-suppress-time**: durata massima della soppressione di una route. L'intervallo è compreso tra 1 e 255 minuti e il valore predefinito è 4 volte il tempo di emivita.




```
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
```

L'RTB è configurato per il dampening della route con parametri predefiniti. Se si presuppone che il collegamento eBGP a RTD sia stabile, la tabella BGP RTB è simile alla seguente:

<#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
```

Per simulare un flap di indirizzamento, utilizzare il comando clear ip bgp 192.168.10.6 su RTD. La tabella BGP RTB è simile alla seguente:

<#root>

```
RTB#
```

show ip bgp

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

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

La voce BGP per 192.168.10.0 è in uno stato storico. Questo significa che non si dispone di un percorso migliore per la route, ma esistono ancora informazioni sul flapping.

<#root>

```
RTB#
```

show ip bgp 192.168.10.0

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

La rotta ha ricevuto una sanzione per il flapping, ma la sanzione è ancora al di sotto del limite di soppressione. Il valore predefinito è 2000.

Eliminazione della route non ancora eseguita. Se la route si blocca ancora, viene visualizzato:

```
    <#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
```

Il percorso è stato smorzato o soppresso. La route viene riutilizzata quando la sanzione raggiunge il "valore di riutilizzo". In questo caso, il valore di riutilizzo è quello predefinito, ovvero 750. Le informazioni di smorzamento vengono eliminate quando la penalità diventa inferiore alla metà del limite di riutilizzo. In questo caso, l'eliminazione avviene quando la sanzione diventa 375 (750/2 = 375). Questi comandi mostrano e cancellano le informazioni sulle statistiche flapping:

- **show ip bgp flap-statistics**: visualizza le statistiche di flap per tutti i percorsi.
- **show ip bgp flap-statistics regexpregular-expression**: visualizza le statistiche di flap per tutti i percorsi che corrispondono all'espressione regolare.
- **show ip bgp flap-statistics filter-listlist**: visualizza le statistiche di flap per tutti i percorsi che passano il filtro.
- **show ip bgp flap-statisticsA.B.C.D m.m.m.m**— Visualizza le statistiche di flap per una singola voce.
- **show ip bgp flap-statisticsA.B.C.D m.m.m.mlong-prefix**: visualizza le statistiche di flap per voci più specifiche.
- **show ip bgp neighbor [dampened-routes] | [flap-statistics]**: visualizza le statistiche di flap per tutti i percorsi di un router adiacente.
- **clear ip bgp flap-statistics**: cancella le statistiche di flap per tutte le route.
-

clear ip bgp flap-statistics regexpregular-expression: cancella le statistiche di flap per tutti i percorsi che corrispondono all'espressione regolare.

- **clear ip bgp flap-statistics filter-listlist**: cancella le statistiche di flap per tutti i percorsi che passano il filtro.

- **clear ip bgp flap-statisticsA.B.C.D m.m.m.m**— Cancella le statistiche di flap per una singola voce.

- **clear ip bgpA.B.C.Dflap-statistics**: cancella le statistiche di flap per tutti i percorsi di un router adiacente.

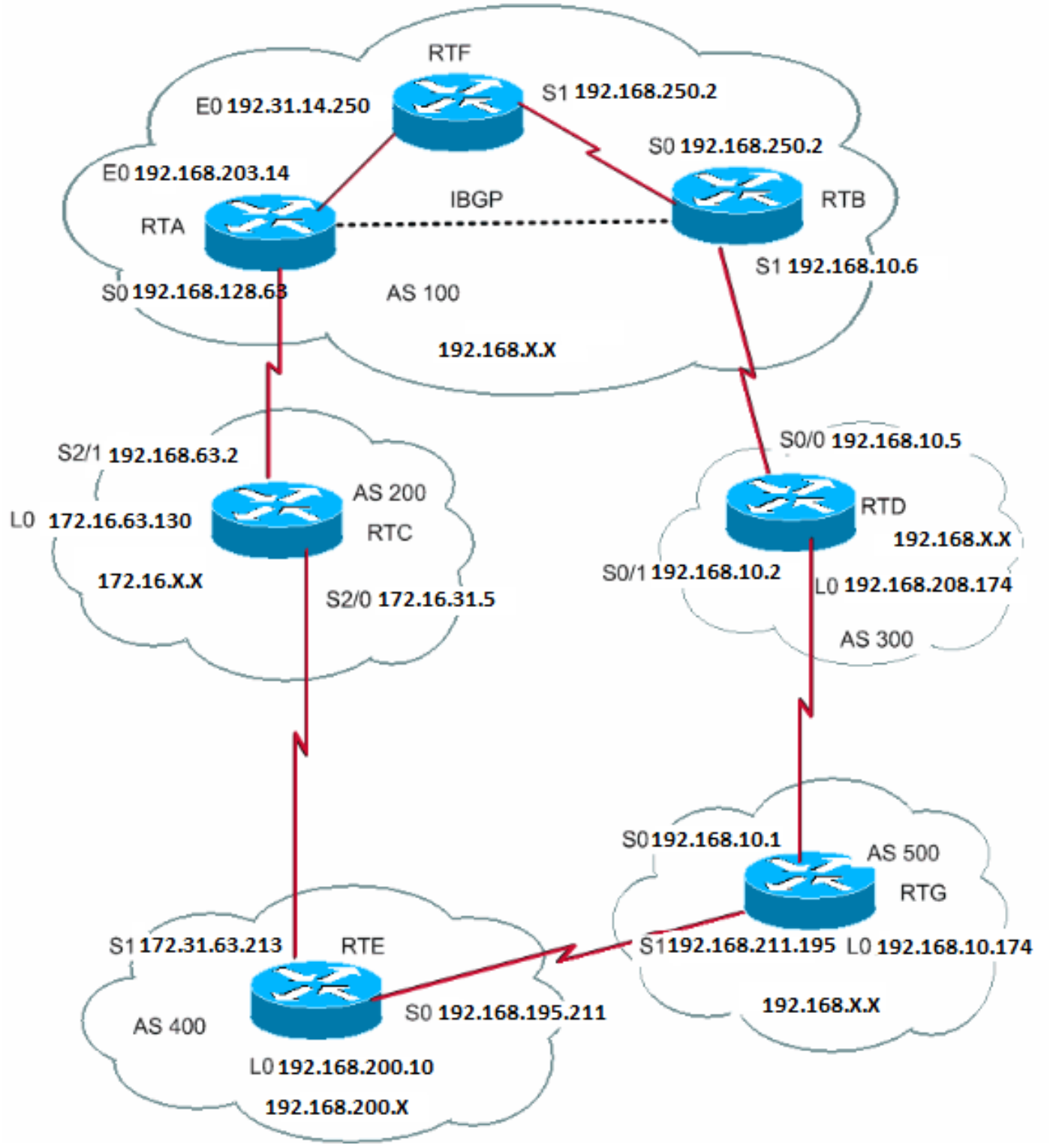
Modalità di selezione del percorso BGP

Ora che si conoscono gli attributi e la terminologia di BGP, fare riferimento a Algoritmo di selezione del miglior percorso BGP.

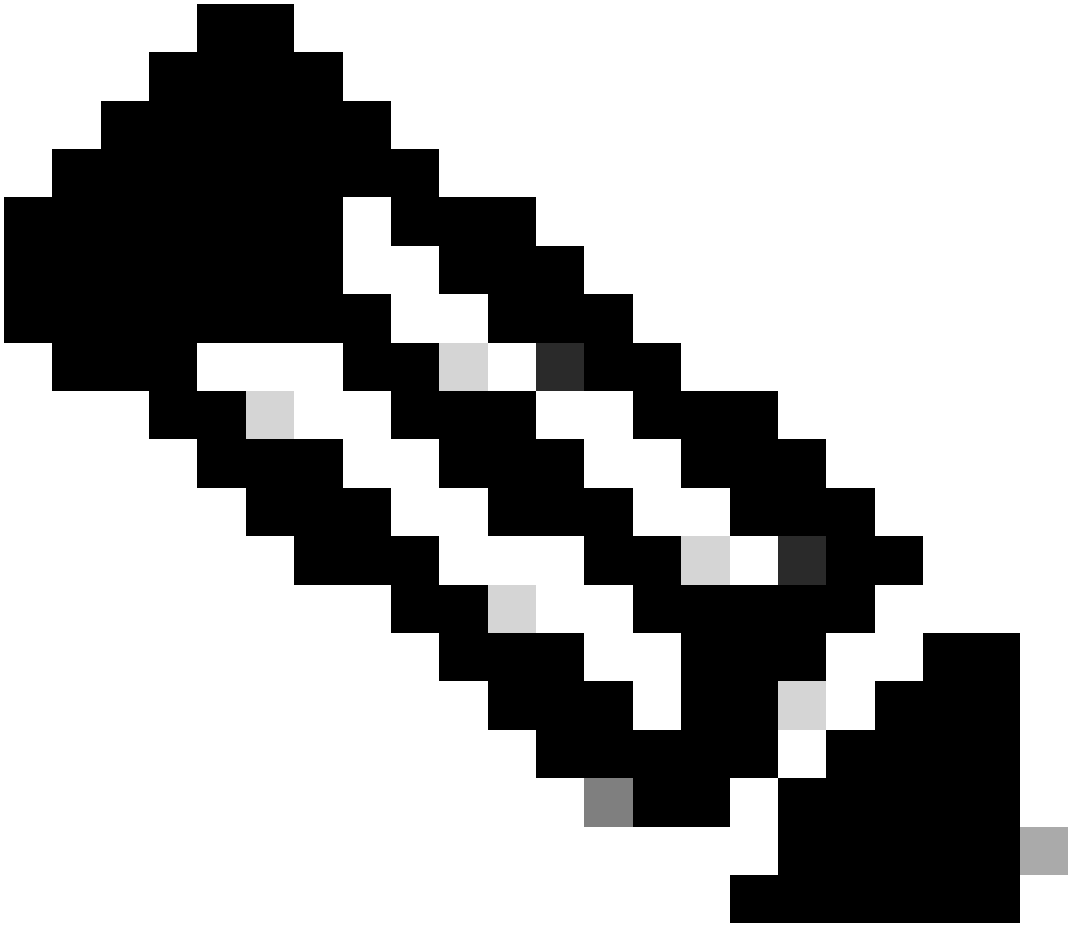
Case study BGP 5

Esempio pratico di progettazione

Questa sezione contiene un esempio di progettazione che mostra le tabelle di configurazione e di routing così come vengono effettivamente visualizzate sui router Cisco.



Questa sezione mostra passo-passo come creare questa configurazione e cosa può andare storto. Ogni volta che si dispone di un AS che si connette a due ISP tramite eBGP, eseguire sempre iBGP all'interno dell'AS per avere un migliore controllo delle route. In questo esempio, iBGP viene eseguito all'interno di AS100 tra RTA e RTB e OSPF viene eseguito come IGP. Si supponga di connettersi a due ISP, AS200 e AS300. Questa è la prima esecuzione delle configurazioni per tutti i router:



Nota: queste configurazioni non corrispondono alle configurazioni finali.

```
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
```

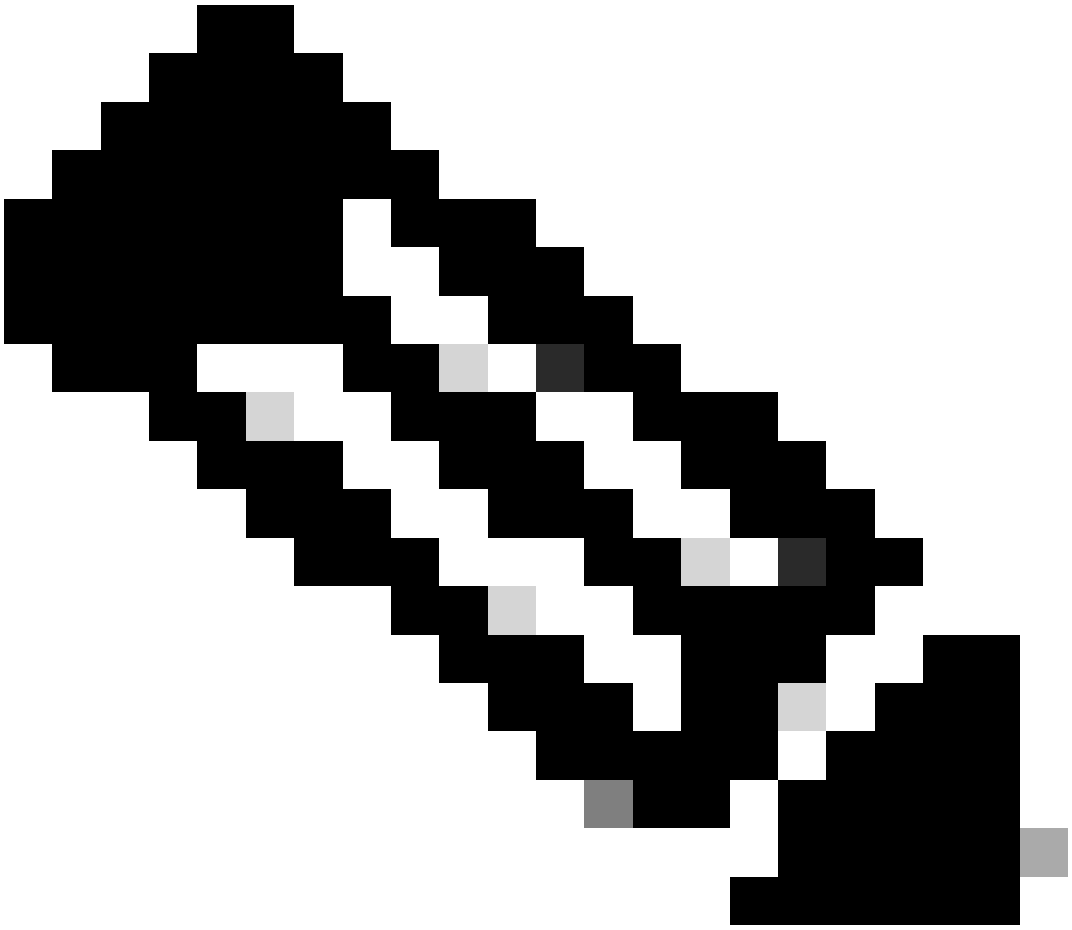
Utilizzare sempre il network comando o ridistribuire le voci statiche in BGP per annunciare le reti. Questo metodo è migliore di una ridistribuzione di IGP in BGP. In questo esempio viene utilizzato il network comando per inserire reti in BGP.

Qui, si inizia con l'interfaccia s1 all'arresto di RTB, come se il collegamento tra RTB e RTD non esistesse. Questa è la tabella RTB BGP:

```
<#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
```

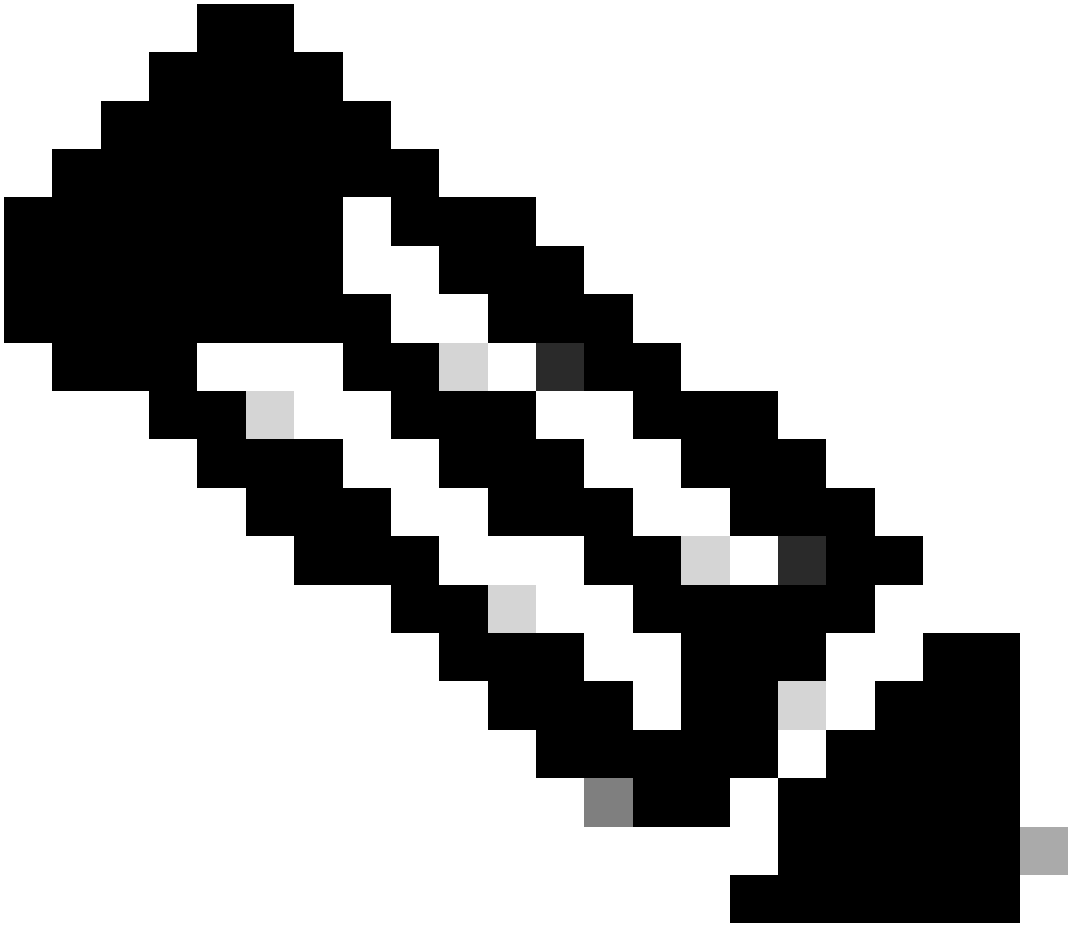
In questa tabella vengono visualizzate le seguenti annotazioni:

- Ani all'inizio (Aniat the begin) - Indica che la voce è stata appresa tramite un peer iBGP.
- Anita alla fine (Aniat the end) - Indica che l'origine delle informazioni sul percorso è IGP.
- Informazioni sui percorsi: queste informazioni sono intuitive. Ad esempio, la rete 172.31.10.0 viene acquisita tramite il percorso 200 con un next-hop di 172.31.63.250.



Nota: qualsiasi voce generata localmente, come ad esempio 192.168.250.15, ha next-hop 0.0.0.0.

- Un simbolo > : indica che BGP ha scelto la route migliore. Il BGP utilizza i passaggi decisionali descritti nel documento Algoritmo di selezione del miglior percorso BGP. Il BGP sceglie il percorso migliore per raggiungere una destinazione, installa il percorso nella tabella di routing IP e lo pubblicizza ad altri peer BGP.



Nota: osservare l'attributo Next Hop . RTB conosce 172.31.10.0 tramite un next-hop di 172.31.63.250, che è il next-hop eBGP trasportato in iBGP.

Osservando la tabella di routing IP:

<#root>

RTB#

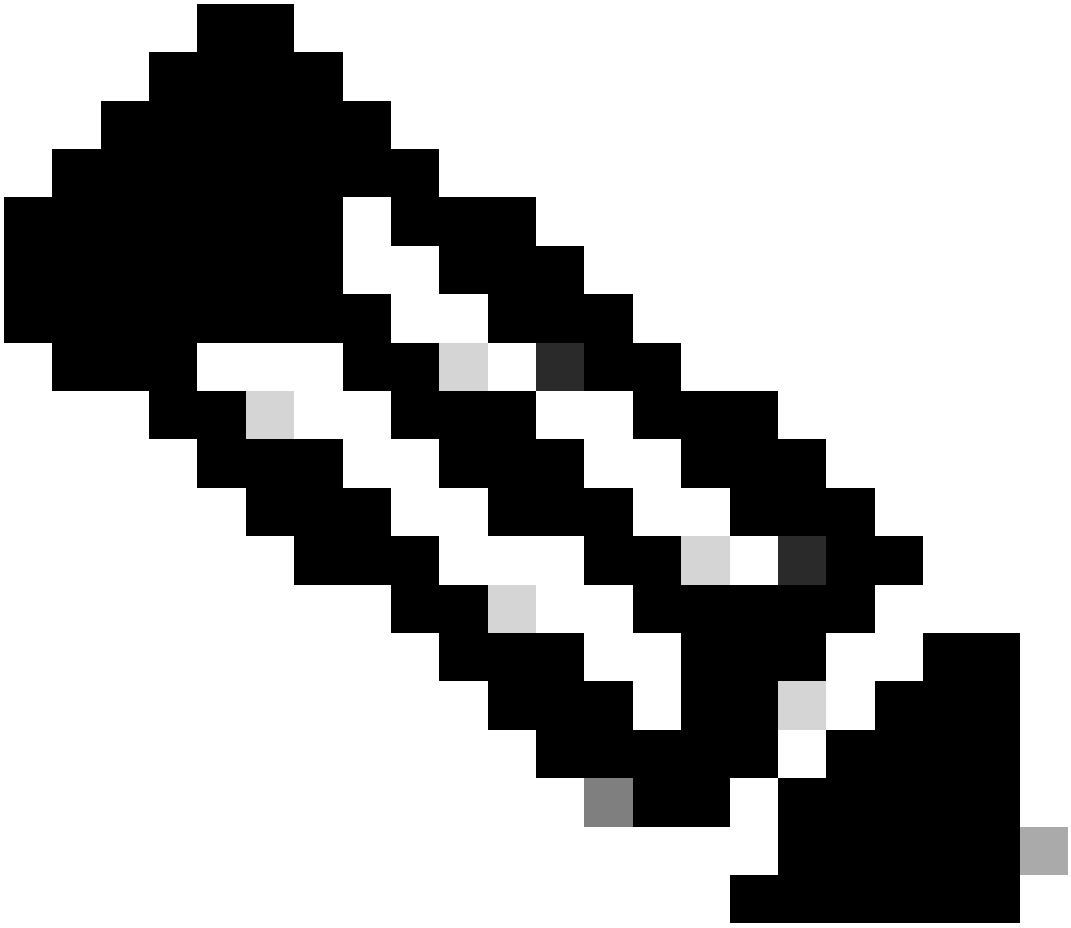
show ip route

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

Gateway of last resort is not set

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

Apparentemente, nessuna delle voci BGP ha raggiunto la tabella di routing. Esistono due problemi.

Il primo problema è che il next-hop per queste voci, 172.31.63.250, è irraggiungibile. Non c'è modo di raggiungere il next-hop tramite questo IGP, che è OSPF. RTB non ha appreso informazioni su 192.168.213.63 tramite OSPF. È possibile eseguire OSPF sull'interfaccia RTA s0 e renderla passiva; in questo modo, RTB può raggiungere l'hop successivo 172.31.63.250. La configurazione dell'aggregazione in tempo reale viene visualizzata qui:

```
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
```



Nota: è possibile usare il comandobgp nexthop self tra RTA e RTB per modificare l'hop successivo.

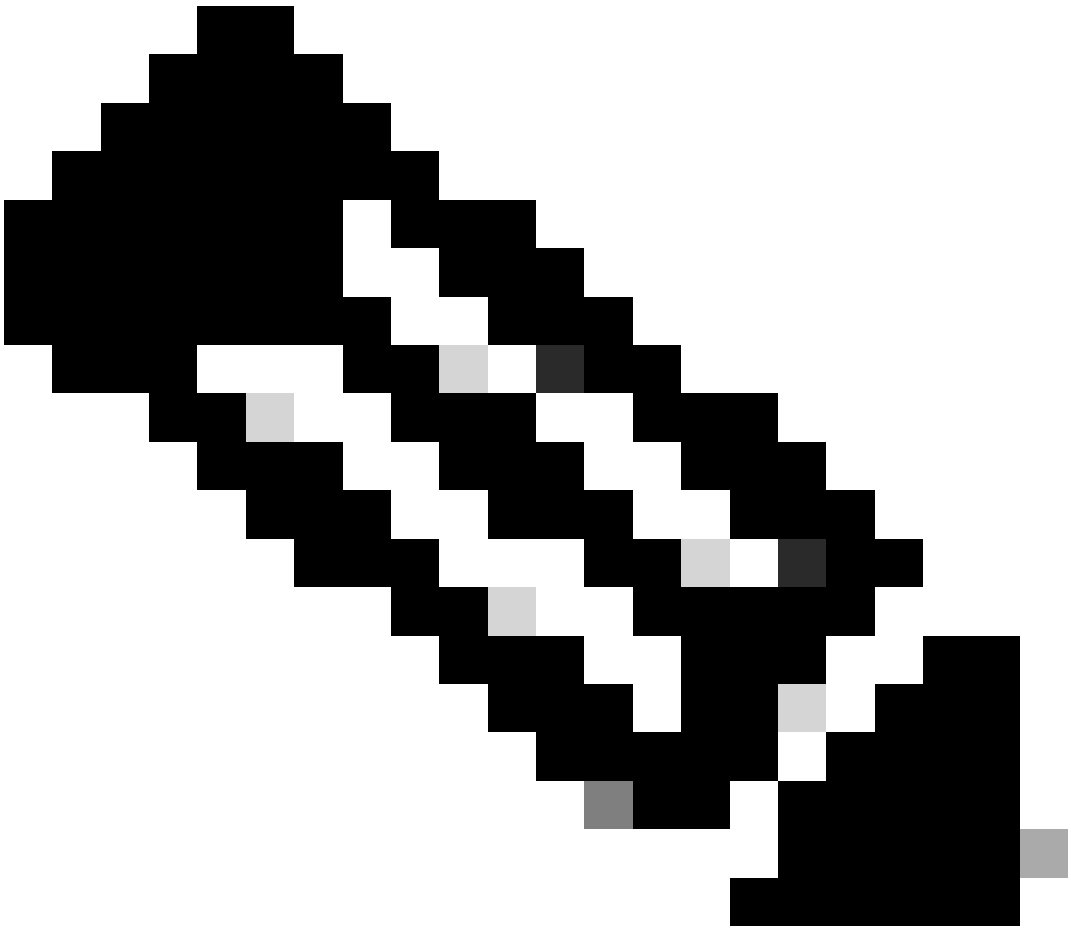
La nuova tabella BGP su RTB è simile alla seguente:

<#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
```



Nota: tutte le voci contengono >, il che significa che il BGP può raggiungere il next-hop.

Osservare la tabella di routing:

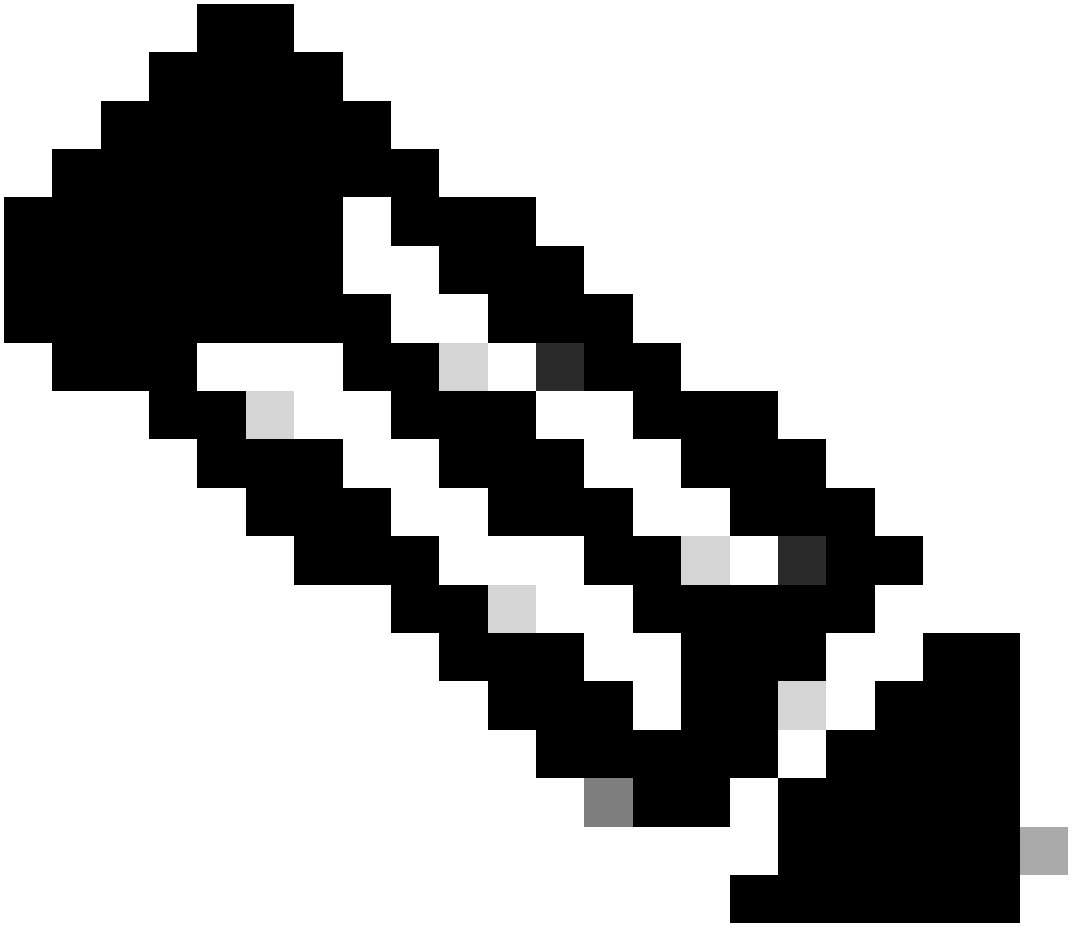
<#root>

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
```

Il secondo problema è che le voci BGP non sono ancora visualizzate nella tabella di routing. L'unica differenza è che 192.168.213.63 è ora raggiungibile tramite OSPF. Questo è un problema di sincronizzazione. BGP non inserisce queste voci nella tabella di routing e non invia le voci negli aggiornamenti BGP a causa della mancanza di sincronizzazione con IGP.



Nota: RTF non ha la nozione di reti 192.168.10.0 e 192.168.211.10 perché non è stato ancora ridistribuito BGP in OSPF.

In questo scenario, se si disattiva la sincronizzazione, le voci vengono visualizzate nella tabella di routing. Ma la connettività è ancora interrotta.

Se si disattiva la sincronizzazione su RTB, ecco cosa succede:

```
<#root>

RTB#
```

show ip route

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

Gateway of last resort is not set

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

La tabella di routing è corretta, ma non è possibile raggiungere le reti. RTF centrale non sa come raggiungere le reti:

```
   <#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
```

Quando si disattiva la sincronizzazione in questa situazione, il problema persiste. Tuttavia, la sincronizzazione servirà in seguito per altri problemi. Ridistribuire il BGP in OSPF su RTA con una metrica 2000:

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 192.168.203.250 255.255.255.0

interface Ethernet0
ip address 192.168.203.14 255.255.255.0

interface Serial0
 ip address 192.168.128.63 255.255.255.252

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

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

La tabella di routing è simile alla seguente:

```
    <#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
```

Le voci BGP sono scomparse perché OSPF ha una distanza migliore di iBGP. La distanza OSPF è 110, mentre la distanza iBGP è 200.

Disattivare la sincronizzazione su RTA in modo che RTA possa annunciare 192.168.250.15. Questa azione è necessaria perché RTA non viene sincronizzato con OSPF a causa della differenza nelle maschere. Mantenere disattivata la sincronizzazione su RTB in modo che RTB possa pubblicizzare 192.168.203.13. Questa azione è necessaria in RTB per lo stesso motivo.

A questo punto, attivare l'interfaccia RTB s1 per vedere come sono le route. Inoltre, abilitare OSPF su seriale 1 di RTB per renderlo passivo. Questo passaggio consente a RTA di conoscere il next-hop 192.168.10.5 tramite IGP. Se non si esegue questa operazione, i loop di routing si verificano perché, per raggiungere il next-hop 192.168.10.5, è necessario procedere in direzione opposta tramite eBGP. Queste sono le nuove configurazioni di RTA e 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
```

Le tabelle BGP sono simili alle seguenti:

<#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
```

È possibile progettare la rete in diversi modi per comunicare con i due diversi ISP, AS200 e AS300. Un modo consiste nel disporre di un ISP primario e di un ISP di backup. È possibile acquisire route parziali da uno degli ISP e route predefinite verso entrambi gli ISP. In questo esempio, si ricevono route parziali da AS200 e solo route locali da AS300. Sia RTA che RTB generano route predefinite in OSPF, con RTB come preferenza a causa della metrica inferiore. In questo modo, è possibile bilanciare il traffico in uscita tra i due ISP.

Una potenziale asimmetria può verificarsi se il traffico che lascia RTA ritorna tramite RTB. Questa situazione può verificarsi se si utilizza lo stesso pool di indirizzi IP, la stessa rete principale, quando si parla con i due ISP. A causa dell'aggregazione, l'intero AS può sembrare un'entità intera all'esterno. I punti di ingresso alla rete possono avvenire tramite RTA o RTB. Si può scoprire che tutto il traffico in arrivo al proprio AS arriva attraverso un unico punto, anche con più punti verso Internet. Nell'esempio, vi sono due reti principali diverse per i due ISP.

Un altro potenziale motivo di asimmetria è la diversa lunghezza del percorso pubblicizzato per raggiungere l'AS. Forse un provider di servizi è più vicino a una determinata destinazione rispetto a un altro. Nell'esempio, il traffico da AS400 che ha la rete come destinazione passa sempre tramite RTA in quanto il percorso è più breve. È possibile provare a mettere in atto tale decisione. È possibile utilizzare il comando set as-path prepend per anteporre i numeri di percorso agli aggiornamenti e aumentare la lunghezza del percorso. Tuttavia, con attributi quali preferenza locale, metrica o peso, AS400 può aver impostato il punto di uscita su AS200. In questo caso, non c'è niente che tu possa fare.

Questa configurazione è la configurazione finale per tutti i router:

```
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
```

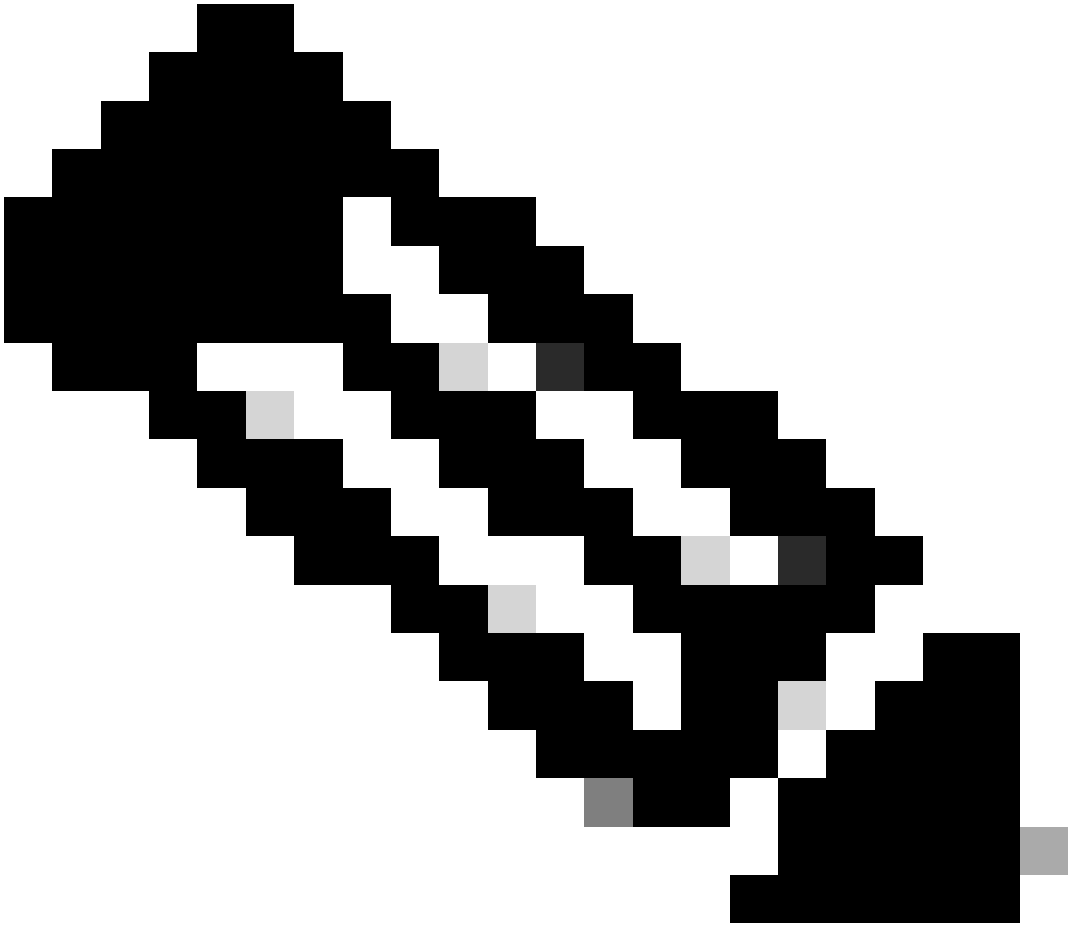
In RTA, la preferenza locale per le rotte che provengono da AS200 è impostata su 200. Inoltre, la rete 172.31.200.200 è la scelta per il candidato predefinito. Il comando ip default-network consente di scegliere l'impostazione predefinita.

Anche in questo esempio, l'uso del comando **default-information originate con OSPF inietta la route predefinita all'interno del dominio OSPF.** Anche in questo esempio viene utilizzato questo comando con il protocollo Intermediate System-to-Intermediate System (protocollo IS-IS) e il BGP. Per il RIP, esiste una ridistribuzione automatica nel RIP di 0.0.0.0, senza configurazione aggiuntiva. Per IGRP ed EIGRP, l'inserimento delle informazioni predefinite nel dominio IGP avviene dopo la ridistribuzione di BGP in IGRP ed EIGRP. Inoltre, con IGRP ed EIGRP, è possibile ridistribuire una route statica a 0.0.0.0 nel dominio IGP.

```
RTF#
hostname RTF

ip subnet-zero
```

```
interface Ethernet0
 ip address 172.31.14.250 255.255.255.0

interface Serial1
 ip address 172.16.15.250 255.255.255.252

router ospf 10
 network 192.168.203.25 0.0.255.255 area 0

ip classless

RTB#
hostname RTB

ip subnet-zero

interface Loopback1
 ip address 172.16.15.2500 255.255.255.252

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

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

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

Per RTB, la preferenza locale per gli aggiornamenti provenienti da AS300 è impostata su 300. Questo valore è superiore al valore della preferenza locale degli aggiornamenti iBGP provenienti da RTA. In questo modo, AS100 sceglie RTB per i percorsi locali di AS300. Qualsiasi altra route su RTB, se esistente, trasmette internamente con una preferenza locale di 100. Questo valore è inferiore alla preferenza locale di 200, che deriva da RTA. RTA è la preferenza.



Nota: sono state pubblicizzate solo le route locali AS300. Tutte le informazioni sul percorso che non corrispondono a ^300$ vengono eliminate. Se si desidera pubblicizzare le route locali e le route neighbor, che sono clienti dell'ISP, utilizzare ^300_[0-9]*.

Questo è l'output dell'espressione regolare che indica le route locali AS300:

<#root>

RTB#

show ip bgp regexp ^300$

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

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

RTC#
hostname RTC

ip subnet-zero

interface Loopback0
 ip address 192.168.128.6330 255.255.255.192

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

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

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

In RTC, si aggrega 172.31.10.0/16 e si indicano i percorsi specifici per l'iniezione in AS100. Se l'ISP rifiuta di eseguire questa operazione, è necessario filtrare in base all'estremità in ingresso di 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
```

Una dimostrazione di come usare il filtro della community è su RTG. Si aggiunge una no-export community agli aggiornamenti 192.168.211.0 verso RTD. In questo modo, RTD non esporta tale route in RTB. Tuttavia, in questo caso, RTB non accetta comunque queste route.

```
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 aggregates 172.31.200.200/16. Ecco le tabelle finali di BGP e routing per RTA, RTF e 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#
```

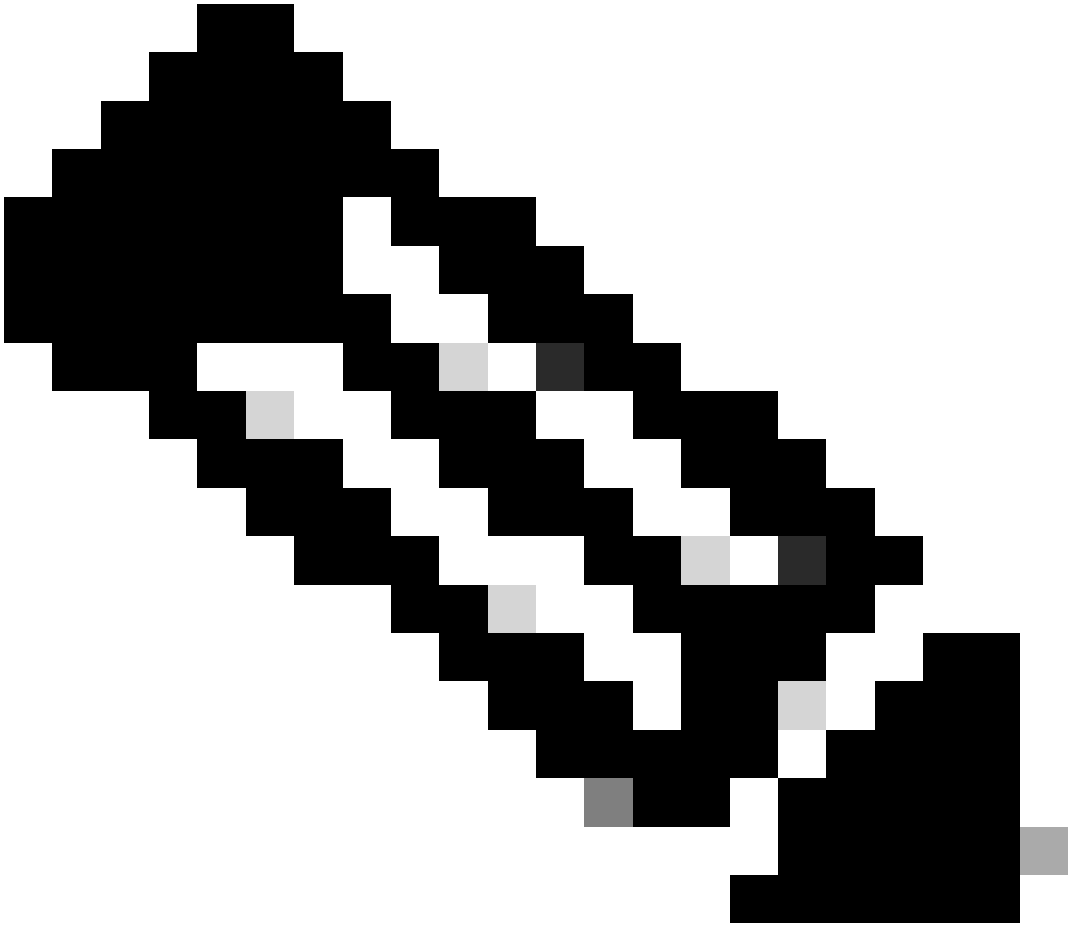
show ip route

```
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
```



Nota: la tabella di routing RTF indica che il modo per raggiungere le reti locali per AS300, ad esempio 192.168.10.0, è tramite RTB. Il modo per raggiungere altre reti note, ad esempio 172.31.200.200, è tramite RTA. Il gateway di ultima istanza è impostato su RTB. Se succede qualcosa alla connessione tra RTB e RTD, l'impostazione predefinita che pubblicizza RTA entra in gioco con una metrica di 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
```

Informazioni correlate

- **BGP: domande frequenti**
- **Esempi di configurazioni di BGP in un firewall PIX**
- **Come utilizzare HSRP per fornire ridondanza in una rete BGP multihome**
- **Configurazione della ridondanza in modalità singolo router e di BGP su un modulo MSFC Cat6000**
- **Routing ottimale e riduzione del consumo di memoria BGPR**
- **Risoluzione dei problemi comuni di BGP**
- **Risoluzione dei problemi di CPU elevata causati dal processo dello scanner o del router BGP**
- **Informazioni sulla condivisione del carico con il protocollo BGP in ambienti singlehomed e multihomed**
- **Pagina di supporto BGP**
- **Supporto tecnico e download Cisco**

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).