

# Border Gateway Protocol 고객 사례 조사

## 목차

---

### [소개](#)

#### [사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[표기규칙](#)

#### [BGP 사례 연구 1](#)

[BGP는 어떻게 작동합니까?](#)

[eBGP 및 iBGP](#)

[BGP 라우팅 활성화](#)

[양식 BGP 네이버](#)

[BGP 및 루프백 인터페이스](#)

[eBGP 멀티홉](#)

[eBGP 멀티홉\(로드 밸런싱\)](#)

[경로 맵](#)

[Match 및 Set 컨피그레이션 명령](#)

[예 1](#)

[예 2](#)

[네트워크 명령](#)

[재배포](#)

[고정 경로 및 재배포](#)

[iBGP](#)

[BGP 결정 알고리즘](#)

#### [BGP 사례 연구 2](#)

[AS\\_PATH 속성](#)

[Origin 특성](#)

[BGP Next Hop 특성](#)

[BGP Next Hop\(다중 액세스 네트워크\)](#)

[BGP Next Hop\(NBMA\)](#)

[Next-hop-self 명령](#)

[BGP 백도어](#)

[Synchronization\(동기화\)](#)

[동기화 사용 안 함](#)

[가중치 속성](#)

[로컬 기본 설정 속성](#)

[메트릭 특성](#)

[커뮤니티 특성](#)

#### [BGP 사례 연구 3](#)

[BGP 필터](#)

[경로 필터](#)

[경로 필터](#)

---

[AS 정규식](#)

[BGP 커뮤니티 필터](#)

[BGP 네이버 및 경로 맵](#)

[Set As-path Prepend 명령 사용](#)

[BGP 피어 그룹](#)

## [BGP 사례 연구 4](#)

[CIDR 및 집계 주소](#)

[집계 명령](#)

[CIDR 예 1](#)

[CIDR 예 2\(as-set\)](#)

[BGP 연합](#)

[경로 리플렉터](#)

[클러스터 내의 여러 RR](#)

[RR 및 일반 BGP 스피커](#)

[라우팅 정보의 루프 방지](#)

[경로 플랜 댐프닝](#)

[BGP가 경로를 선택하는 방법](#)

## [BGP 사례 연구 5](#)

[실용적인 디자인 예시](#)

[관련 정보](#)

---

## 소개

이 문서에는 5개의 BGP(Border Gateway Protocol) 고객 사례를 설명합니다.

## 사전 요구 사항

### 요구 사항

이 문서에 대한 특정 요건이 없습니다.

### 사용되는 구성 요소

이 문서는 특정 소프트웨어 및 하드웨어 버전으로 한정되지 않습니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

### 표기 규칙

문서 규칙에 대한 자세한 내용은 Cisco 기술 팁 표기 규칙을 참고하십시오.

## BGP 사례 연구 1

RFC 1771에서 정의한 [BGP](#)를 통해 AS(Autonomous System) 간에 루프 프리(loop-free) 도메인 간

라우팅을 생성할 수 있습니다. AS는 단일 기술 관리를 받는 라우터 집합입니다. AS의 라우터는 여러 IGP(Interior Gateway Protocol)를 사용하여 AS 내에서 라우팅 정보를 교환할 수 있습니다. 라우터는 외부 게이트웨이 프로토콜을 사용하여 AS 외부의 패킷을 라우팅할 수 있습니다.

## BGP는 어떻게 작동합니까?

BGP는 포트 179에서 TCP를 전송 프로토콜로 사용합니다. 두 BGP 라우터가 서로 TCP 연결을 형성합니다. 이 라우터는 피어 라우터입니다. 피어 라우터는 메시지를 교환하여 연결 매개변수를 열고 확인합니다.

BGP 라우터는 네트워크 연결 정보를 교환합니다. 이 정보는 주로 목적지 네트워크에 도달하기 위해 경로가 취해야 하는 전체 경로를 나타냅니다. 경로는 BGP AS 번호입니다. 이 정보는 루프 프리(loop-free) AS의 그래프 구성에 도움이 됩니다. 또한 이 그래프는 라우팅 동작에 대한 몇 가지 제한을 적용하기 위해 라우팅 정책을 적용할 위치를 보여줍니다.

BGP 라우팅 정보를 교환하기 위해 TCP 연결을 형성하는 두 라우터는 "피어" 또는 "네이버"입니다. BGP 피어는 처음에 전체 BGP 라우팅 테이블을 교환합니다. 이 교환 후 피어는 라우팅 테이블이 변경될 때 증분 업데이트를 전송합니다. BGP는 BGP 테이블의 버전 번호를 유지합니다. 버전 번호는 모든 BGP 피어에서 동일합니다. BGP가 라우팅 정보가 변경되어 테이블을 업데이트할 때마다 버전 번호가 변경됩니다. keepalive 패킷을 전송하면 BGP 피어 간의 연결이 유지됩니다. 알림 패킷은 오류 또는 특수 조건에 대한 응답으로 전송됩니다.

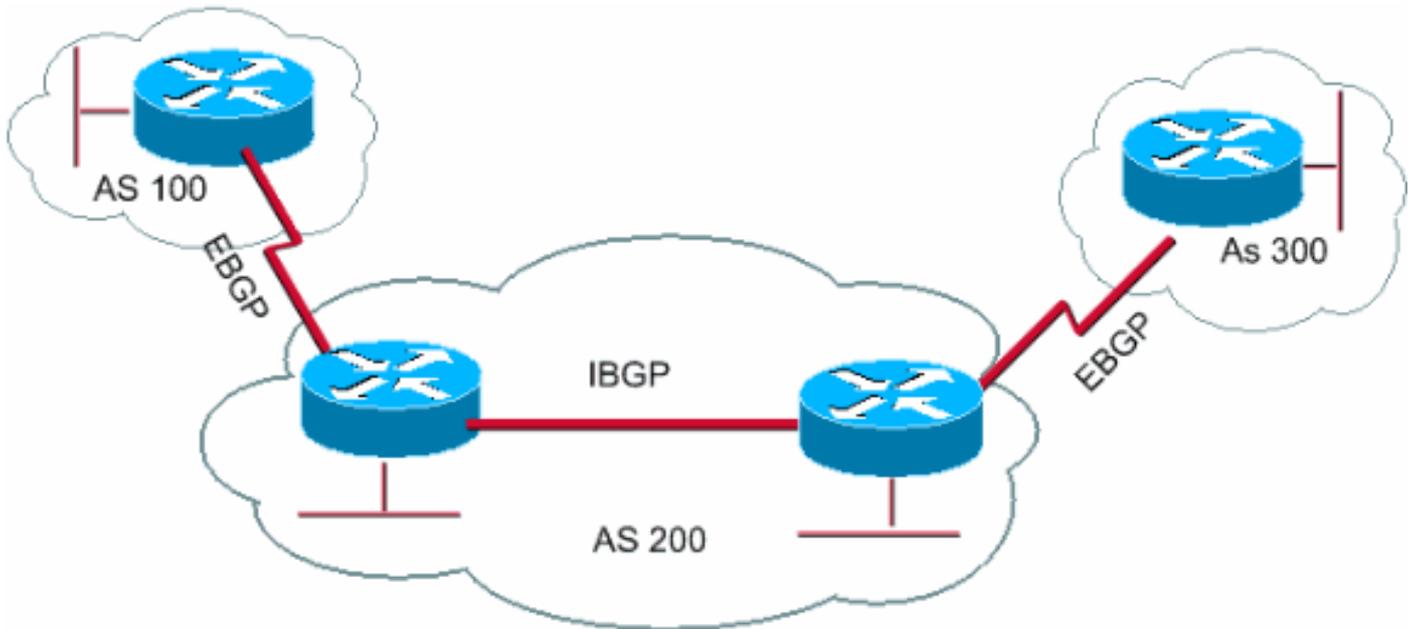
## eBGP 및 iBGP

AS에 여러 BGP 스피커가 있는 경우 AS는 다른 AS에 대한 전송 서비스 역할을 할 수 있습니다. 이 섹션의 다음 다이어그램에서 볼 수 있듯이 AS200은 AS100 및 AS300에 대한 전송 AS입니다.

외부 AS에 정보를 보내려면 네트워크에 대한 도달 가능성이 보장되어야 합니다. 네트워크 연결을 보장하기 위해 다음 프로세스가 수행됩니다.

- AS 내부의 라우터 간 iBGP(internal BGP) 피어링
- AS에서 실행되는 IGP에 대한 BGP 정보 재배포

서로 다른 두 AS에 속하는 라우터 간에 BGP가 실행되는 경우 이를 eBGP(exterior BGP)라고 합니다. 동일한 AS의 라우터 간에 BGP가 실행되는 경우 이를 iBGP라고 합니다.



BGP는 동일한 AS의 라우터 간에 실행됨

## BGP 라우팅 활성화

BGP를 활성화하고 구성하려면 다음 단계를 완료합니다.

RTA와 RTB라는 두 개의 라우터가 BGP를 통해 통신한다고 가정합니다. 첫 번째 예에서 RTA와 RTB는 서로 다른 AS입니다. 두 번째 예에서는 두 라우터가 모두 동일한 AS에 속합니다.

1. 라우터 프로세스 및 라우터가 속한 AS 번호를 정의합니다.

라우터에서 BGP를 활성화하려면 다음 명령을 실행합니다.

```
<#root>
```

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

```
RTA#
```

```
router bgp 100
```

```
RTB#
```

```
router bgp 200
```

이러한 명령문은 RTA가 BGP를 실행하고 AS100에 속함을 나타냅니다. RTB는 BGP를 실행하며 AS200에 속합니다.

2. BGP 인접 디바이스를 정의합니다.

BGP 인접 디바이스 형성은 BGP를 통해 통신을 시도하는 라우터를 나타냅니다. 다음 섹션에서는 이 프로세스에 대해 설명합니다.

## 양식 BGP 네이버

두 BGP 라우터가 서로 TCP 연결을 설정한 후 인접 디바이스가 됩니다. 두 피어 라우터가 라우팅 업데이트 교환을 시작하려면 TCP 연결이 필수적입니다.

TCP 연결이 설정되면 라우터가 값을 교환하기 위해 열린 메시지를 전송합니다. 라우터가 교환하는 값에는 AS 번호, 라우터가 실행하는 BGP 버전, BGP 라우터 ID 및 keepalive 보류 시간이 포함됩니다. 이러한 값의 확인 및 수락 이후에 네이버 연결의 설정이 발생합니다. Established 이외의 상태는 두 라우터가 네이버가 되지 않았으며 라우터가 BGP 업데이트를 교환할 수 없음을 나타냅니다.

TCP 연결 neighbor 을 설정하려면 다음 명령을 실행합니다.

```
<#root>
```

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

명령의 숫자는 BGP와 연결할 라우터의 AS 번호입니다. ip 주소는 eBGP에 대한 직접 연결을 사용하는 다음 홑 주소입니다. iBGP의 경우 *ip-address*는 다른 라우터의 IP 주소입니다.

피어 라우터의 명령에서 사용하는 neighbor 두 IP 주소는 서로 연결할 수 있어야 합니다. 연결성을 확인하는 한 가지 방법은 두 IP 주소 간의 확장 ping입니다. 확장된 ping은 ping 라우터가 명령에서 지정하는 IP 주소를 스스로 사용하도록 neighbor 합니다. 라우터는 패킷이 전송되는 인터페이스의 IP 주소보다는 이 주소를 사용해야 합니다.

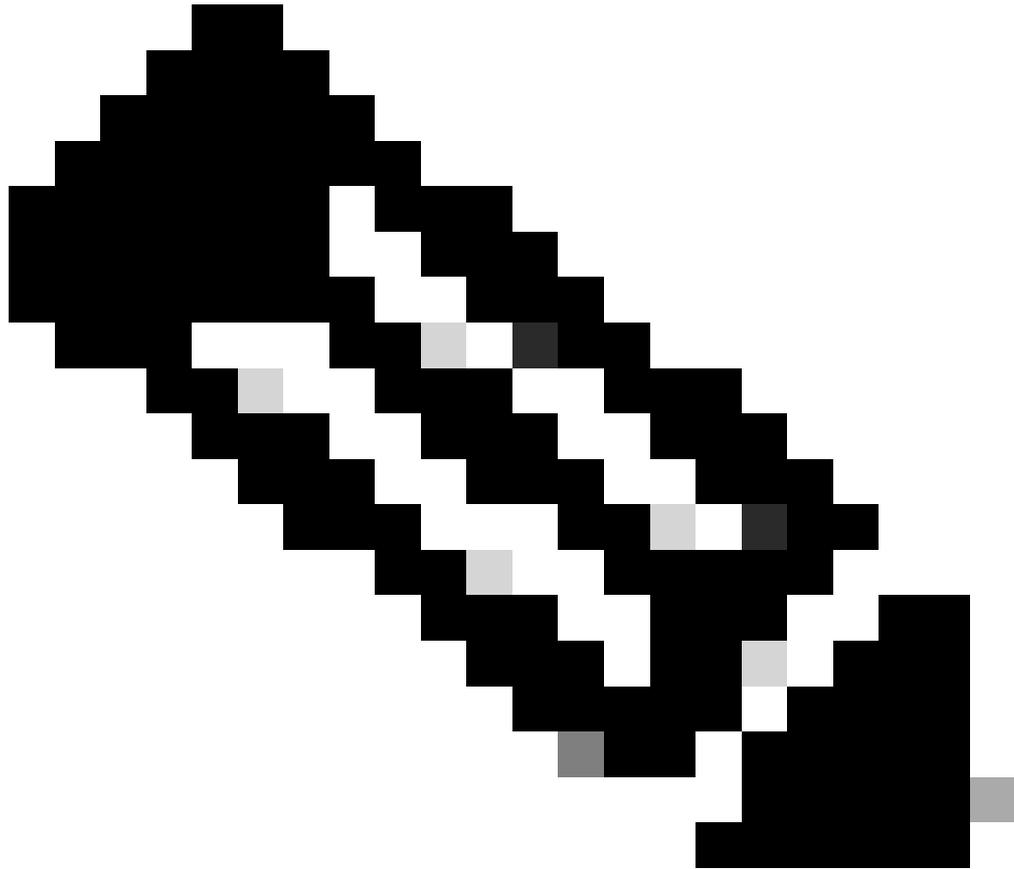
BGP 컨피그레이션 변경이 있는 경우 새 매개변수가 적용되도록 인접 디바이스 연결을 재설정해야 합니다. .

- 

```
clear ip bgp address
```

---

---



참고: 주소는 네이버 주소입니다

---

•

`clear ip bgp *`

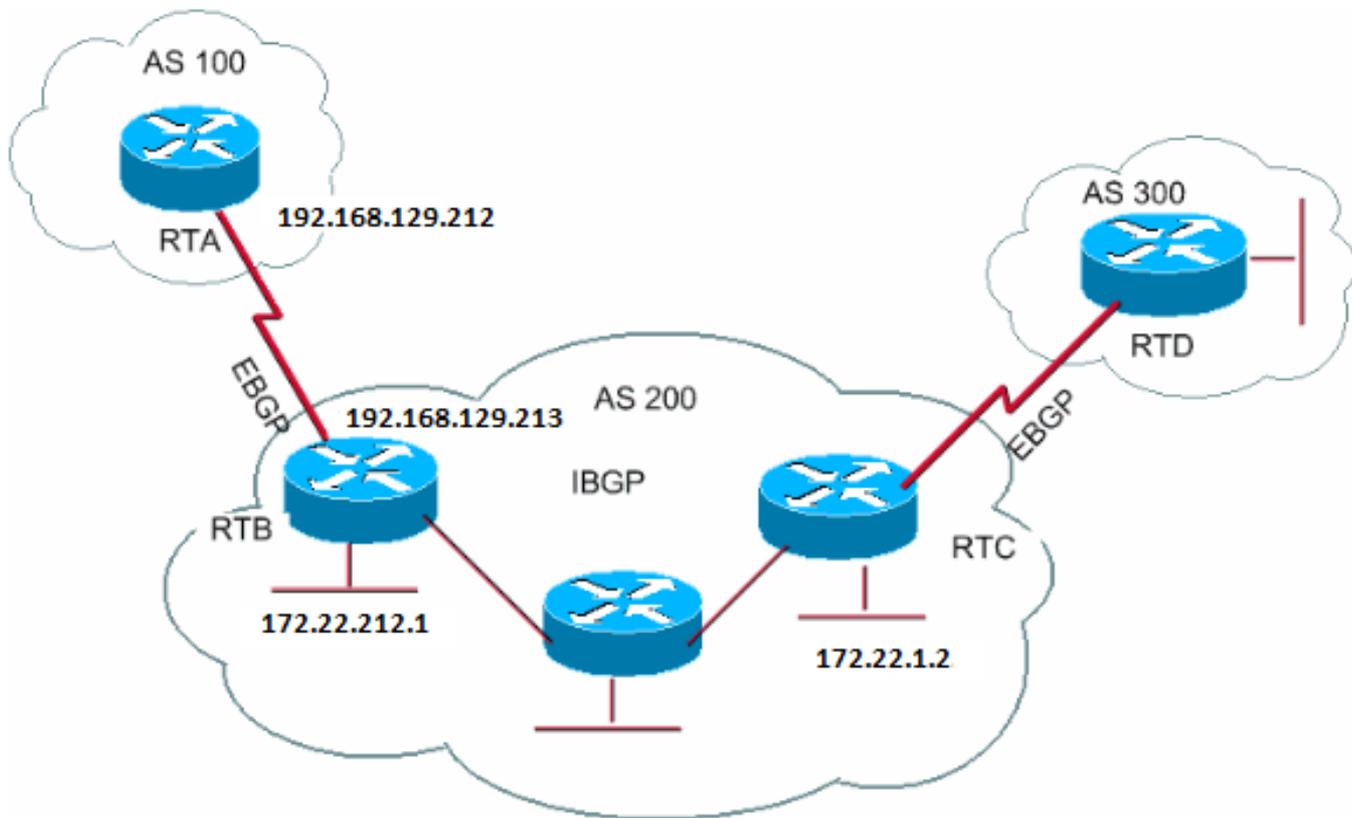
이 명령은 모든 네이버 연결을 지웁니다.

기본적으로 BGP 세션은 BGP 버전 4를 사용하는 것으로 시작되며 필요한 경우 이전 버전으로 하향 협상합니다. 협상을 방지하고 라우터가 네이버와 통신하는 데 사용하는 BGP 버전을 강제할 수 있습니다. 라우터 컨피그레이션 모드에서 다음 명령을 실행합니다.

```
<#root>
```

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

다음은 neighbor 명령 컨피그레이션의 예입니다.



RTA#

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

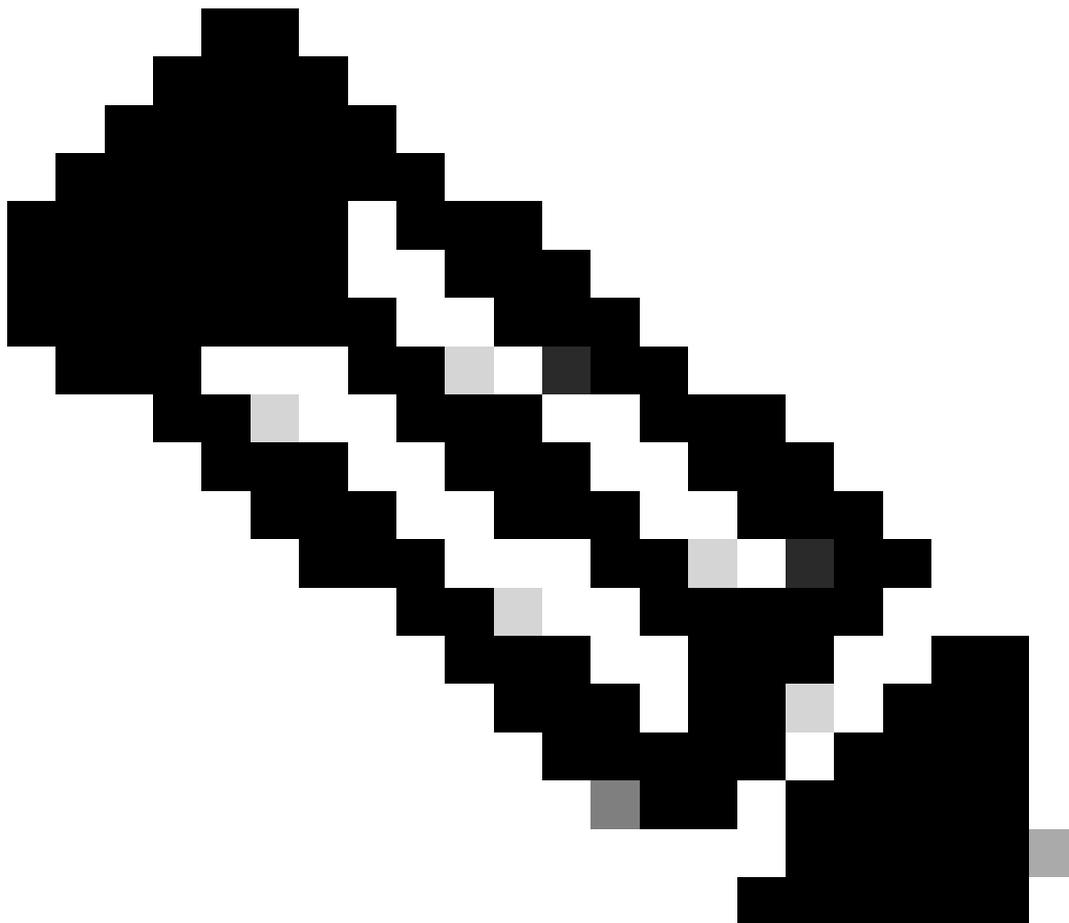
```
RTB#
router bgp 200
 neighbor 192.168.129.212 remote-as 100
 neighbor 172.22.1.2 remote-as 200
```

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

이 예에서 RTA 및 RTB는 eBGP를 실행합니다. RTB 및 RTC는 iBGP를 실행합니다. 원격 AS 번호는 외부 또는 내부 AS를 가리키며, 이는 eBGP 또는 iBGP를 나타냅니다. 또한 eBGP 피어에는 직접 연결이 있지만 iBGP 피어에는 직접 연결이 없습니다. iBGP 라우터에는 직접 연결이 필요하지 않습니다. 그러나 실행 하고 두 이웃이 서로 도달 할 수 있도록 허용 일부 IGP가 있어야 합니다.

이 섹션에서는 show ip bgp neighbors 명령이 표시하는 [정보의 예](#)를 제공합니다.

---



---

**참고:** BGP 상태에 각별히 주의하십시오. Established 이외의 상태는 피어가 작동 중이 아님을 나타냅니다. 다음 항목도 참고하십시오.

---

- 

BGP 버전, 4

- 

원격 라우터 ID

이 번호는 라우터에서 가장 높은 IP 주소 또는 가장 높은 루프백 인터페이스(있는 경우)입니다.

- 

테이블 버전

테이블 버전은 테이블의 상태를 제공합니다. 새 정보가 들어올 때마다 테이블이 버전을 늘립니다. 계속 증가하는 버전은 경로의 지속적인 업데이트를 유발하는 일부 경로 플랩이 있음을 나타냅니다.

<#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 및 루프백 인터페이스

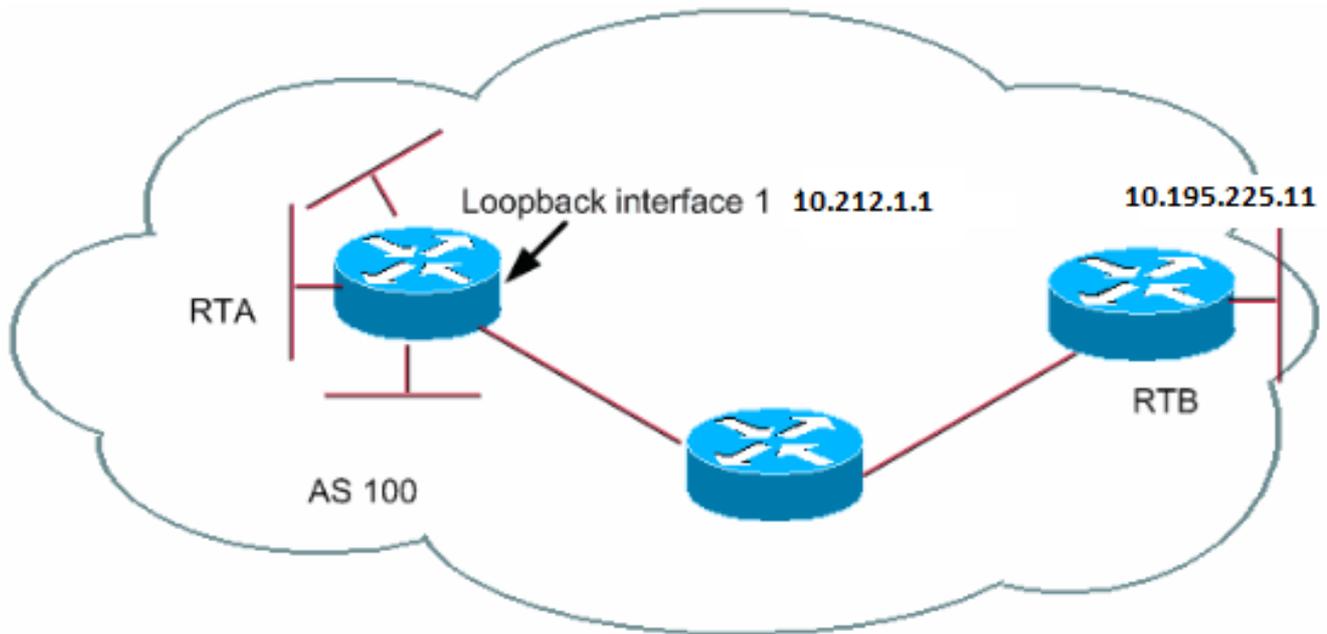
루프백 인터페이스를 사용하여 네이버를 정의하는 것은 iBGP에서는 공통적이지만 eBGP에서는 공통적이지 않습니다. 일반적으로 루프백 인터페이스를 사용하여 네이버의 IP 주소가 작동 상태를 유지하고 정상적으로 작동하는 하드웨어와 독립적인지 확인합니다. eBGP의 경우 피어 라우터가 직접 연결되는 경우가 많으며 루프백은 적용되지 않습니다.

이 명령에서 루프백 인터페이스의 IP 주소를 사용할 경우 neighbor 네이버 라우터에 추가 컨피그레이션이 필요합니다. 인접 라우터는 BGP 인접 디바이스 TCP 연결을 시작하기 위해 물리적 인터페이스가 아닌 루프백 인터페이스를 사용했음을 BGP에 알려야 합니다. 루프백 인터페이스를 나타내려면 다음 명령을 실행합니다.

```
<#root>
```

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

다음 예에서는 이 명령의 사용을 설명합니다.



```

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

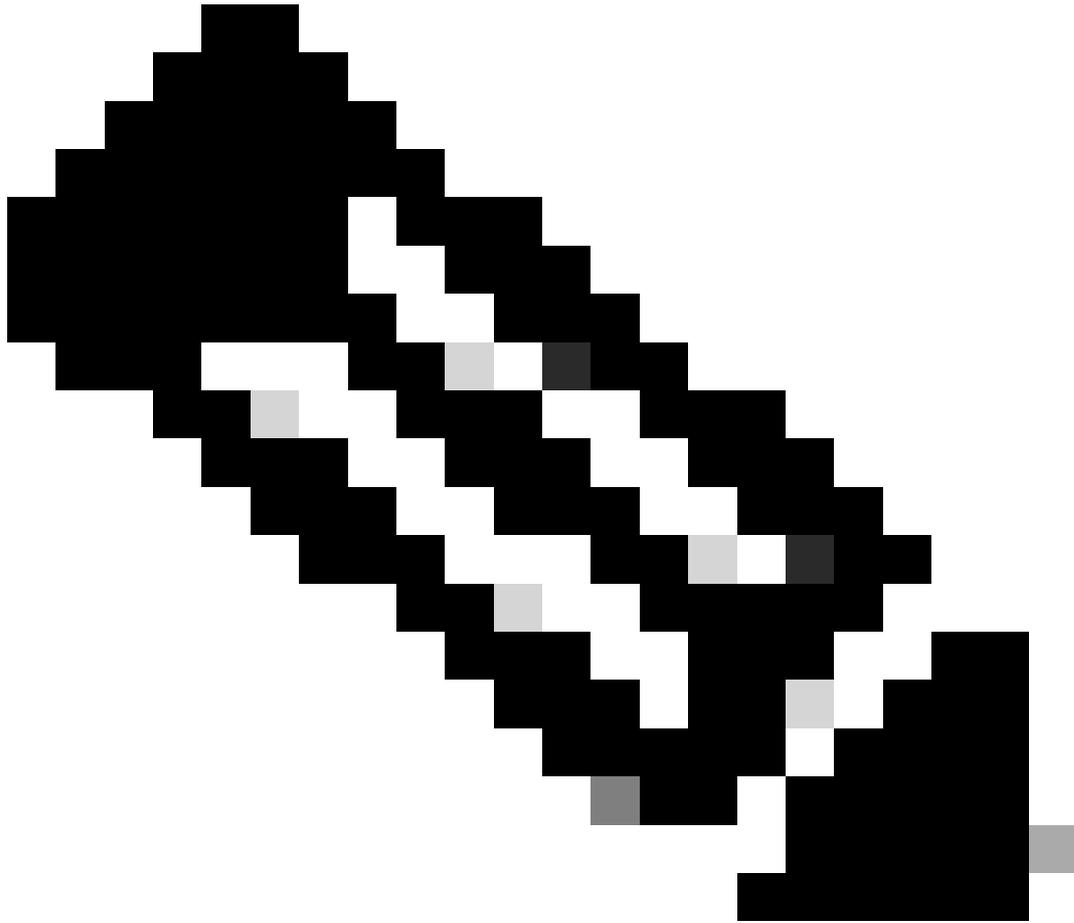
```

```

RTB#
router bgp 100
 neighbor 10.212.1.1 remote-as 100

```

이 예에서 RTA 및 RTB는 AS100 내에서 iBGP를 실행합니다. 명령에서 neighbor RTB는 RTA의 루프백 인터페이스인 10.212.1.1을 사용합니다. 이 경우 RTA는 TCP 네이버 연결의 소스로 루프백 IP 주소를 사용하도록 BGP를 강제해야 합니다. 이 작업을 강제로 수행하려면 RTA가 명령을 추가하도록 **update-source interface-type interface-number** 추가합니다neighbor 10.195.225.11 update-source loopback 1. 이 명령문은 BGP가 인접 디바이스 10.195.225.11과 통신할 때 BGP가 루프백 인터페이스의 IP 주소를 사용하도록 합니다

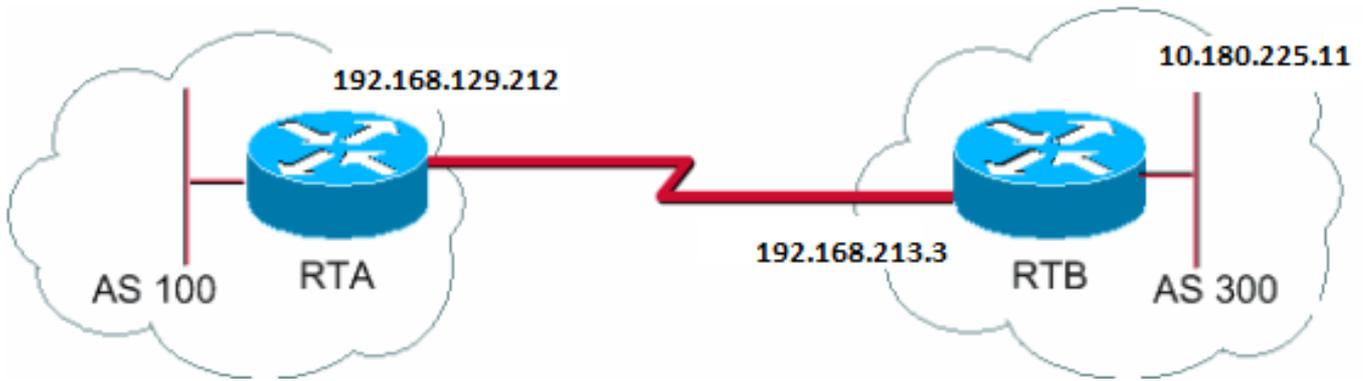


**참고:** RTA는 RTB의 물리적 인터페이스 IP 주소 10.195.225.11을 네이버로 사용했습니다. 이 IP 주소를 사용하면 RTB에 특별한 컨피그레이션이 필요하지 않습니다. 전체 네트워크 시나리오 샘플 컨피그레이션은 루프백 주소가 있거나 없는 iBGP 및 eBGP에 대한 샘플 컨피그레이션을 참조하십시오.

---

## eBGP 멀티홉

경우에 따라 Cisco 라우터는 두 외부 피어의 직접 연결을 허용하지 않는 서드파티 라우터로 eBGP를 실행할 수 있습니다. 연결을 위해 eBGP 멀티홉을 사용할 수 있습니다. eBGP 멀티홉은 직접 연결이 없는 두 외부 피어 간의 네이버 연결을 허용합니다. multihop은 eBGP에만 해당하고 iBGP에는 해당되지 않습니다. 다음 예는 eBGP 멀티홉을 보여줍니다.



```

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

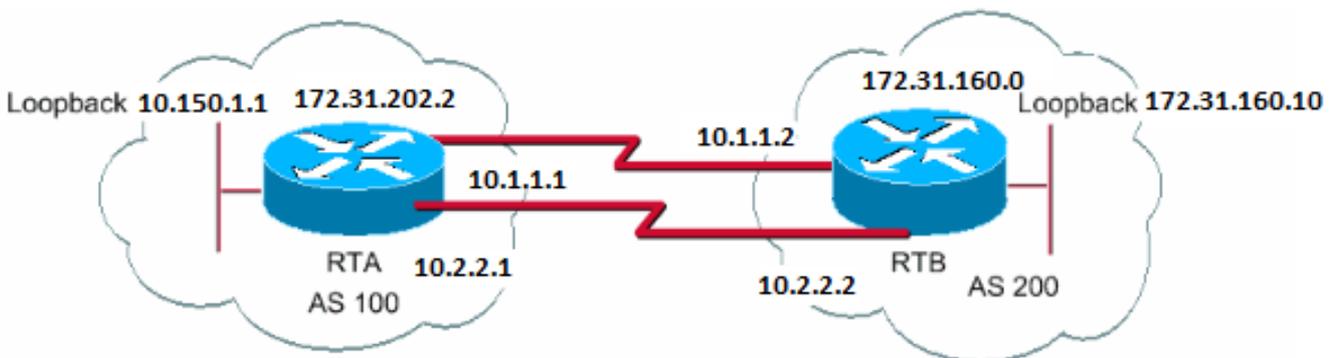
RTB#
router bgp 300
 neighbor 192.168.129.212 remote-as 100

```

RTA는 직접 연결되지 않은 외부 인접 디바이스를 나타냅니다. RTA는 neighbor ebgp-multihop 명령의 [사용을 나타내야](#) 합니다. 반면, RTB는 직접 연결된 인접 디바이스(192.168.129.212)를 나타냅니다. 이러한 직접 연결 때문에 RTB는 명령이 필요하지 neighbor ebgp-multihop 않습니다. 또한 IGP 또는 고정 라우팅을 구성하여 연결되지 않은 인접 디바이스가 서로 도달할 수 있도록 해야 합니다.

BGP multihop (Load Balancing) 섹션의 예에서는 병렬 라인을 통한 eBGP가 있는 경우 BGP와의 로드 밸런싱을 수행하는 방법을 보여 줍니다.

eBGP 멀티홉(로드 밸런싱)



```

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
```

이 예에서는 루프백 인터페이스, update-source 및 의 사용을 ebgp-multihop 설명합니다. 이 예는 병렬 직렬 회선을 통해 두 eBGP 스피커 간의 로드 밸런싱을 달성하는 방법입니다. 일반적인 상황에서는 BGP가 패킷을 전송할 라인 중 하나를 선택하므로 로드 밸런싱이 발생하지 않습니다. 루프백 인터페이스가 도입되면 eBGP에 대한 다음 홉은 루프백 인터페이스입니다. 고정 경로 또는 IGP를 사용하여 대상에 도달하기 위해 두 개의 동일 비용 경로를 도입합니다. RTA에는 다음 홉 172.31.160.10에 도달하는 두 가지 옵션이 있습니다. 즉, 10.1.1.2를 통한 경로 하나와 10.2.2.2를 통한 경로 하나가 있습니다. RTB도 마찬가지입니다.

## 경로 맵

BGP에서 경로 맵을 많이 사용합니다. BGP 컨텍스트에서 경로 맵은 라우팅 정보를 제어하고 수정하는 방법입니다. 라우팅 정보의 제어 및 수정은 한 라우팅 프로토콜에서 다른 라우팅 프로토콜로 경로를 재배포하기 위한 조건 정의를 통해 이루어집니다. 또는 BGP를 드나드는 과정에서 라우팅 정보를 제어할 수 있습니다. 경로 맵의 형식은 다음과 같습니다.

<#root>

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

맵 태그는 단순히 경로 맵에 부여하는 이름입니다. 동일한 경로 맵 또는 동일한 이름 태그의 여러 인스턴스를 정의할 수 있습니다. 시퀀스 번호는 동일한 이름으로 이미 구성한 경로 맵 목록에 새 경로 맵이 있을 위치를 나타내는 것입니다.

이 예에서는 경로 맵의 두 인스턴스가 정의되었으며 이름은 MYMAP입니다. 첫 번째 인스턴스의 시퀀스 번호는 10이고, 두 번째 인스턴스의 시퀀스 번호는 20입니다.

- 

**route-map MYMAP permit 10**(첫 번째 조건 집합은 여기로 이동)

- 

**route-map MYMAP permit 20**(두 번째 조건 집합은 여기에 있습니다.)

수신 또는 발신 경로에 경로 맵 MYMAP을 적용할 경우 첫 번째 조건 집합이 인스턴스 10을 통해 적용됩니다. 첫 번째 조건 집합이 충족되지 않으면 경로 맵의 상위 인스턴스로 진행합니다.

Match 및 Set 컨피그레이션 명령

각 경로 맵은 match 및 set 컨피그레이션 명령 목록으로 구성됩니다. match는 기준을 match 지정하고 set은 명령에서 set 적용하는 기준이 충족되는 경우 match 작업을 지정합니다.

예를 들어 발신 업데이트를 확인하는 경로 맵을 정의할 수 있습니다. IP 주소 10.1.1.1에 대한 일치がある 경우 해당 업데이트에 대한 메트릭은 5로 설정됩니다. 다음 명령은 예를 보여줍니다.

```
<#root>
```

```
match ip address 10.1.1.1
```

```
set metric 5
```

이제 일치 기준이 충족되고 가 있는 경우 set 작업permit에서 지정한 대로 경로가 재배포되거나 제어됩니다. 명단에서 빠지셨네요

일치 기준이 충족되고 가 있는 경우deny 경로의 재배포나 제어가 없습니다. 명단에서 빠지셨네요

일치 기준이 충족되지 않은 상태에서 또는 permit deny 이 있으면 경로 맵의 다음 인스턴스가 선택됩니다. 예를 들어, 인스턴스 20이 선택됩니다. 이 next-instance 검사는 경로 맵의 모든 인스턴스를 중단하거나 완료할 때까지 계속됩니다. 일치 항목 없이 목록을 완료 하면 경로가 다음과 not accepted nor forwarded 같습니다.

Cisco IOS Software Release 11.2 이전의 Cisco IOS® Software 릴리스에서 프로토콜 간에 재배포하지 않고 경로 맵을 사용하여 BGP 업데이트를 필터링하는 경우 IP 주소에서 **match** 명령을 사용할 때 인바운드에서 필터링할 수 없습니다. 아웃바운드의 필터를 사용할 수 있습니다. Cisco IOS Software 릴리스 11.2 이상에는 이러한 제한이 없습니다.

에 대한 관련 명령은 다음과 match 같습니다.

- 

match as-path

- 

match community

- 

match clns

- 

match interface

- 

match ip address

- 

match ip next-hop

- 

match ip route-source

- 

match metric

- 

match route-type

- 

match tag

에 대한 관련 명령은 다음과 set 같습니다.

- 

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

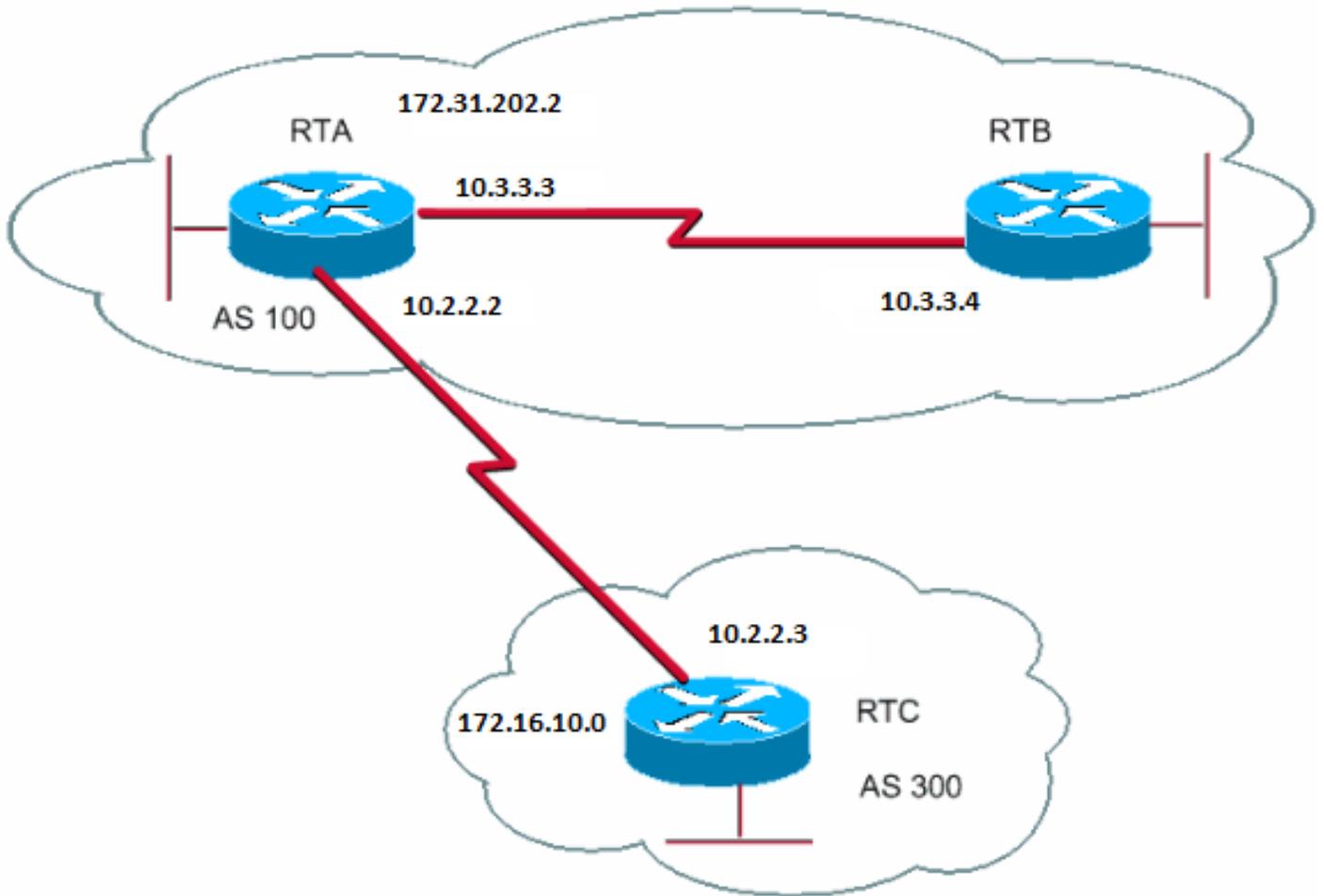
- 

set tag

- 

set weight

몇 가지 경로 맵 예를 살펴보겠습니다.



경로 맵 예

예 1

RTA 및 RTB가 RIP(Routing Information Protocol)를 실행하고 RTA 및 RTC가 BGP를 실행한다고 가정합니다. RTA는 BGP를 통해 업데이트를 가져오고 RIP에 업데이트를 재배포합니다. RTA가 메트릭이 2인 약 172.16.10.0의 RTB 경로와 메트릭이 5인 다른 모든 경로에 재배포하고자 한다고 가정하자. 이 경우 다음 컨피그레이션을 사용할 수 있습니다.

```

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

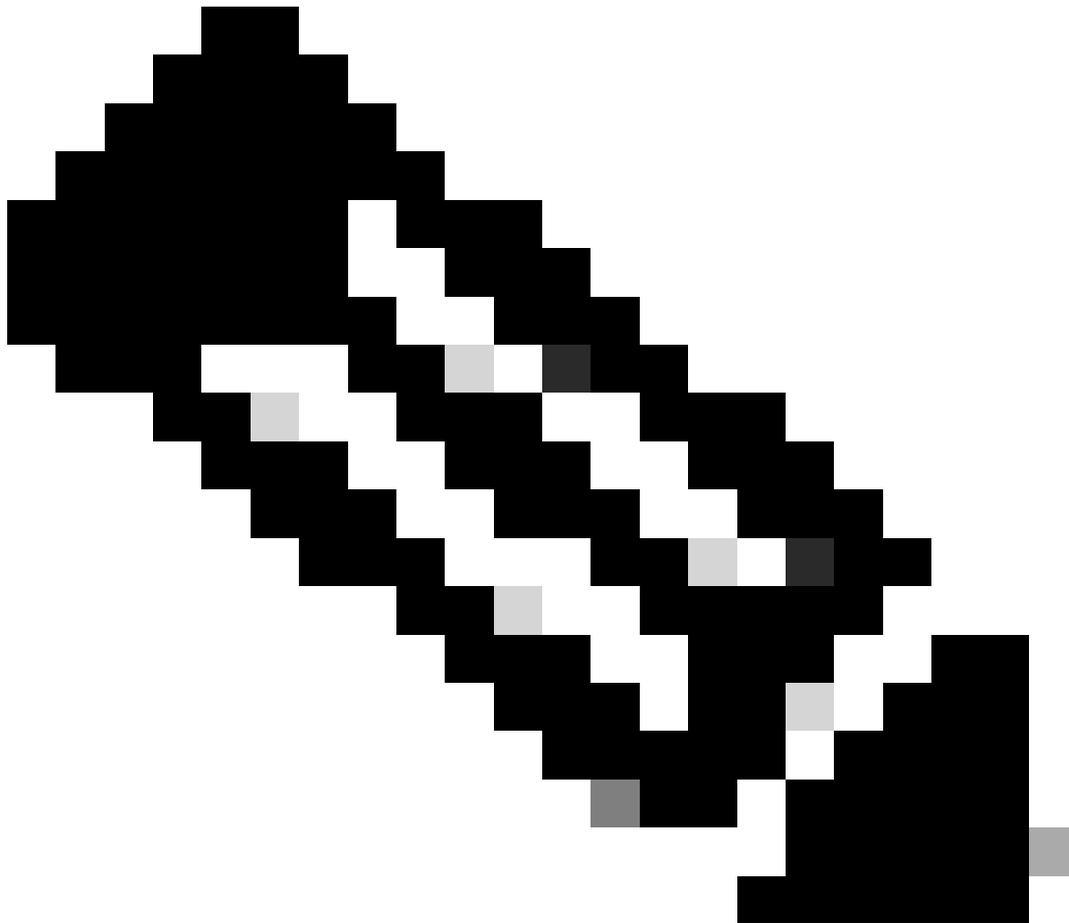
route-map SETMETRIC permit 20
set metric 5

```

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

이 예에서 경로가 IP 주소 172.16.10.0과 일치하면 경로의 메트릭은 2입니다. 그런 다음 경로 맵 목록에서 나갑니다. 일치하는 항목이 없으면 경로 맵 목록 아래로 진행합니다. 그러면 다른 모든 항목이 메트릭 5로 설정됩니다.

---



**참고:** 항상 "어떤 일치 명령문과도 일치하지 않는 경로는 어떻게 됩니까?"라는 질문을 하십시오. 이러한 경로는 기본적으로 삭제됩니다.

---

예 1에서 AS100이 약 172.16.10.0에 대한 업데이트를 수락하지 않도록 한다고 가정합니다. IP 주소를 기준으로 일치하는 경우 인바운드에 경로 맵을 적용할 수 없습니다. 따라서 RTC에서 아웃바운드 경로 맵을 사용해야 합니다.

```
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
```

이제 BGP를 시작하는 방법과 인접 디바이스를 정의하는 방법에 대해 더 익숙해졌습니다. 네트워크 정보 교환을 시작하는 방법을 알아보십시오.

BGP를 사용하여 네트워크 정보를 전송하는 방법은 여러 가지가 있습니다. 다음 섹션에서는 각 방법을 하나씩 살펴봅니다.

- 

네트워크 명령

- 

재배포

- 

고정 경로 및 재배포

네트워크 명령

명령의 형식은 network 다음과 같습니다.

<#root>

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

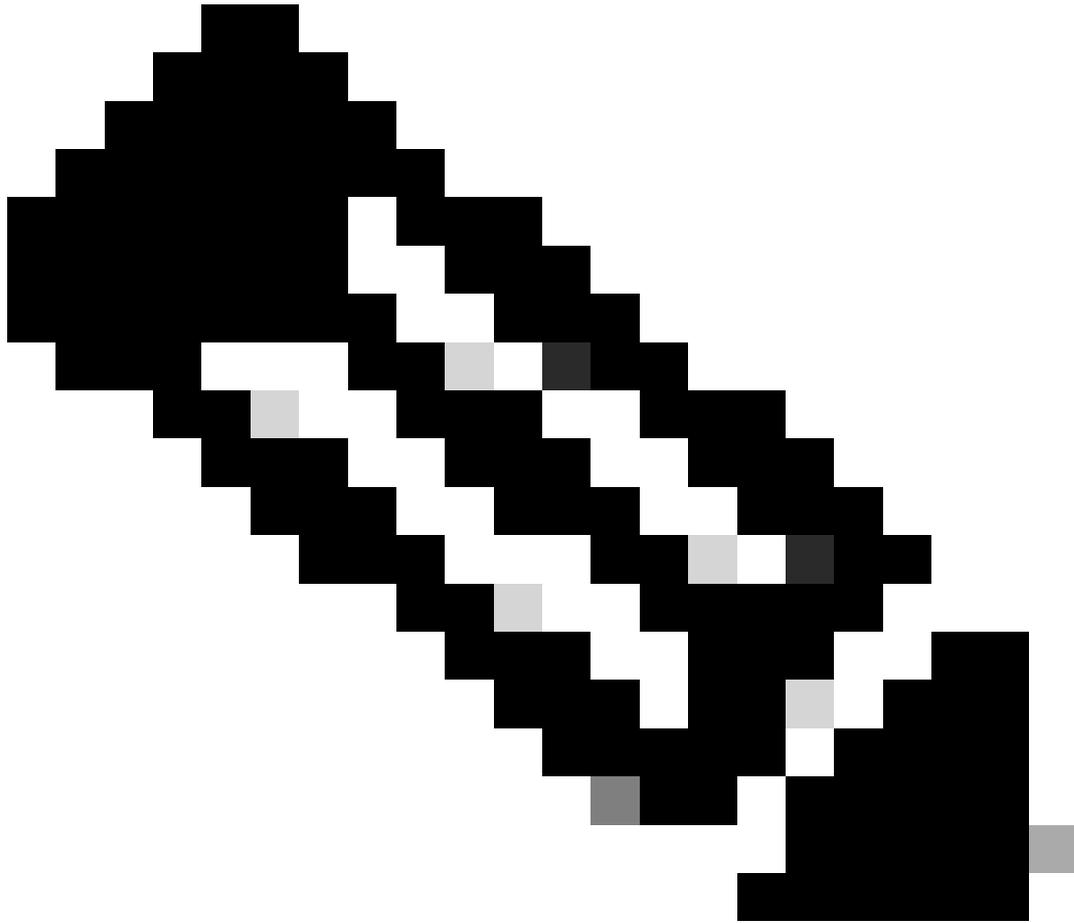
이 network 명령은 이 상자에서 시작되는 네트워크를 제어합니다. 이 개념은 IGRP(Interior Gateway Routing Protocol) 및 RIP를 사용하는 친숙한 컨피그레이션과는 다릅니다. 이 명령을 사용하면 특정 인터페이스에서 BGP를 실행하려고 하지 않습니다. 대신 BGP가 이 상자에서 시작해야 하는 네트워크를 BGP에 표시하려고 합니다. BGP 버전 4(BGP4)에서 서브넷팅 및 수퍼네팅을 처리할 수 있으므로 이 명령은 마스크 부분을 사용합니다. 최대 200개의 network 명령 항목을 입력할 수 있습니다.

이 network 명령은 라우터가 네트워크에 연결되었는지, 고정인지 또는 동적으로 학습했는지를 알고 있는 경우에 작동합니다.

network 명령의 예는 다음과 같습니다.

```
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
```

이 예는 라우터 A가 192.168.213.0/16에 대한 네트워크 엔트리를 생성함을 나타냅니다. /16은 클래스 C 주소의 수퍼넷을 사용하고 처음 두 개의 옥텟 또는 처음 16비트를 광고함을 나타냅니다.



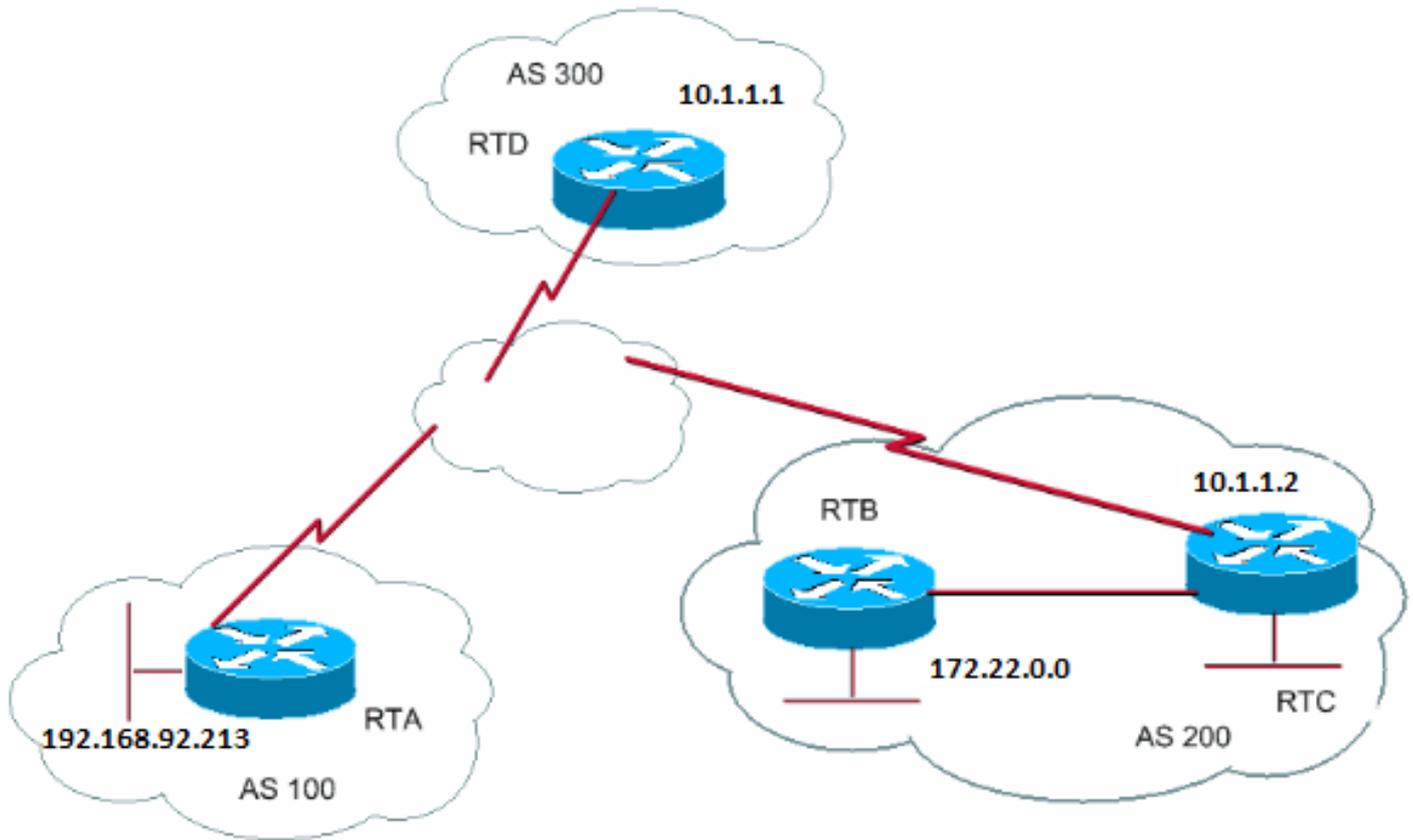
**참고:** 고정 경로는 라우팅 테이블에 일치하는 엔트리를 배치하므로 라우터가 192.168.213.0을 생성하도록 하려면 고정 경로가 필요합니다.

---

## 재배포

이 `network` 명령은 BGP를 통해 네트워크를 광고하는 한 가지 방법입니다. 또 다른 방법은 IGP를 BGP로 재배포하는 것입니다. IGP는 IGRP, OSPF(Open Shortest Path First) 프로토콜, RIP, EIGRP(Enhanced Interior Gateway Routing Protocol) 또는 다른 프로토콜일 수 있습니다. 이러한 재배포는 무서운 것처럼 보일 수 있습니다. 이제 모든 내부 경로를 BGP로 덤프합니다. 이러한 경로 중 일부는 BGP를 통해 학습될 수 있으며 다시 보낼 필요가 없습니다. 필터링하여 광고할 인터넷 전용 경로에만 전송할지, 보유한 모든 경로에는 전송할지를 확인해야 합니다. 이제 DDoS 공격의 실제 사례를 살펴보겠습니다.

RTA는 192.168.92.213을, RTC는 172.22.0.0을 발표합니다. RTC 컨피그레이션을 확인합니다.



명령을 실행하면 다음network이 수행됩니다.

```
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.*

재배포를 대신 사용하는 경우 다음과 같은 이점이 있습니다.

```
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.
```

이러한 재배포로 인해 AS에 의해 192.168.92.213이 생성됩니다. 사용자는 192.168.92.213의 소스가 아니며 AS100은 소스입니다. 따라서 AS가 해당 네트워크에서 소스를 유출하지 못하도록 필터를 사용해야 합니다. 올바른 컨피그레이션은 다음과 같습니다.

```
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
```

AS200access-list에서 시작된 네트워크를 제어하려면 이 명령을 사용합니다.

OSPF를 BGP로 재배포하는 것은 다른 IGP에 대한 재배포와 약간 다릅니다. 아래의 단순한 redistribute ospf 1문제router bgp는 통하지 않는다. 각 경로를 재배포하기 위해 internal, external 및 **nssa-external** 와 같은 특정 키워드가 필요합니다. 자세한 [내용은 BGP에 대한 OSPF 경로 재배포](#) 이해를 참조하십시오.

고정 경로 및 재배포

네트워크 또는 서브넷을 시작하려면 항상 정적 경로를 사용 할 수 있습니다. 유일한 차이점은 BGP가 이러한 경로를 불완전하거나 알 수 없는 출처로 간주한다는 것입니다. Redistribution 섹션의 예에서 수행한 것과 동일한 결과를 수행할 수 있습니다.

```
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
```

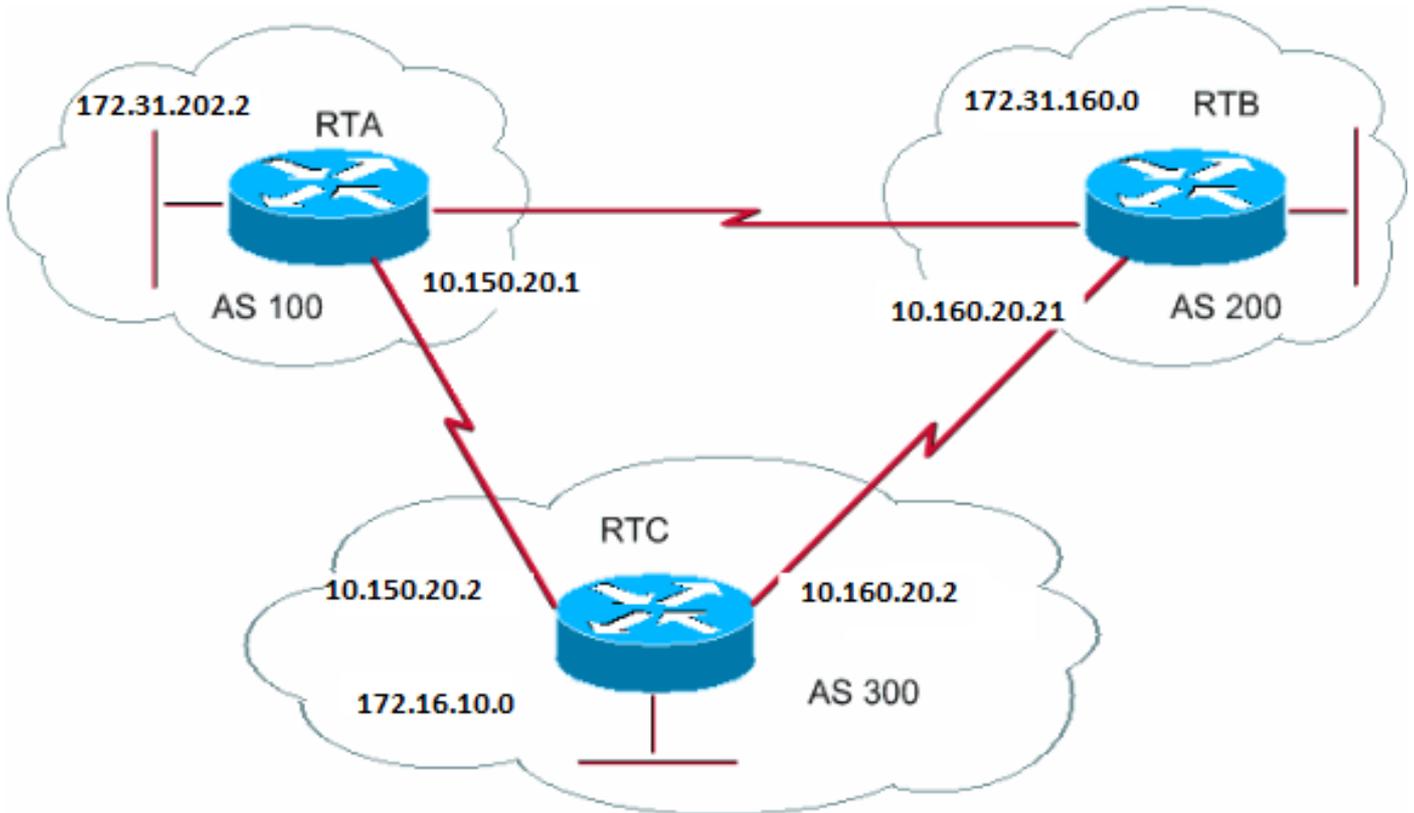
인터페이스null0 는 패킷을 무시합니다. 따라서 패킷을 가져오고 172.22.0.0보다 더 구체적인 일치기 존재할 경우 라우터는 패킷을

특정 일치로 전송합니다. 그렇지 않으면 라우터가 패킷을 무시합니다. 이 방법은 수퍼넷을 광고하는 좋은 방법입니다.

이 문서에서는 다양한 방법을 사용하여 AS에서 경로를 시작하는 방법에 대해 설명했습니다. 이러한 경로는 BGP가 인접 디바이스를 통해 학습한 다른 BGP 경로(내부 또는 외부)와 함께 생성된다는 점에 유의하십시오. BGP는 BGP가 학습한 정보를 한 피어에서 다른 피어로 전달합니다. 차이점은 network 명령, 재배포 또는 정적으로 생성된 경로는 AS가 이러한 네트워크의 출처임을 나타냅니다.

재배포는 항상 IGP에 BGP를 삽입하는 방법입니다.

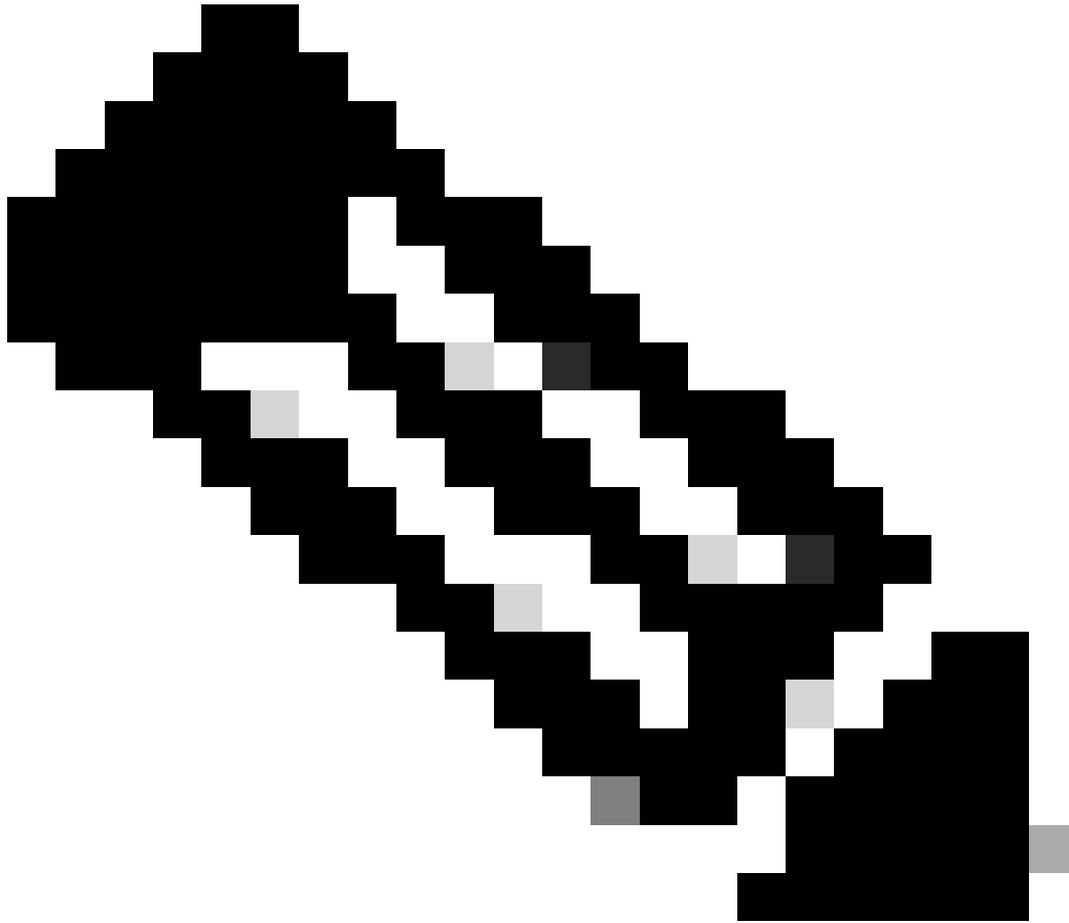
예를 들면 다음과 같습니다.



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

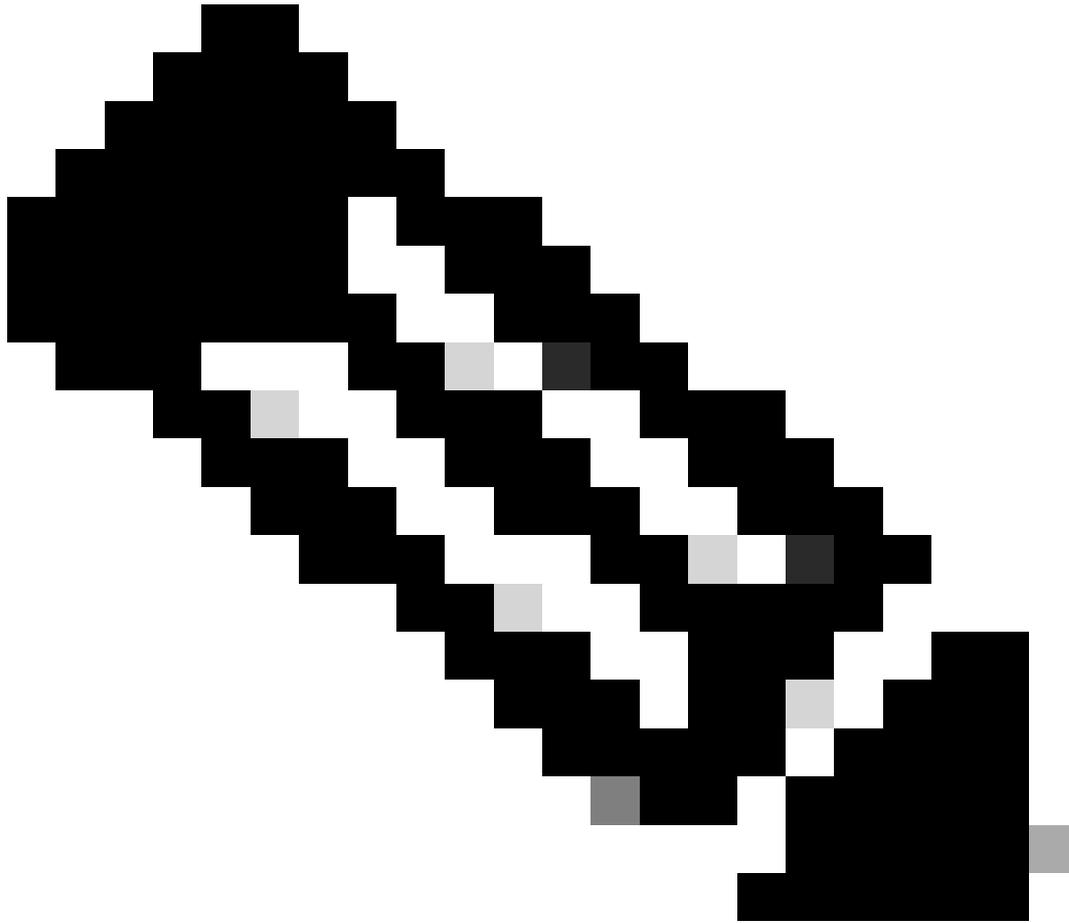
```
RTB#  
router bgp 200  
neighbor 10.160.20.2 remote-as 300  
network 172.31.160.0
```

```
RTC#  
router bgp 300  
neighbor 10.150.20.1 remote-as 100  
neighbor 10.160.20.21 remote-as 200  
network 170.10.0.0
```



**참고:** RTC에서 이러한 네트워크를 생성하고 AS100 및 AS200에서 들어오는 이러한 네트워크를 전달하려는 경우가 아니라면 RTC에 네트워크 172.31.202.2 또는 네트워크 172.31.160.0이 필요하지 않습니다. 또 다른 차이점은 `network` 명령이 이러한 동일한 네트워크에 추가 광고를 추가한다는 것입니다. 이는 AS300이 이러한 경로의 출처이기도 함을 나타냅니다.

---



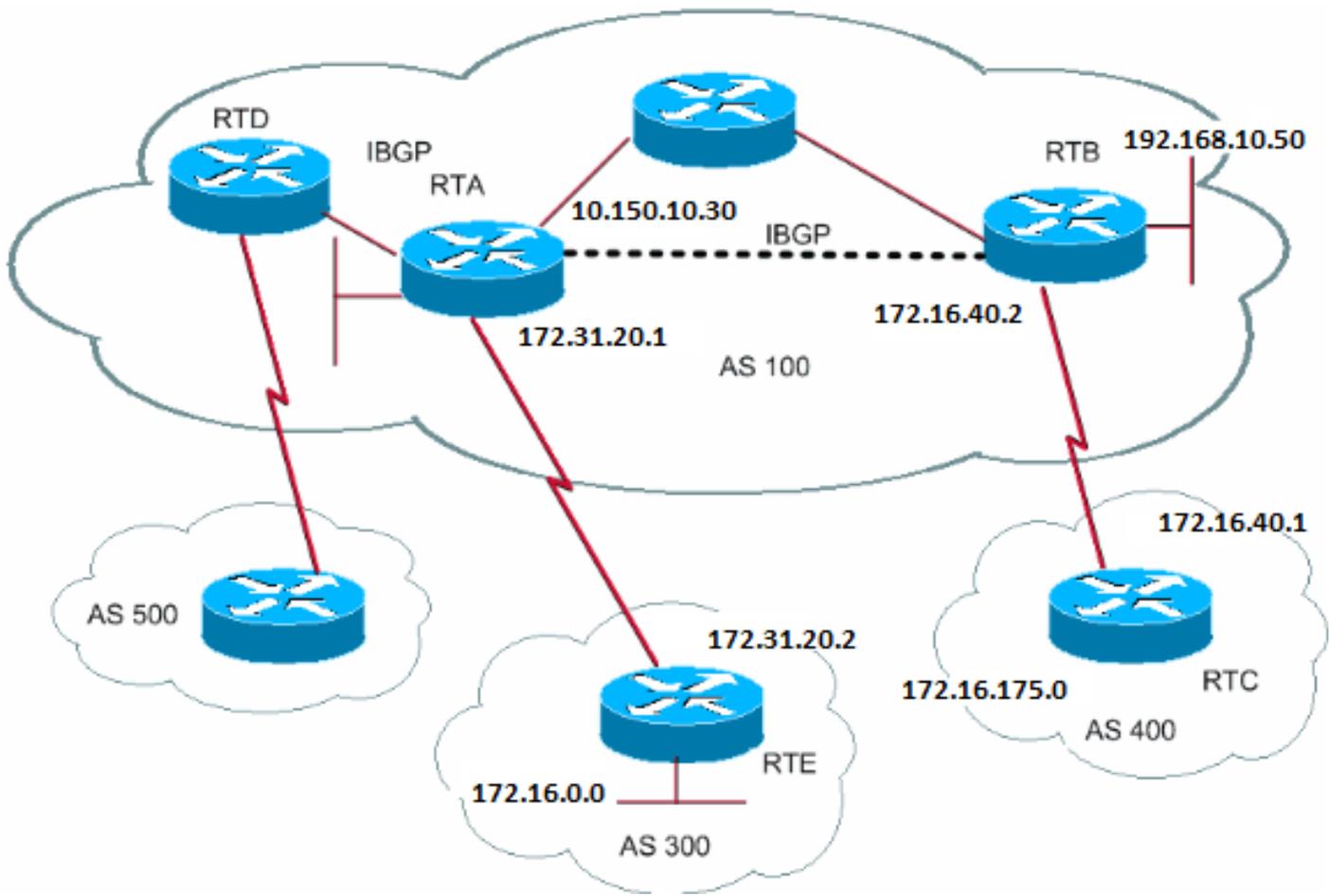
**참고:** BGP는 자체 AS에서 시작된 업데이트를 수락하지 않습니다. 이러한 거부는 루프 없는 도메인 간 토폴로지를 보장합니다.

예를 들어, 이 섹션의 예에서 AS200이 AS100에 대한 직접 BGP 연결을 갖는다고 가정합니다. RTA는 경로 172.31.202.2를 생성하고 AS300으로 경로를 전송합니다. 그런 다음 RTC는 이 경로를 AS200으로 전달하고 원점을 AS100으로 유지합니다. RTB는 172.31.202.2를 AS100으로 통과하며, 원점은 여전히 AS100입니다. RTA는 업데이트가 자체 AS에서 시작되었음을 알리고 업데이트를 무시합니다.

#### iBGP

AS가 다른 AS에 대한 전송 시스템 역할을 하려는 경우 iBGP를 사용합니다. eBGP를 통해 학습하고 IGP로 재배포한 다음 다시 다른 AS로 재배포하는 경우에도 동일한 작업을 수행할 수 있습니다. 그러나 iBGP는 AS 내에서 정보를 교환할 수 있는 더 유연하고 효율적

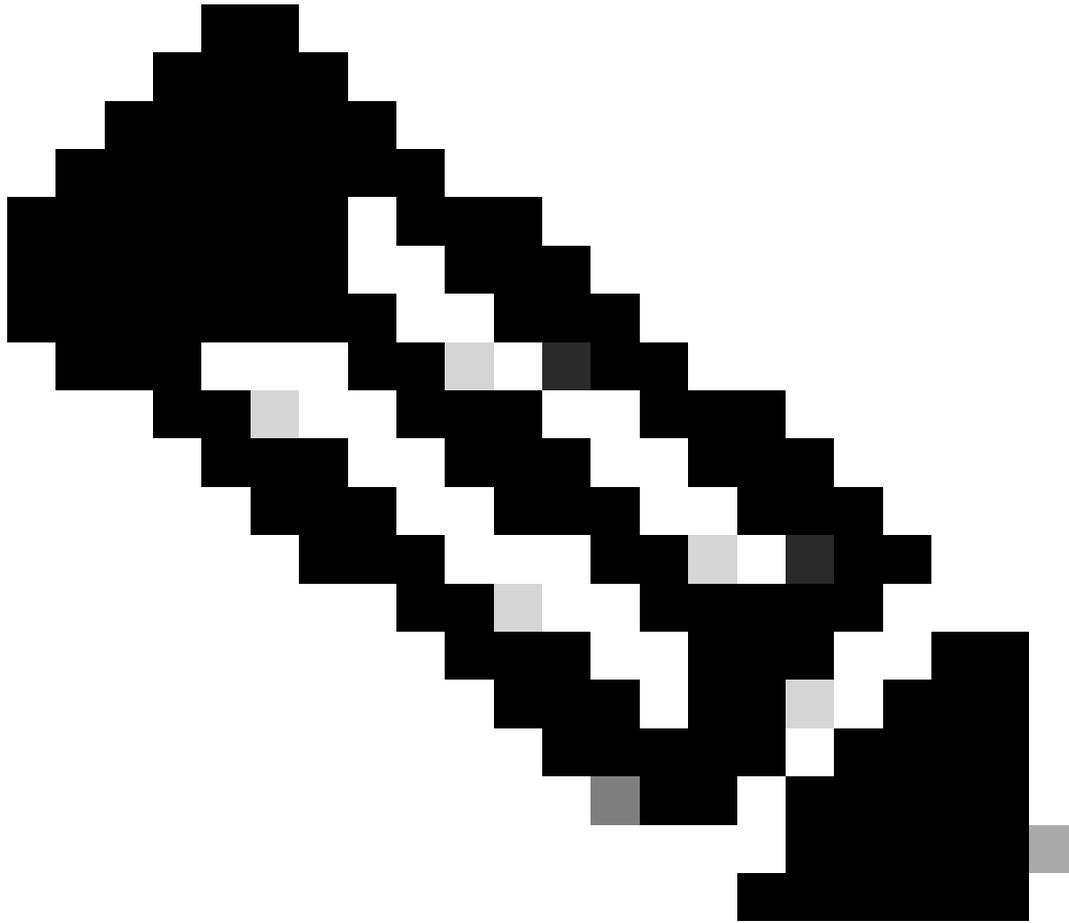
인 방법을 제공합니다. 예를 들어, iBGP는 로컬 환경 설정을 사용하여 AS에서 최상의 종료 지점을 제어하는 방법을 제공합니다. 로컬 기본 설정에 대한 자세한 내용은 로컬 기본 설정 속성 섹션을 참조하십시오.



```
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
```



**참고:** BGP 스피커가 자체 AS(iBGP)의 다른 BGP 스피커로부터 업데이트를 수신할 경우 업데이트를 수신한 BGP 스피커가 해당 정보를 자체 AS의 다른 BGP 스피커에 재배포하지 않습니다. 업데이트를 수신한 BGP 스피커가 AS 외부의 다른 BGP 스피커에 정보를 재배포합니다. 따라서 AS 내에서 iBGP 스피커 간에 풀 메시를 유지합니다.

RTA 및 RTB는 iBGP를 실행합니다. RTA 및 RTD는 iBGP도 실행합니다. RTB에서 RTA로 전송되는 BGP 업데이트는 AS 외부에 있는 RTE로 전송됩니다. 업데이트는 AS 내부에 있는 RTD로 전송되지 않습니다. 따라서 RTB와 RTD 간에 iBGP 피어링을 수행하여 업데이트의 흐름이 끊어지지 않도록 합니다.

#### BGP 결정 알고리즘

BGP가 서로 다른 자율 시스템에서 서로 다른 대상에 대한 업데이트를 수신한 후 프로토콜에서 특정 대상에 도달할 경로를 선택해야 합니다. BGP는 특정 대상에 도달하기 위해 단일 경로만 선택합니다.

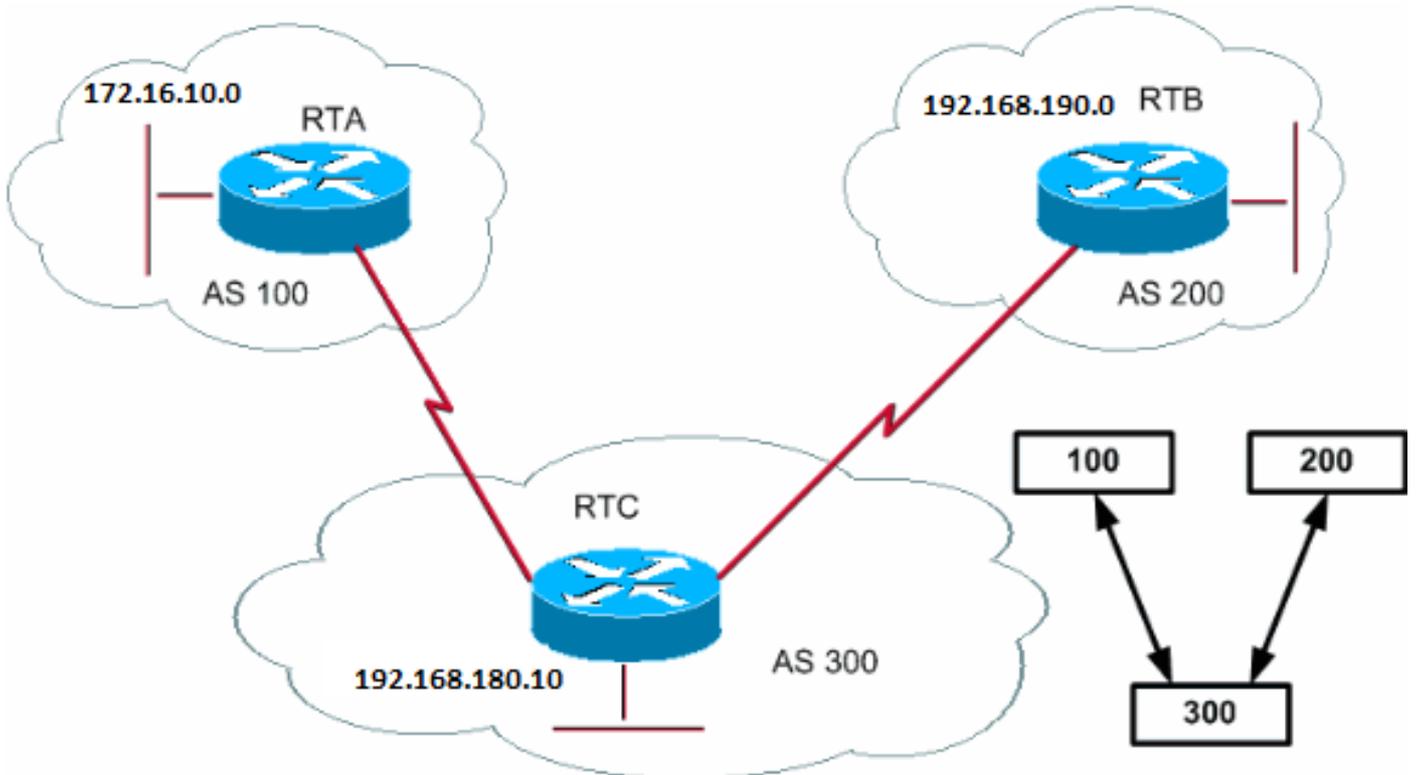
BGP는 다음 홉, 관리 가중치, 로컬 기본 설정, 경로 출처, 경로 길이, 출처 코드, 메트릭, 기타 속성 등 다른 attributes 특성을 기준으로 결정을 내립니다.

BGP는 항상 인접 디바이스에 최상의 경로를 전파합니다. 자세한 내용은 [내용은 BGP 최적 경로 선택 알고리즘을 참조하십시오.](#)

다음 섹션에서는 이러한 특성과 그 사용에 대해 설명합니다.

## BGP 사례 연구 2

### AS\_PATH 속성



경로 업데이트가 AS를 통과할 때마다 AS 번호가 해당 업데이트 앞에 추가됩니다. AS\_PATH 속성은 실제로 목적지에 도달하기 위해 경로가 통과한 AS 번호 목록입니다. AS\_SET는 트래버스된 모든 AS의 정렬된 수학적 세트입니다. 이 문서의 CIDR 예 2(as-set) 섹션에서는 AS\_SET의 예를 제공합니다.

이 섹션의 예에서 RTB는 AS200의 네트워크 192.168.190.0을 광고합니다. 해당 경로가 AS300을 통과하면 RTC는 자체 AS 번호를 네트워크에 추가합니다. 192.168.190.0이 RTA에 도달하면 네트워크에는 두 개의 AS 번호가 연결됩니다. 첫 번째 AS 번호는 200이고 그 다음 AS 번호는 300입니다. RTA의 경우 192.168.190.0에 도달하는 경로는 (300, 200)입니다.

172.16.10.0 및 192.168.180.10에도 동일한 프로세스가 적용됩니다. RTB는 경로(300, 100)를 취해야 합니다. RTB는 172.16.10.0에 도달하기 위해 AS300을 거쳐 AS100을 통과합니다. RTC는 192.168.190.0에 도달하려면 경로(200)를 통과해야 하고 172.16.10.0에 도달하려면 경로(100)를 통과해야 합니다.

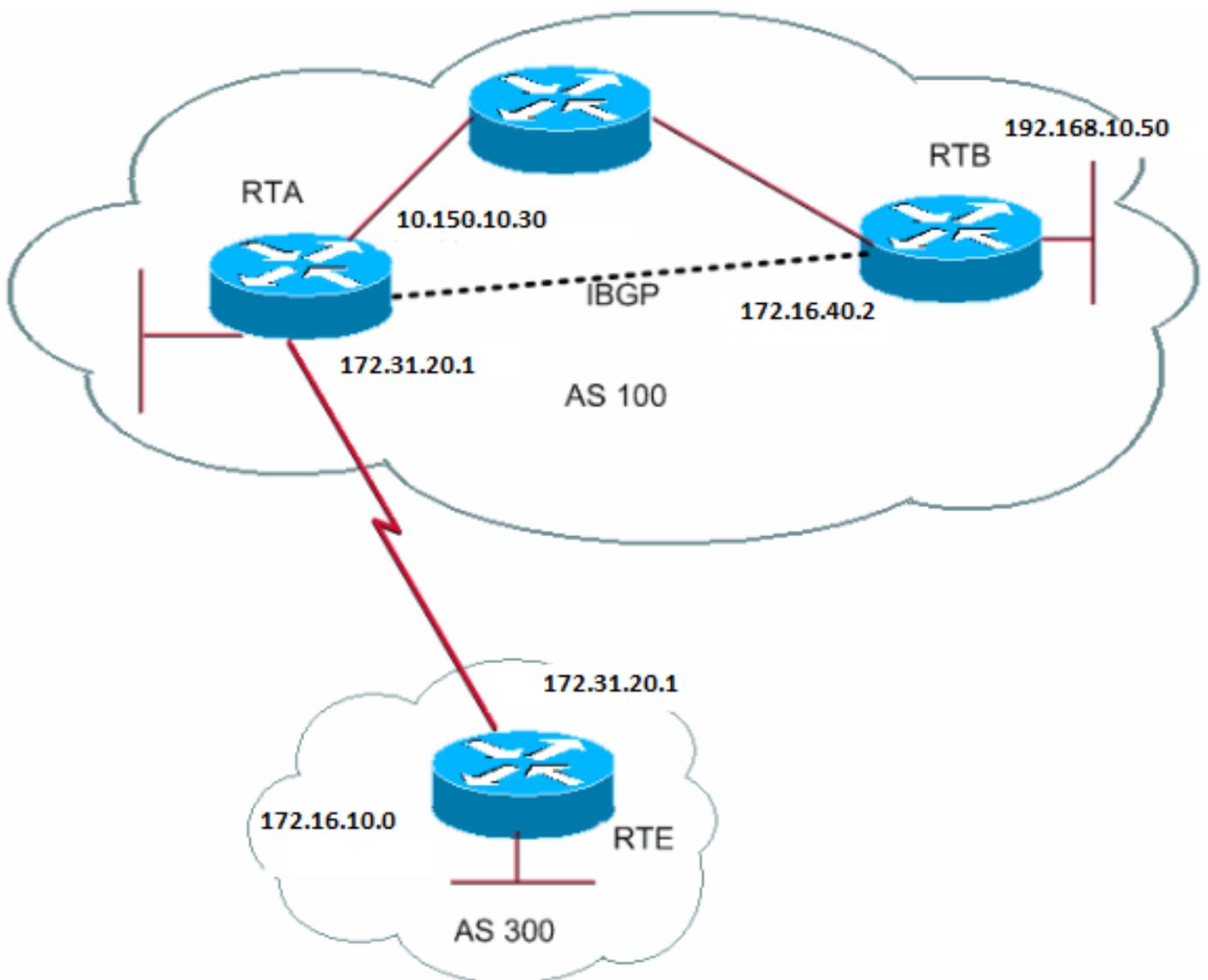
### Origin 특성

Origin은 경로 정보의 출처를 정의하는 필수 속성입니다. origin 특성은 다음 세 가지 값을 가질 수 있습니다.

IGP - NLRI(Network Layer Reachability Information)는 AS 시작 지점 내부에 있습니다. 이 작업은 일반적으로 명령을 실행할 때 **bgp network**발생합니다. BGP 테이블의 숫자는 IGP를 나타냅니다.

EGP - EGP(외부 게이트웨이 프로토콜)를 통해 NLRI를 학습합니다. BGP 테이블에서 EGP를 나타냅니다.

INCOMPLETE(불완전) - NLRI를 알 수 없거나 다른 방법을 통해 학습합니다. INCOMPLETE는 일반적으로 다른 라우팅 프로토콜의 경로를 BGP로 재배포하고 경로의 출처가 불완전할 때 발생합니다. BGP 테이블에서는 INCOMPLETE를 나타냅니다.



```
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#

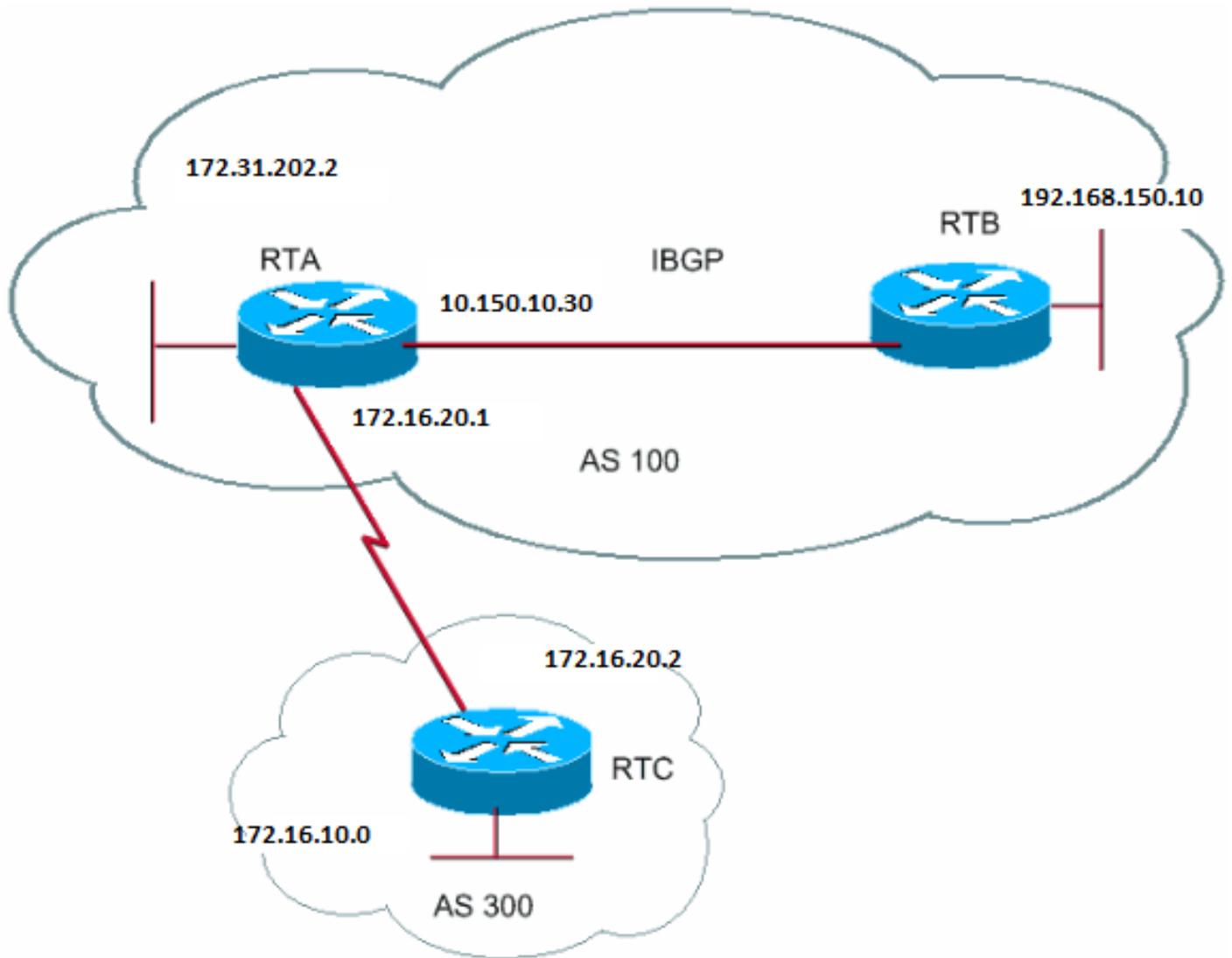
```
router bgp 100
neighbor 10.150.10.30 remote-as 100
network 192.168.10.150
```

RTE#

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

RTA는 300 i를 통해 172.16.10.0에 도달합니다. '300 i'는 다음 AS 경로가 300이고 경로의 기원이 IGP임을 의미한다. RTA도 i를 통해 192.168.10.150에 도달합니다. 이 "i"는 항목이 동일한 AS에 있고 원본이 IGP임을 의미합니다. RTE는 100i를 통해 172.31.202.2에 도달합니다. '100 i'는 다음 AS가 100이고 기원이 IGP라는 의미이다. RTE도 100 ?를 통해 192.168.190.0에 도달합니다. "100 ?"은 다음 AS가 100이고 원점이 불완전하며 고정 경로에서 나온다는 의미입니다.

BGP Next Hop 특성



### BGP Next Hop 특성

BGP next hop 특성은 특정 대상에 도달하기 위해 사용할 next hop IP 주소입니다.

eBGP의 경우 next hop은 항상 명령에서 지정하는 네이버의 IP neighbor 주소입니다. 이 섹션의 예에서 RTC는 다음 홉이 172.31.20.2인 RTA에 172.16.10.0을 광고합니다. RTA는 172.31.20.2를 다음 홉 172.31.20.1과 함께 RTC에 광고합니다. iBGP의 경우, 프로토콜은 eBGP가 광고하는 다음 홉을 iBGP로 전달해야 한다고 진술합니다. 이 규칙 때문에 RTA는 iBGP 피어 RTB에 다음 홉 172.31.20.2로 172.16.10.0을 알립니다. RTB를 기준으로 172.16.10.0에 도달하는 다음 홉은 172.31.20.2이며 10.150.10.30은 아닙니다.

IGP를 통해 RTB가 172.31.20.2에 도달할 수 있는지 확인합니다. 그렇지 않으면 다음 홉 주소에 액세스할 수 없으므로 RTB는 대상이 172.16.10.0인 패킷을 삭제합니다. 예를 들어 RTB에서 iGRP를 실행하는 경우 RTA 네트워크 172.16.10.0에서도 iGRP를 실행할 수 있습니다. BGP만 교환되도록 RTC에 대한 링크에서 iGRP를 패시브로 설정하려고 합니다.

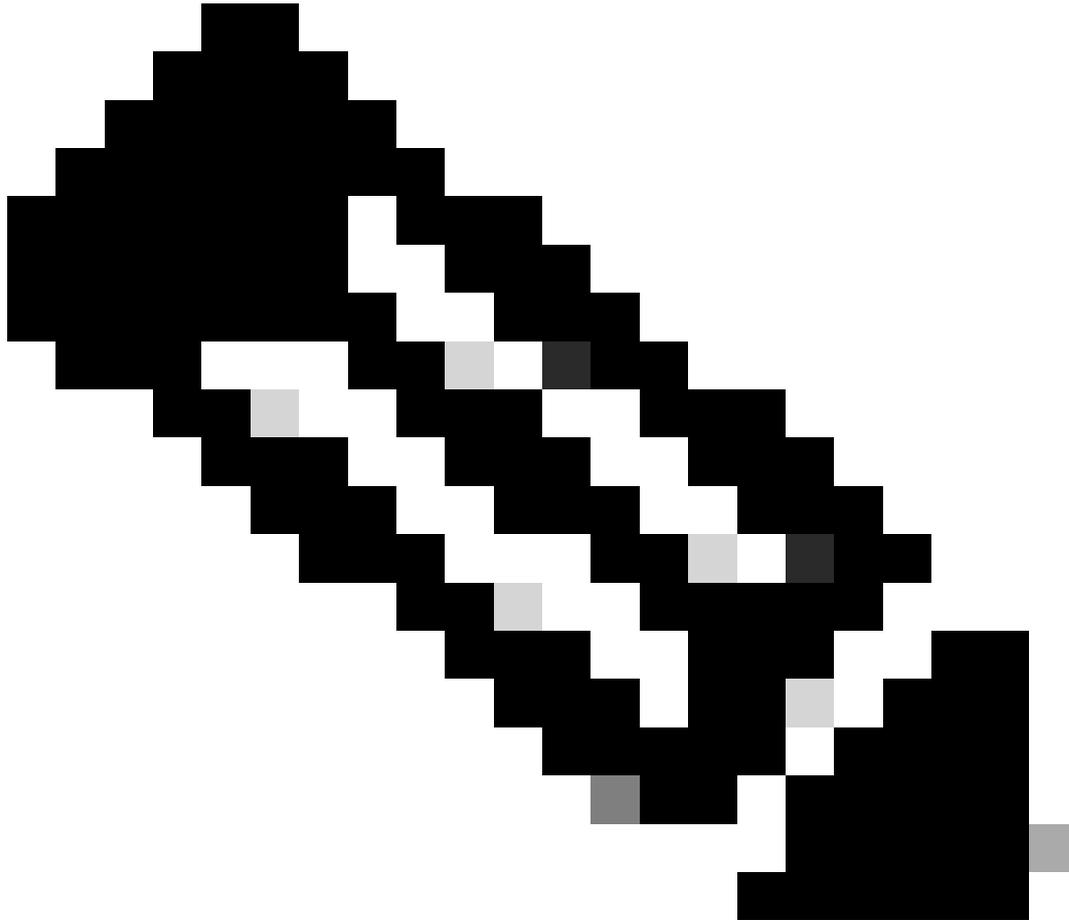
```
RTA#
router bgp 100
neighbor 172.31.20.2 remote-as 300
neighbor 192.168.150.10 remote-as 100
network 172.31.20.2
```

RTB#

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

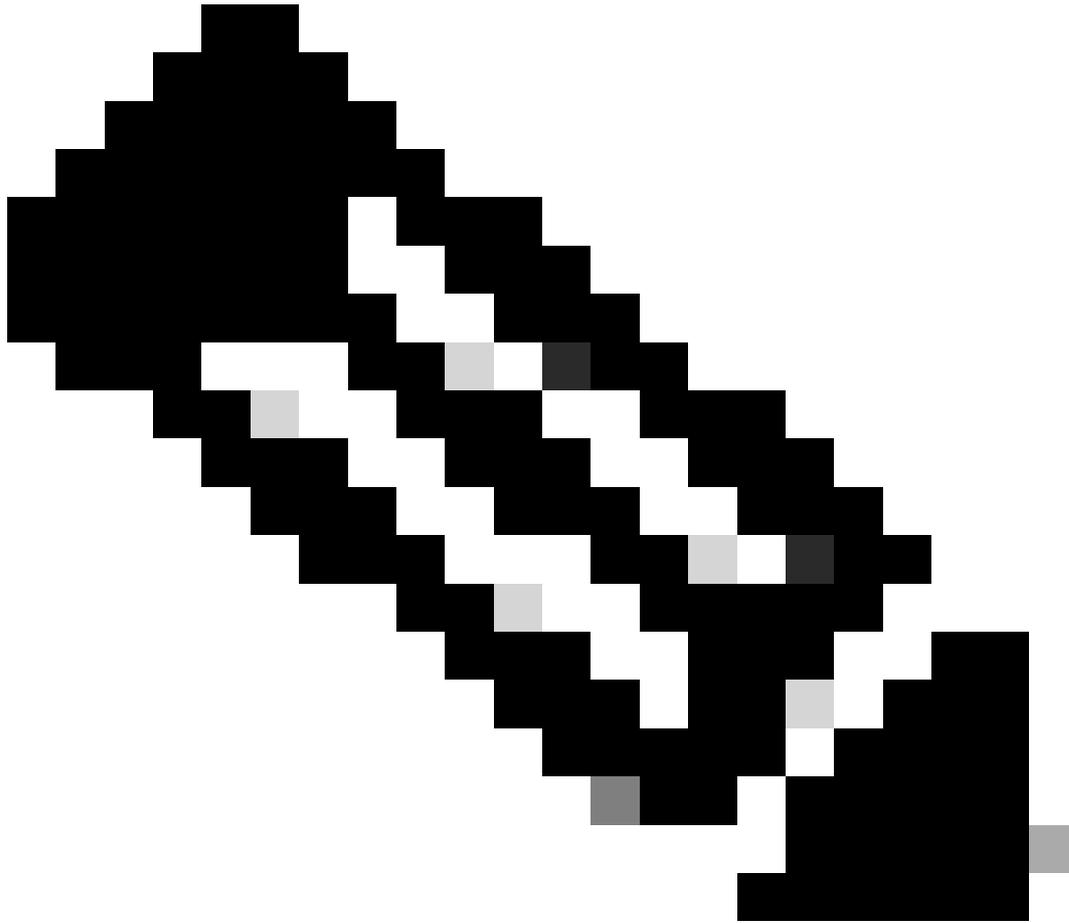
```
RTC#
router bgp 300
 neighbor 172.31.20.1 remote-as 100
 network 172.16.10.0
```

---



**참고:** RTC는 172.31.20.2와 동일한 차기 홉을 사용하여 RTA에 172.16.10.0을 광고합니다.

---

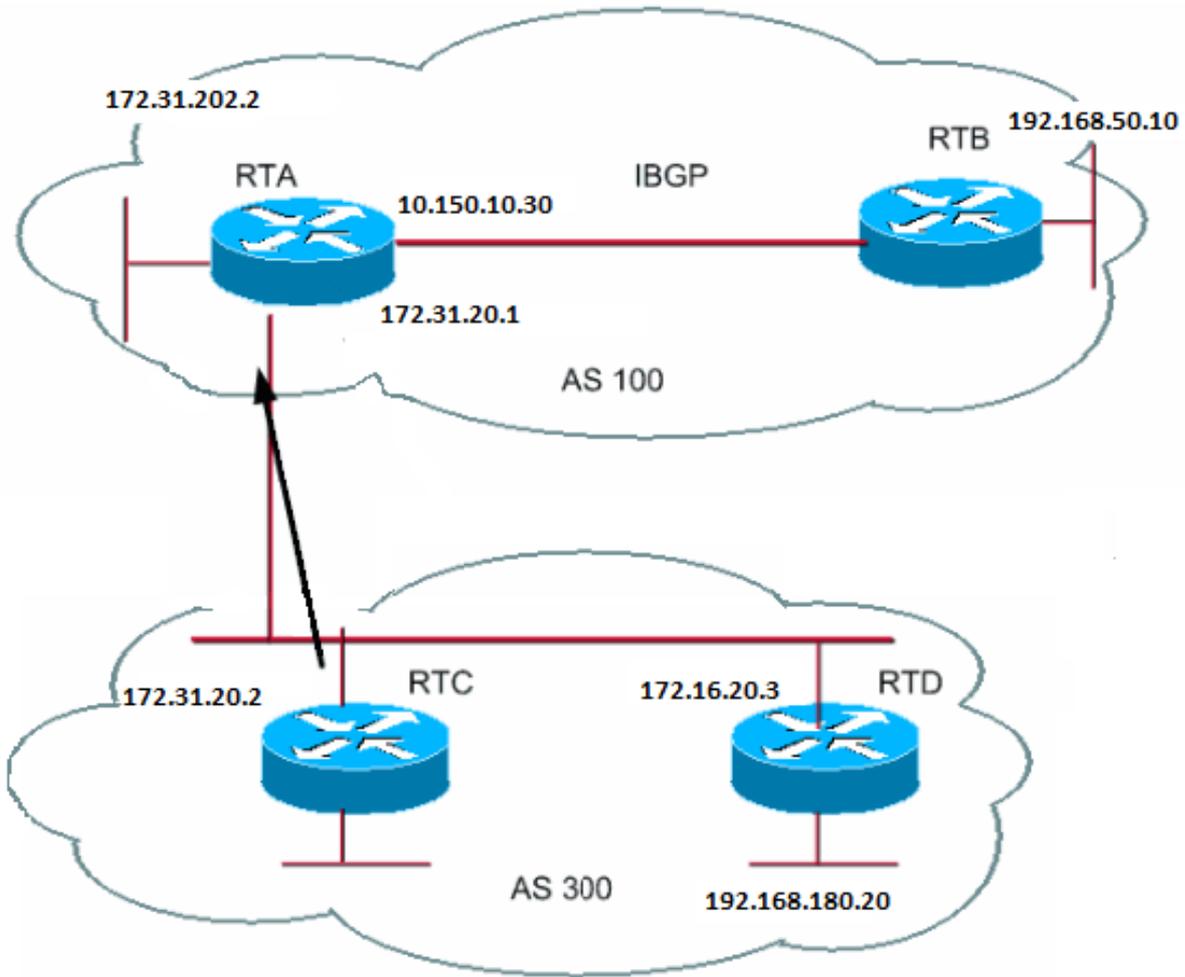


**참고:** RTA는 172.31.20.2와 동일한 차기 홉을 사용하여 RTB에 172.16.10.0을 광고합니다. eBGP next hop은 iBGP에서 전달됩니다.

---

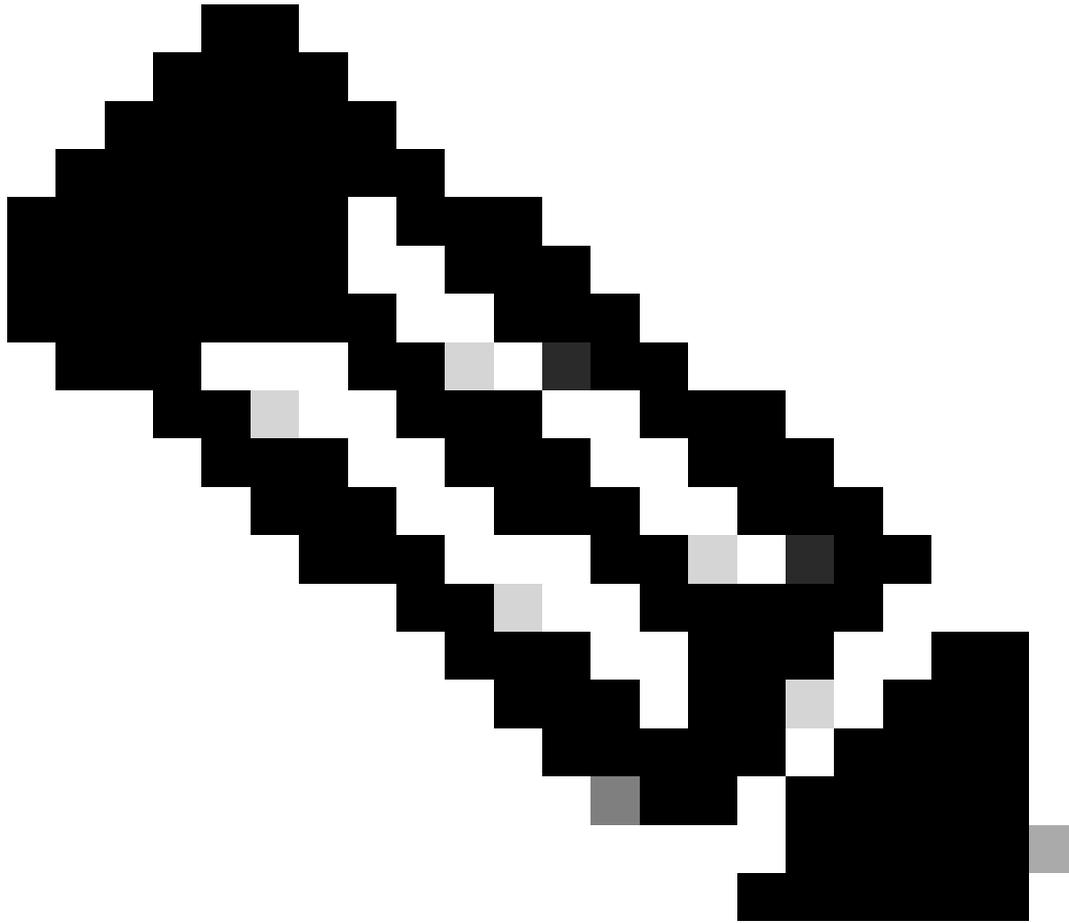
다중 액세스 및 비 브로드캐스트 다중 액세스(NBMA) 네트워크를 다룰 때는 특히 주의해야 합니다. 자세한 내용은 BGP Next Hop(Multiaccess Networks) 및 BGP Next Hop(NBMA) 섹션을 참조하십시오.

BGP Next Hop(다중 액세스 네트워크)



이 예에서는 다음 홉이 이더넷과 같은 다중 액세스 네트워크에서 어떻게 동작하는지 보여줍니다.

AS300의 RTC 및 RTD가 OSPF를 실행한다고 가정합니다. RTC는 RTA와 함께 BGP를 실행합니다. RTC는 172.16.20.3을 통해 네트워크 192.168.180.20에 연결할 수 있습니다. RTC가 192.168.180.20과 관련하여 RTA에 BGP 업데이트를 전송하면 RTC는 다음 홉 172.16.20.3으로 사용합니다. RTC는 자체 IP 주소 172.31.20.2를 사용하지 않습니다. RTC, RTC 및 RTD 간의 네트워크는 다중 액세스 네트워크이므로 RTC는 이 주소를 사용합니다. 192.168.180.20에 도달하기 위해 다음 홉으로 RTD를 사용하는 RTA는 RTC를 통한 추가 홉보다 더 합리적입니다.

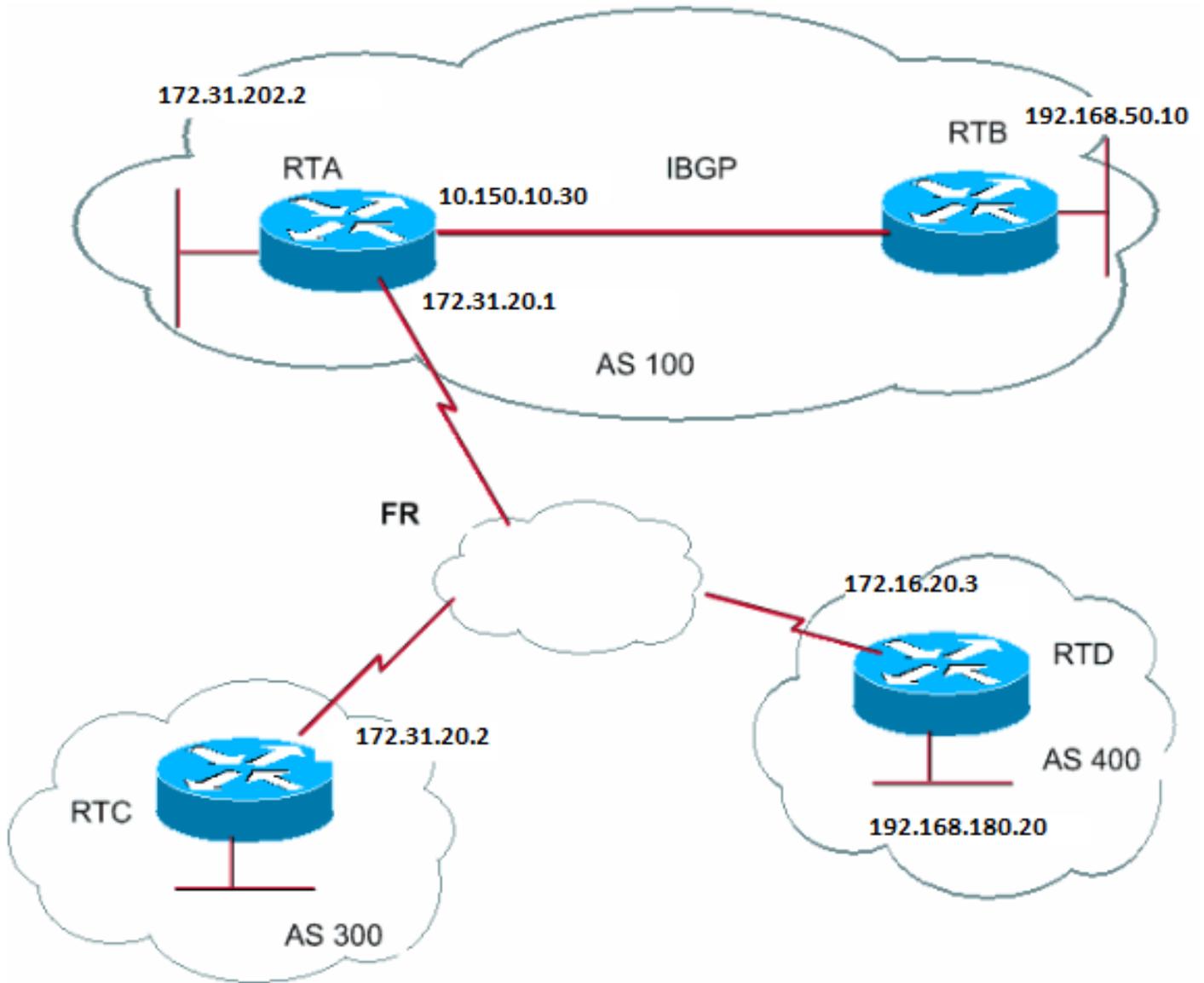


참고: RTC는 다음 홉 172.16.20.3과 함께 RTA에 192.168.180.20을 광고합니다.

---

RTA, RTC, RTD에 대한 공통 매체가 다중 액세스가 아닌 NBMA인 경우 추가 합병증이 발생한다.

BGP Next Hop(NBMA)



다이어그램에서 공통 미디어는 클라우드로 표시됩니다. 공통 미디어가 프레임 릴레이 또는 NBMA 클라우드인 경우 정확한 동작은 마치 이더넷을 통해 연결된 것과 같습니다. RTC는 RTA에 192.168.180.20을 광고하며 다음 홉은 172.16.20.3입니다.

문제는 RTA가 RTD에 대한 PVC(Direct Permanent Virtual Circuit)가 없고 다음 홉에 도달할 수 없다는 점이다. 이 경우 라우팅이 실패합니다.

명령에서 next-hop-self이 상황을 치료합니다.

Next-hop-self 명령

NBMA(BGP Next Hop) 예에서처럼 next hop이 있는 경우 이 명령을 사용할 수 next-hop-self 있습니다. 구문은 다음과 같습니다.

<#root>

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

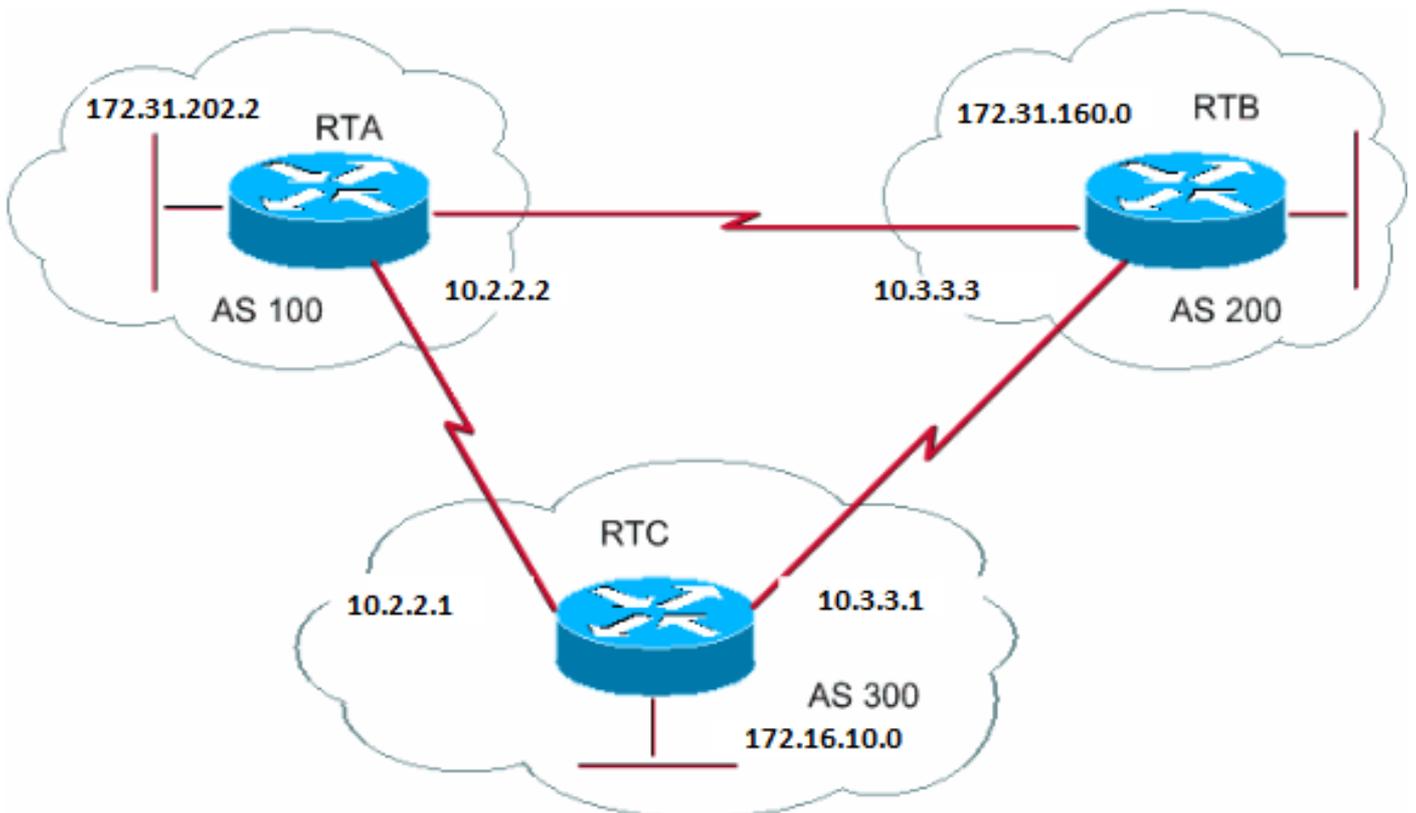
이 next-hop-self 명령을 사용하면 BGP가 특정 IP 주소를 다음 홉으로 사용하도록 강제할 수 있습니다.

BGP Next Hop(NBMA) 예의 경우 이 컨피그레이션에서 문제를 해결합니다.

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

RTC는 192.168.180.20에 172.31.20.2와 동일한 차기 홉을 광고합니다.

BGP 백도어



이전 다이어그램에서 RTA와 RTC는 eBGP를 실행합니다. RTB 및 RTC는 eBGP를 실행합니다. RTA 및 RTB는 RIP, IGRP 또는 다른 프로토콜과 같은 일종의 IGP를 실행합니다. 정의상 eBGP 업데이트의 거리는 20이며, 이는 IGP 거리보다 짧습니다. 기본 거리는 다음과

같습니다.

- RIP의 경우 120
- IGRP의 경우 100
- EIGRP의 경우 90
- OSPF의 경우 110

RTA는 두 가지 라우팅 프로토콜을 통해 172.31.160.0에 대한 업데이트를 수신합니다.

- 거리가 20인 eBGP
- 거리가 20보다 큰 IGP

기본적으로 BGP에는 다음과 같은 거리가 있습니다.

- 외부 거리 - 20
- 내부 거리 - 200
-

그러나 이 명령을 사용하여 distance 기본 거리를 변경할 수 있습니다.

```
<#root>
```

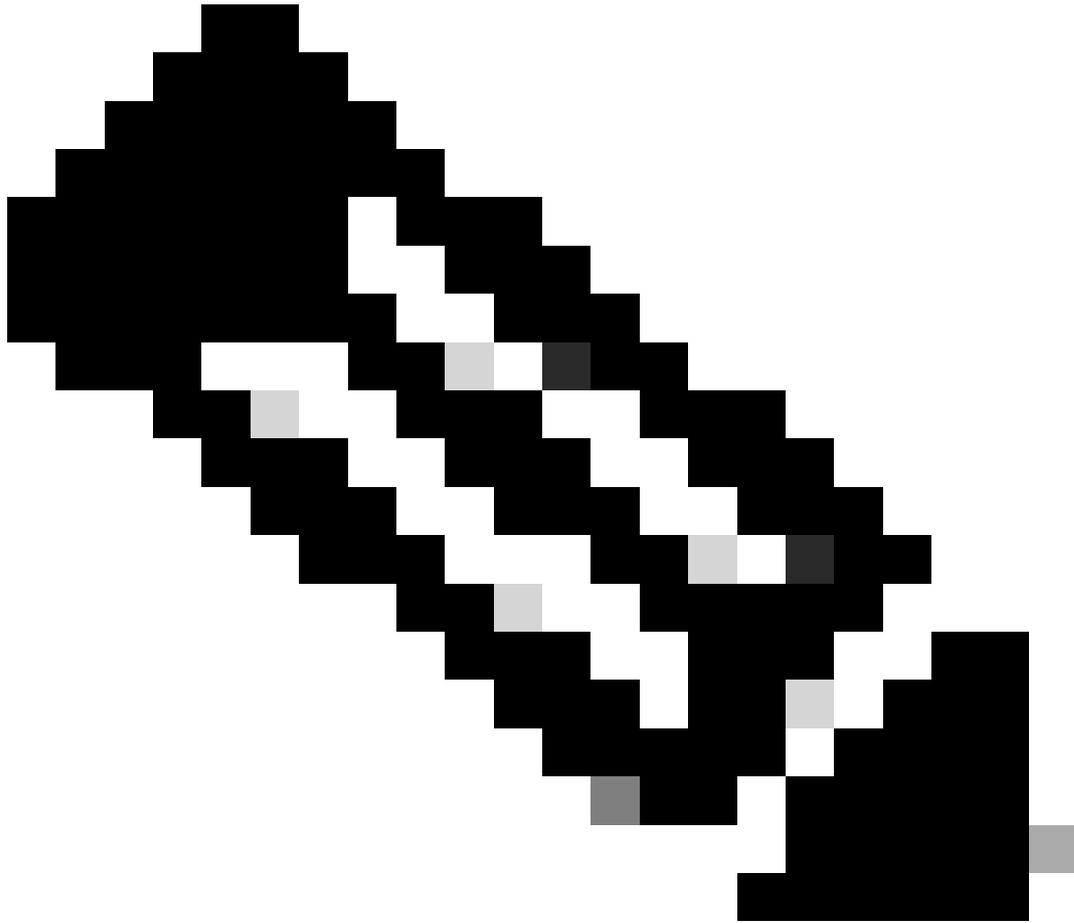
```
distance bgp <external-distance> <internal-distance> <local-distance>
```

RTA는 더 짧은 거리 때문에 RTC를 통해 eBGP를 선택합니다.

RTA가 RTB(IGP)를 통해 172.31.160.0에 대해 학습하도록 하려면 두 가지 옵션을 사용할 수 있습니다.

- 

eBGP의 외부 거리 또는 IGP 거리를 변경합니다.



참고: 이 변경은 권장되지 않습니다.

---

- 

BGP 백도어를 사용합니다.

BGP 백도어는 IGP 경로를 기본 경로로 설정합니다.

networkaddressbackdoor [명령](#)을 실행합니다.

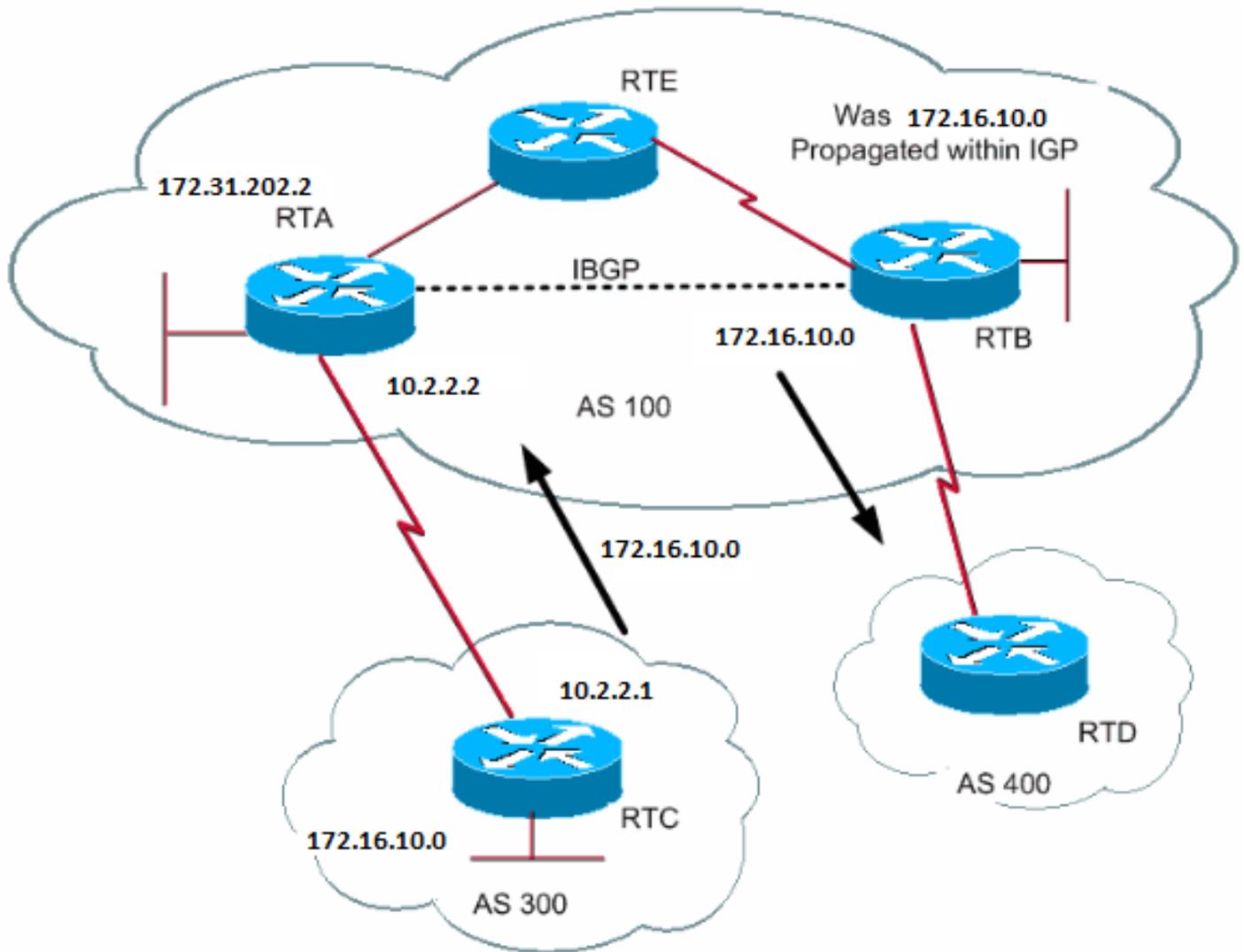
구성된 네트워크는 IGP를 통해 연결하려는 네트워크입니다. BGP의 경우 이 네트워크는 로컬로 할당된 네트워크와 동일한 처리를 받습니다. 단, BGP 업데이트에서는 이 네트워크를 광고하지 않습니다.

```
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
```

네트워크 172.31.160.0은 로컬 항목으로 처리되지만 일반 네트워크 항목으로 광고되지 않습니다.

RTA는 거리 90의 EIGRP를 통해 RTB에서 172.31.160.0을 학습합니다. 또한 RTA는 거리 20의 eBGP를 통해 RTC에서 주소를 학습합니다. 일반적으로 eBGP는 기본 설정이지만, `network backdoor` 명령 때문에 EIGRP가 기본 설정입니다.

Synchronization(동기화)



동기화에 대한 논의 전에 이 시나리오를 살펴보십시오. AS300의 RTC는 172.16.10.0에 대한 업데이트를 전송합니다. RTA 및 RTB는 iBGP를 실행하므로 RTB는 업데이트를 받고 다음 홉 10.2.2.1을 통해 172.16.10.0에 도달할 수 있습니다. 다음 홉은 iBGP를 통해 전달됩니다. 다음 홉에 도달하기 위해 RTB는 트래픽을 RTE로 전송해야 합니다.

RTA가 네트워크 172.16.10.0을 IGP로 재배포하지 않았다고 가정합니다. 이 시점에서 RTE는 172.16.10.0도 존재한다는 사실을 전혀 알지 못한다.

RTB가 AS400에 RTB가 172.16.10.0에 도달할 수 있다고 알리기 시작하면, RTD에서 대상 172.16.10.0을 가진 RTB로 들어오는 트래픽은 RTE에서 유입되고 삭제됩니다.

As에서 다른 AS의 트래픽을 세 번째 AS로 전달하는 경우 BGP는 AS의 모든 라우터가 IGP를 통해 경로를 학습하기 전에 경로를 광고해서는 안 됩니다. BGP는 IGP가 AS 내에서 경로를 전파할 때까지 기다립니다. 그런 다음 BGP는 외부 피어에 경로를 광고합니다.

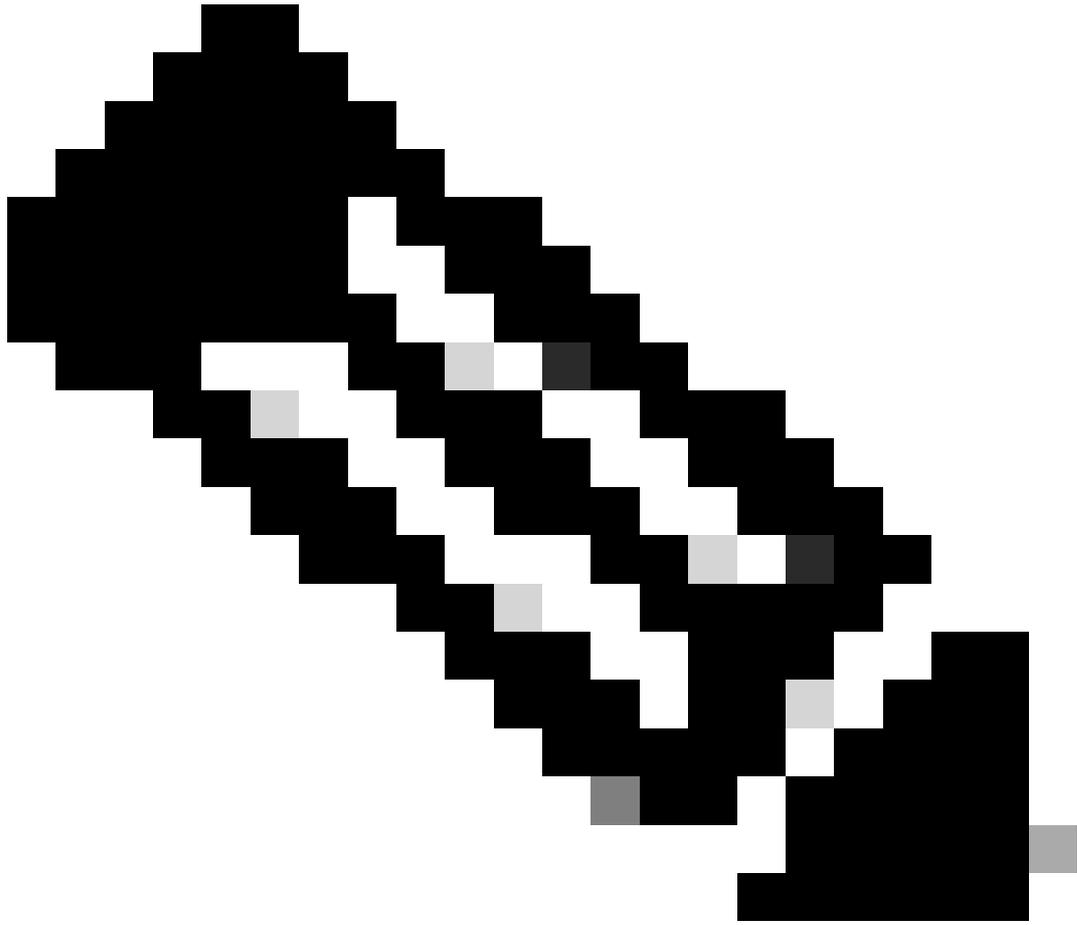
이 섹션의 예에서 RTB는 IGP를 통해 약 172.16.10.0을 듣기 위해 대기합니다. 그런 다음 RTB가 업데이트를 RTD로 보내기 시작합니다. RTB에서 172.16.10.0을 가리키는 고정 경로를 추가하면 RTB가 IGP가 정보를 전파했다고 생각하도록 할 수 있습니다. 다른 라우터가 172.16.10.0에 도달할 수 있는지 확인합니다.

#### 동기화 사용 안 함

경우에 따라 동기화가 필요하지 않습니다. 다른 AS의 트래픽을 AS를 통해 전달하지 않을 경우 동기화를 비활성화할 수 있습니다. AS의 모든 라우터가 BGP를 실행하는 경우 동기화를 비활성화할 수도 있습니다. 이 기능을 비활성화하면 IGP에서 더 적은 수의 경로를 전달할 수 있고 BGP가 더 빨리 통합되도록 할 수 있습니다.

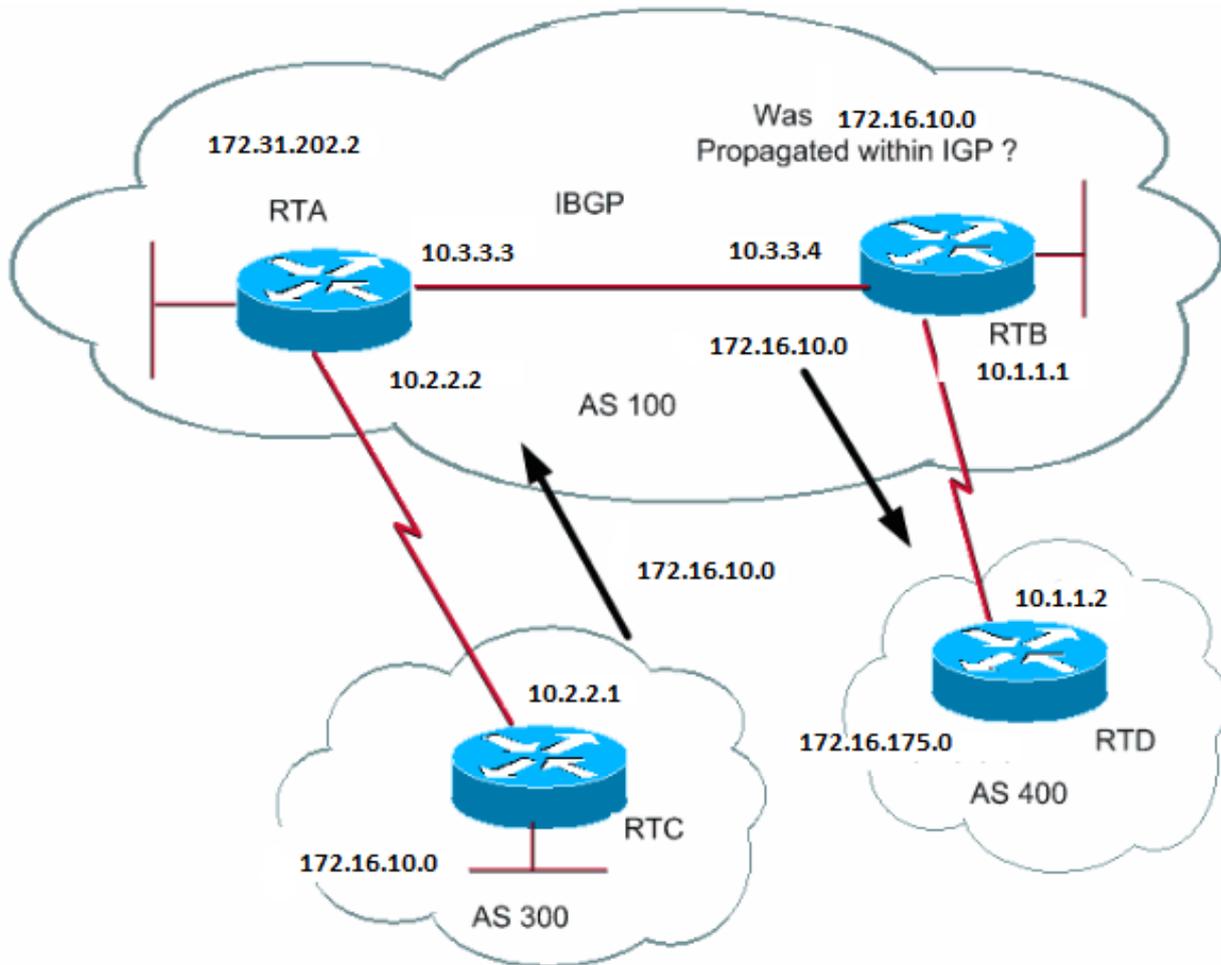
동기화를 비활성화하는 것은 자동으로 수행되지 않습니다. AS의 모든 라우터가 BGP를 실행하는데 IGP를 전혀 실행하지 않으면 라우터가 알 수 있는 방법이 없습니다. 라우터가 외부 피어에 경로를 전송하기 전에 특정 경로에 대한 IGP 업데이트를 무한정 기다립니다. 라우팅이 올바르게 작동하려면 이 경우 수동으로 동기화를 비활성화해야 합니다.

```
router bgp 100
no synchronization
```



참고: clear ip bgp address 명령을 실행하여 세션을 재설정해야 합니다.

---



```

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

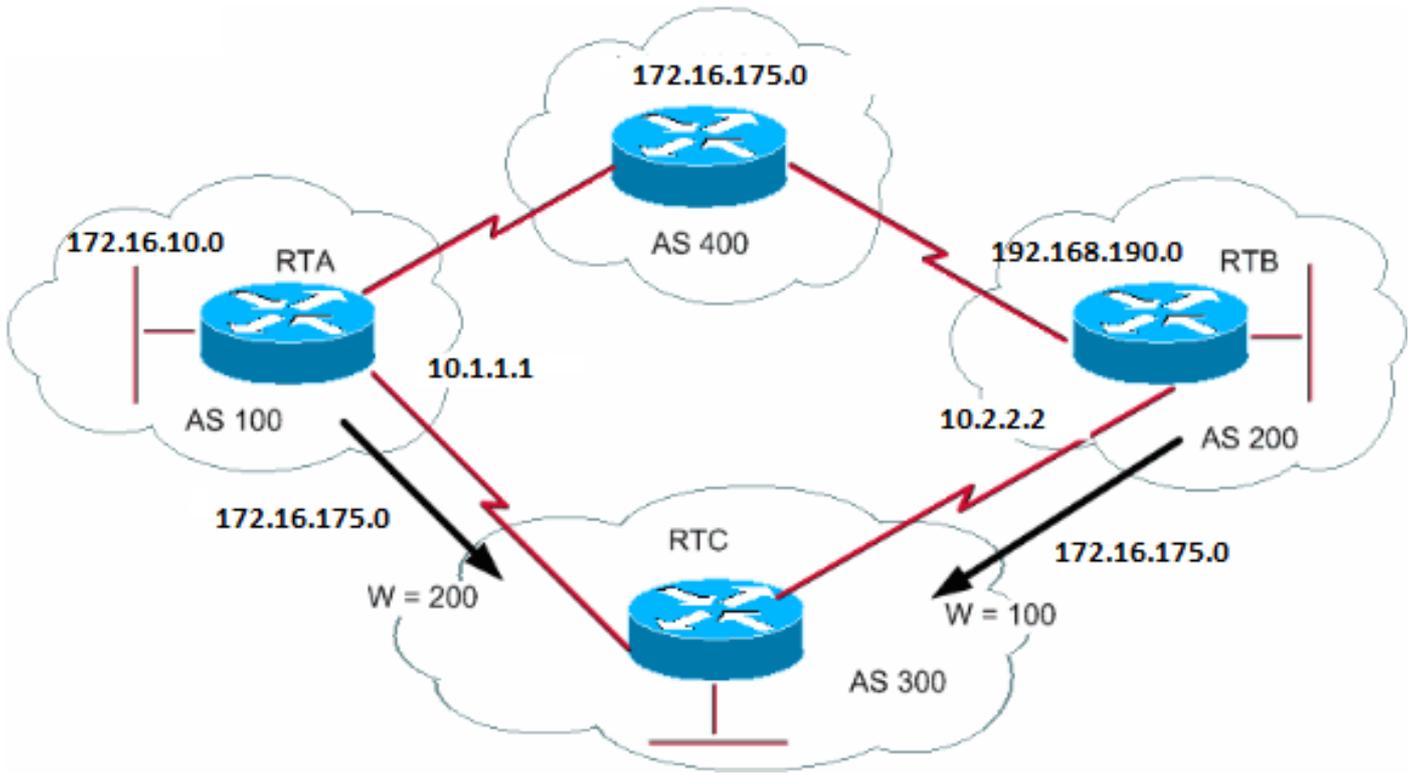
```

```

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

```

가중치 속성



가중치 속성은 Cisco 정의 속성입니다. 이 특성은 가중치를 사용하여 최상의 경로를 선택합니다. 가중치는 라우터에 로컬로 할당됩니다. 이 값은 특정 라우터에만 적용됩니다. 값은 경로 업데이트를 통해 전파되거나 전달되지 않습니다. 가중치는 0에서 65,535 사이의 숫자일 수 있습니다. 라우터가 시작하는 경로의 가중치는 기본적으로 32,768이며, 다른 경로의 가중치는 0입니다.

가중치가 더 높은 경로는 동일한 대상에 대한 경로가 여러 개 있을 때 우선권을 갖습니다. 이 섹션의 예를 참조하십시오. RTA는 AS4를 통해 네트워크 172.16.0.0에 대해 학습했습니다. RTA가 업데이트를 RTC에 전파합니다. RTB는 AS4를 통해 네트워크 172.16.0.0에 대해서도 학습했습니다. RTB가 업데이트를 RTC에 전파합니다. RTC는 이제 172.16.0.0에 도달하는 두 가지 방법을 가지고 있으며 어떤 방법으로 갈지 결정해야 합니다. RTA에서 가져온 RTC의 업데이트 가중치를 설정하여 RTB에서 가져온 업데이트 가중치보다 더 큰 경우, RTC에서 RTA를 다음 홉으로 사용하여 172.16.0.0에 도달하도록 강제합니다. 여러 방법으로 이 가중치 집합을 얻을 수 있습니다.

- neighbor 명령을 사용합니다.

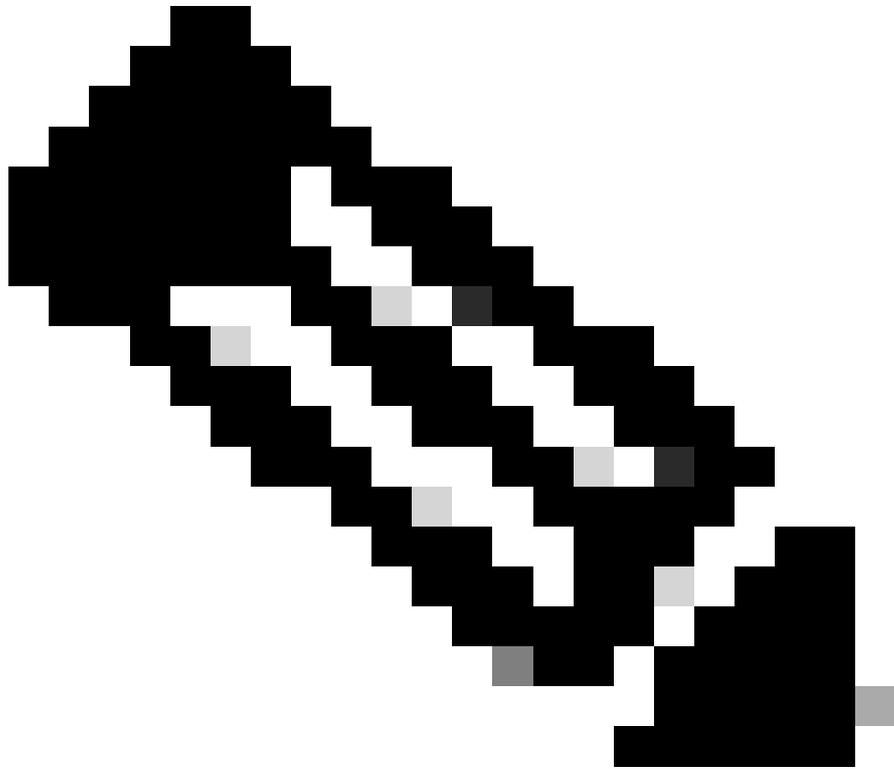
네이버 {ip-address|peer-group} 가중치 <weight>

- AS\_PATH 액세스 목록을 사용합니다.

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

◦  
**neighbor <ip-address>filter-list <access-list-number>weight <weight>**

---



**참고:** 일부 시나리오에서는 일부 소프트웨어 버전에서 사용할 수 없는 명령이 거의 없을 수 있습니다.

---

•

경로 맵을 사용합니다.

```
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.
```

가중치가 높은 RTA는 다음 홉으로 선호도가 높다.

IP AS\_PATH 및 필터 목록을 사용하여 동일한 결과를 얻을 수 있습니다.

```
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$
...
```

또한 경로 맵을 사용하여 동일한 결과를 얻을 수 있습니다.

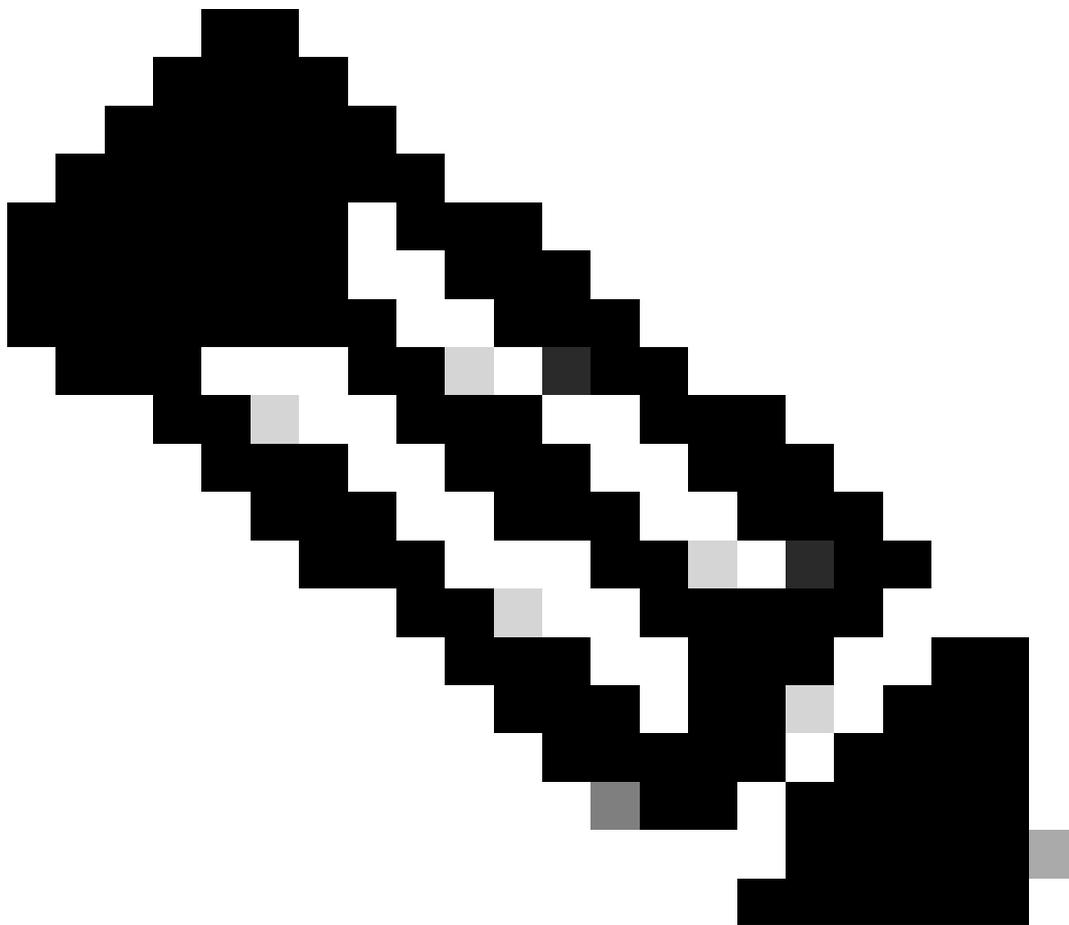
```
RTC#
router bgp 300
  neighbor 10.1.1.1 remote-as 100
  neighbor 10.1.1.1 route-map setweightin in
  neighbor 10.2.2.2 remote-as 200
  neighbor 10.2.2.2 route-map setweightin in
  ...
ip as-path access-list 5 permit ^100$
...
```

```
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
```

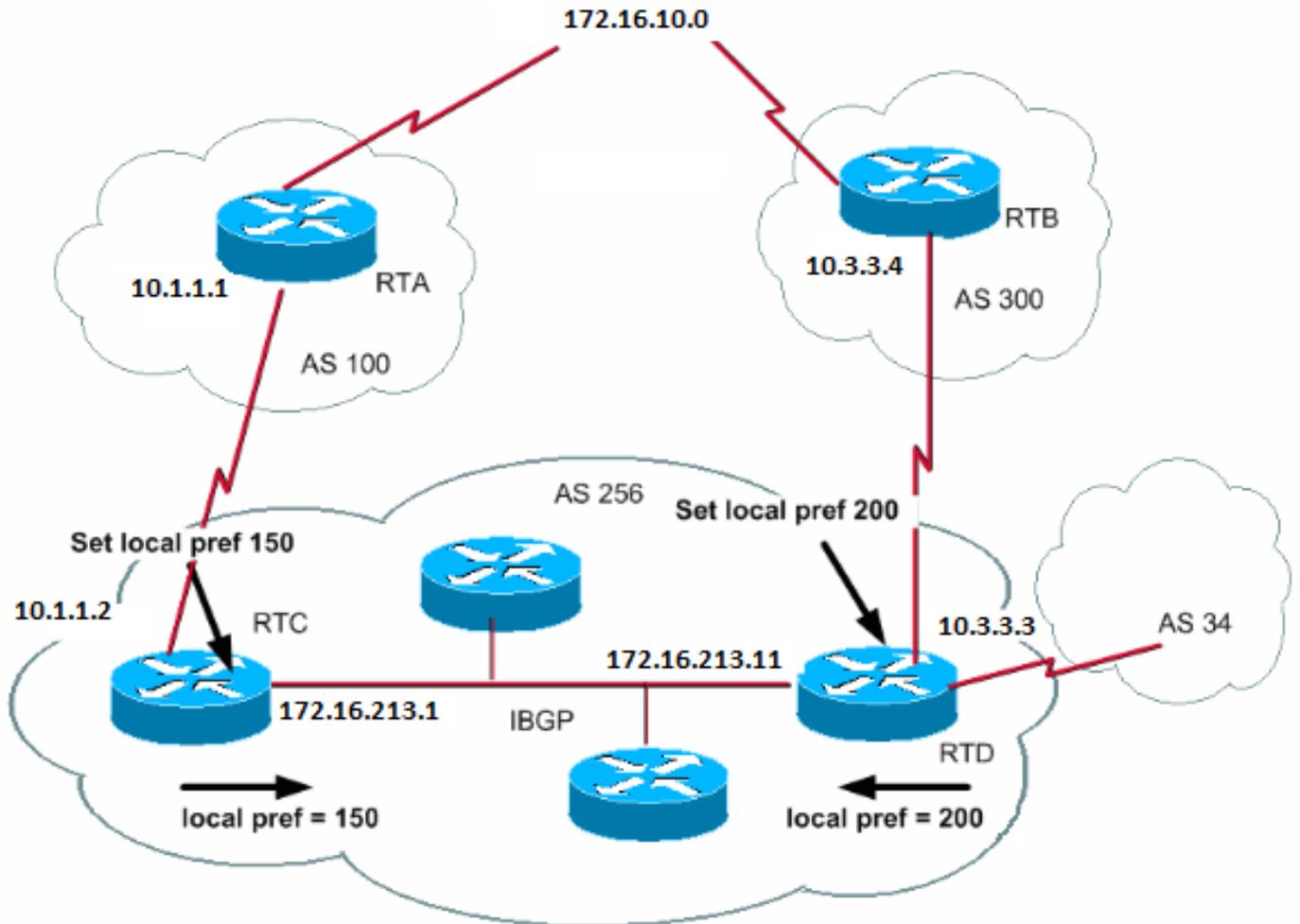
*!--- Anything else has weight 100.*



참고: IGP 경로를 백업으로 사용하는 MPLS VPN BGP 경로를 선호하도록 가중치를 수정할 수 있습니다.

---

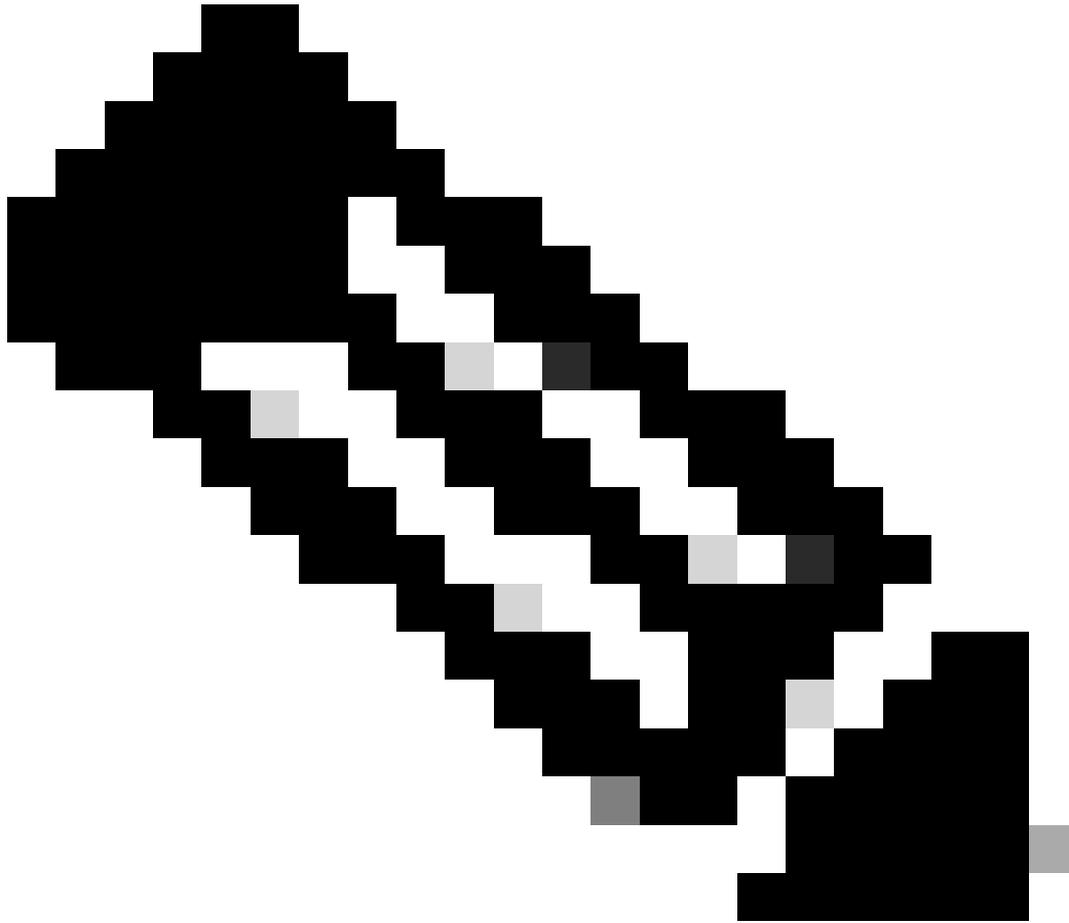
## 로컬 기본 설정 속성



로컬 환경 설정은 AS에 어떤 경로가 특정 네트워크에 연결하기 위해 AS를 종료하는 것을 선호하는지 나타내는 것입니다. 로컬 선호도가 높은 경로가 더 선호됩니다. 로컬 기본 설정의 기본값은 100입니다.

로컬 라우터에만 관련된 가중치 속성과 달리 로컬 환경 설정은 동일한 AS에서 라우터가 교환하는 속성입니다.

`bgp default local-preference value` 명령 문제와 함께 로컬 환경 설정을 설정합니다. 이 섹션의 예제와 같이 경로 맵을 사용하여 로컬 환경 설정을 지정할 수도 있습니다.



**참고:** 변경 사항을 고려하려면 소프트 리셋(즉, 라우터의 bgp 프로세스 지우기)을 수행해야 합니다. bgp 프로세스를 지우려면 `clear ip bgp [soft][in/out]` 명령을 사용합니다. 여기서 `soft`는 소프트 재설정을 나타내며 세션을 해제하지 않으며 `[in/out]`은 인바운드 또는 아웃바운드 컨피그레이션을 지정합니다. `in/out`을 지정하지 않으면 인바운드 세션과 아웃바운드 세션이 모두 재설정됩니다.

`bgp default local-preference` 명령은 동일한 AS의 피어로 이동하는 라우터의 업데이트에 대한 로컬 기본 설정을 설정합니다. 이 섹션의 다이어그램에서 AS256은 조직의 서로 다른 두 축에서 약 172.16.10.0에 대한 업데이트를 수신합니다. 로컬 환경 설정을 사용하면 해당 네트워크에 연결하기 위해 AS256을 종료할 방법을 결정할 수 있습니다. RTD가 종료 지점 기본 설정이라고 가정합니다. 이 컨피그레이션은 AS300에서 200으로 제공되는 업데이트 및 AS100에서 150으로 제공되는 업데이트에 대한 로컬 환경 설정을 지정합니다.

```
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
```

이 컨피그레이션에서 RTC는 모든 업데이트의 로컬 환경 설정을 150으로 설정합니다. 동일한 RTD는 모든 업데이트의 로컬 기본 설정을 200으로 설정합니다. AS256 내에서 지역 선호도의 교환이 있습니다. 따라서 RTC와 RTD는 모두 네트워크 172.16.10.0이 AS100보다 AS300에서 업데이트될 때 로컬 선호도가 더 높다는 것을 알고 있습니다. 해당 네트워크를 대상으로 하는 AS256의 모든 트래픽은 RTD를 종료 지점으로 하여 전송합니다.

경로 맵을 사용하면 유연성이 향상됩니다. 이 섹션의 예에서, RTD가 수신하는 모든 업데이트는 업데이트가 RTD에 도달하면 로컬 기본 설정 200으로 태그가 지정됩니다. AS34에서 제공되는 업데이트도 로컬 기본 설정인 200으로 태그가 지정됩니다. 이 태그는 불필요할 수 있습니다. 따라서 경로 맵을 사용하여 특정 로컬 기본 설정으로 태그를 지정해야 하는 특정 업데이트를 지정할 수 있습니다. 예를 들면 다음과 같습니다.

```
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$
...

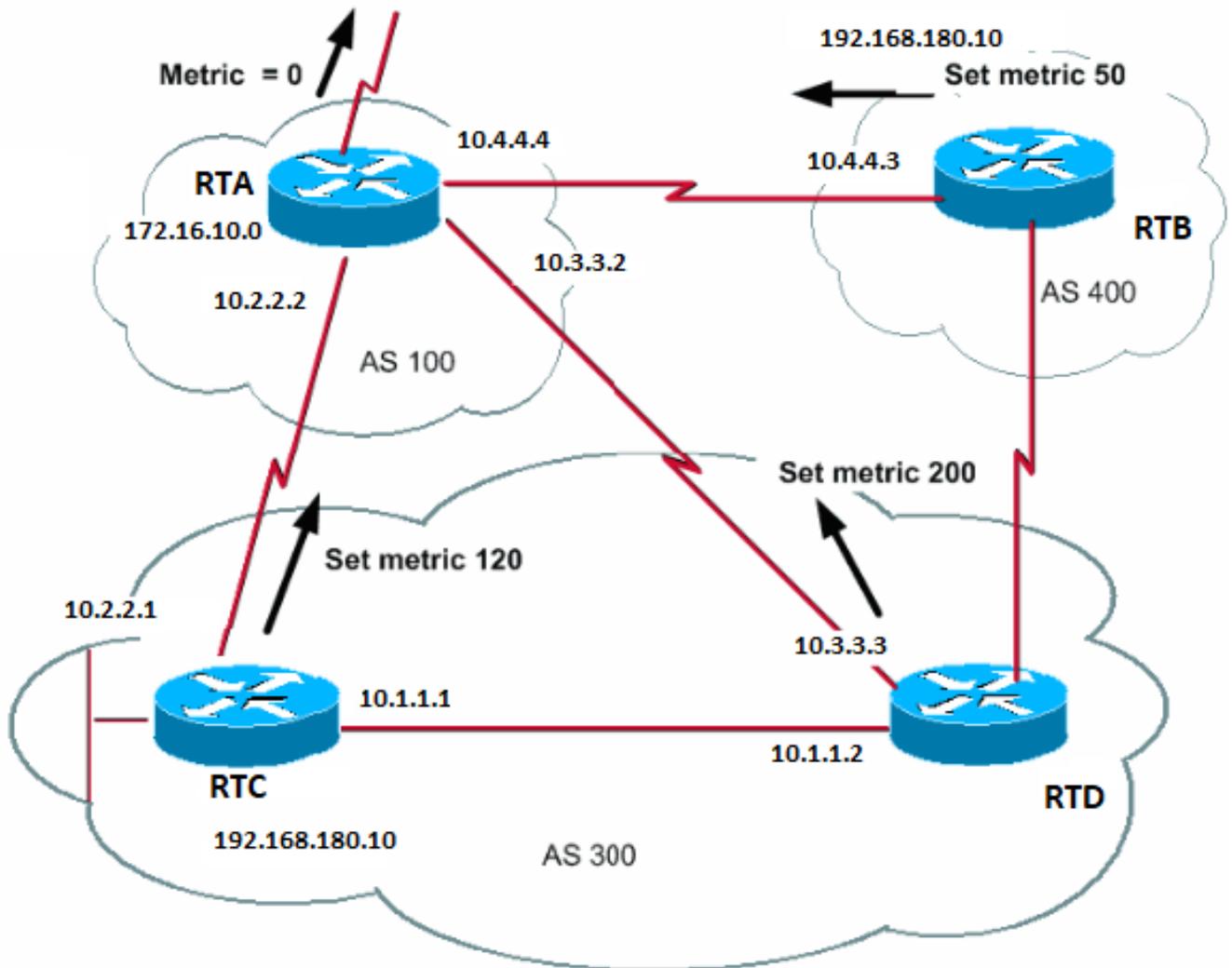
route-map setlocalin permit 10
 match as-path 7
 set local-preference 200

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

이 컨피그레이션에서는 AS300에서 제공되는 모든 업데이트의 로컬 환경 설정이 200입니다. AS34에서 오는 업데이트와 같은 다른 모든 업데이트는 150의 값을 가진다.

메트릭 특성

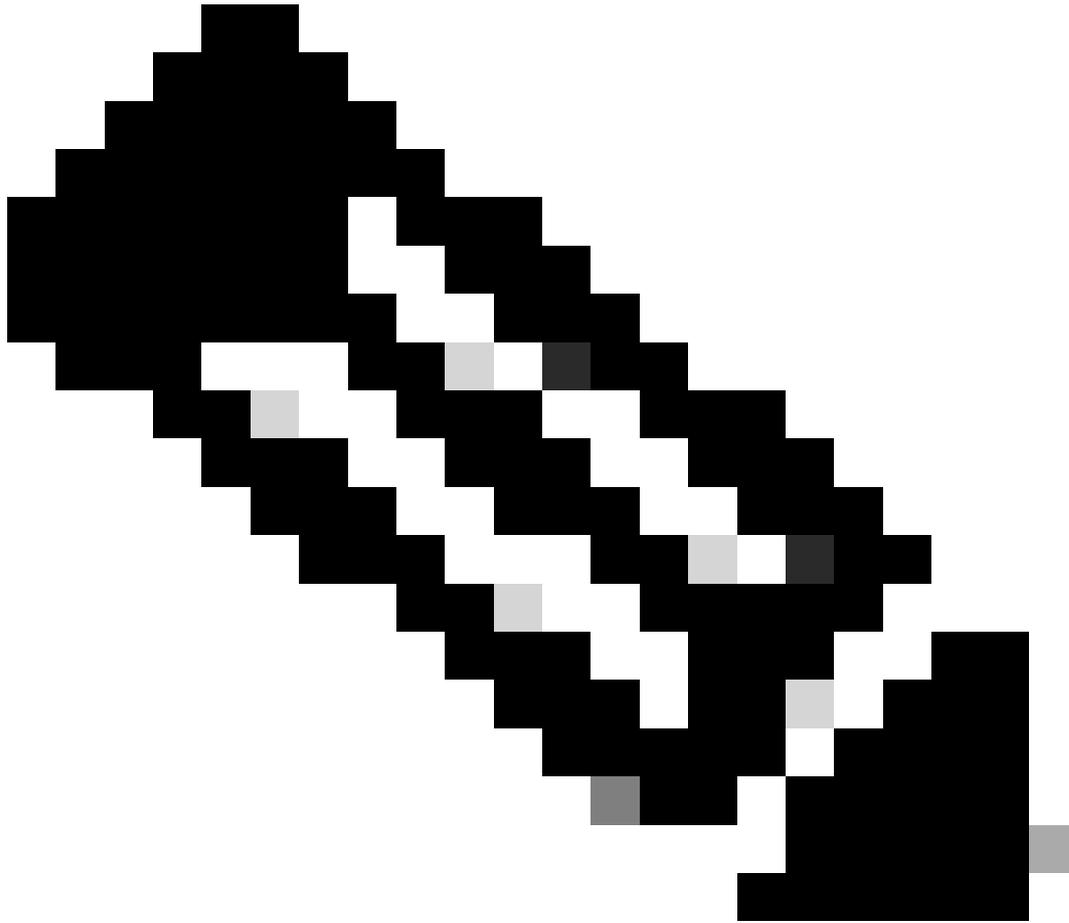
## METRIC (MULTI\_EXIT\_DISC) (INTER\_AS)



메트릭 특성에는 MULTI\_EXIT\_DISCRIMINATOR, MED(BGP4) 또는 INTER\_AS(BGP3)라는 이름도 있습니다. 이 특성은 AS로의 경로 기본 설정에 대한 외부 인접 디바이스에 대한 힌트입니다. 이 특성은 AS에 여러 진입점이 있을 때 특정 경로에 도달하는 방법으로 다른 AS에 영향을 줄 수 있는 동적 방법을 제공합니다. 더 낮은 메트릭 값이 더 선호됩니다.

로컬 선호도와 달리 메트릭은 AS 간에 교환됩니다. 메트릭은 AS로 전달되지만 AS를 벗어나지 않습니다. 업데이트가 특정 메트릭을 사용하여 AS에 들어가면 해당 메트릭은 AS 내에서 결정을 내리는 데 사용됩니다. 동일한 업데이트가 세 번째 AS로 전달되면 해당 메트릭은 0으로 돌아갑니다. 이 섹션의 다이어그램은 메트릭 집합을 보여줍니다. 메트릭 기본값은 0입니다.

라우터가 다른 방향을 수신하지 않는 한 라우터는 동일한 AS의 인접 디바이스에서 오는 경로에 대한 메트릭을 비교합니다. 라우터가 서로 다른 AS에서 온 인접 디바이스의 메트릭을 비교하려면 라우터에서 `bgp always-compare-med` 특수 [컨피그레이션 명령](#)을 실행해야 합니다.



**참고:** MED(multi-exit discriminator) 기반 경로 선택에 영향을 줄 수 있는 BGP 컨피그레이션 명령은 두 개입니다. 이 명령은 `bgp deterministic-med` 명령과 `bgp always-compare-med` 명령입니다. `bgp deterministic-med` 명령의 문제는 서로 다른 피어가 동일한 AS에서 광고할 때 경로 선택 시 MED 변수를 비교하도록 보장합니다. `bgp always-compare-med` 명령의 문제는 서로 다른 AS의 인접 디바이스의 경로에 대한 MED의 비교를 보장합니다. `bgp always-compare-med` 명령은 여러 통신 사업자나 기업이 MED 설정 방법에 대한 일관된 정책에 동의할 때 유용합니다. BGP Deterministic-Med 명령이 BGP Always-Compare-Med 명령과 어떻게 다른지를 참조하여 이러한 명령이 BGP 경로 선택에 어떤 영향을 미치는지 알아보십시오.

이 섹션의 다이어그램에서 AS100은 RTC, RTD 및 RTB의 세 가지 라우터를 통해 네트워크 192.168.180.10에 대한 정보를 가져옵니다. RTC 및 RTD는 AS300에 있고 RTB는 AS400에 있습니다.

이 예에서는 명령 `bgp bestpath as-path ignore`에 의한 RTA의 [AS-Path 비교](#)가 무시됩니다. BGP가 경로 비교를 위해 다음 속성으로 넘어가도록 구성됩니다(이 경우 metric 또는 MED). 명령을 생략하면 BGP는 라우터 RTC에서 가장 짧은 AS-Path를 가진 경로 192.168.180.10을 설치할 수 있습니다.

RTC에서 오는 메트릭을 120으로 설정하고, RTD에서 오는 메트릭을 200으로 설정하고, RTB에서 오는 메트릭을 50으로 설정했다고 가정합니다. 기본적으로 라우터는 동일한 AS의 네이버에서 가져온 메트릭을 비교합니다. 따라서 RTA는 RTC에서 오는 메트릭과 RTD에서 오는 메트릭만 비교할 수 있습니다. RTA는 120이 200보다 작기 때문에 RTC를 가장 좋은 차기 홉으로 선택합니다. RTC와 RTB가 다른 AS에 있으므로 RTA가 메트릭 50을 사용하여 RTB에서 업데이트를 받는 경우 RTA는 메트릭을 120과 비교할 수 없습니다. RTA는 일부 다른 특성을 기준으로 선택해야 합니다.

RTA가 메트릭을 비교하도록 하려면 RTA에서 `bgp always-compare-med 명령`을 실행해야 합니다. 이러한 컨피그레이션에서는 다음 프로세스를 보여 줍니다.

```
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
```

이러한 구성에서 RTA는 다른 모든 특성이 동일하다는 점을 고려하여 RTC를 다음 홉으로 선택합니다. 메트릭 비교에 RTB를 포함하려면 다음과 같이 RTA를 구성해야 합니다.

```
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
```

이 경우 RTA는 네트워크 192.168.180.10에 도달하기 위해 RTB를 최상의 다음 홉으로 선택합니다.

**default-metric**number 명령을 실행하면 경로를 BGP로 재배포하는 동안 메트릭을 설정할 수도 있습니다.

이 섹션의 예에서 RTB는 AS100에 정적으로 네트워크를 주입한다고 가정합니다. 구성은 다음과 같습니다.

```
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.*

#### 커뮤니티 특성

community 특성은 0~4,294,967,200 범위의 전이적, 선택적 특성입니다. 커뮤니티 특성은 특정 커뮤니티의 대상을 그룹화하고 해당 커뮤니티와 일치하는 라우팅 결정을 적용하는 방법입니다. 라우팅 결정은 수락, 선호 및 재배포가 됩니다.

경로 맵을 사용하여 커뮤니티 특성을 설정할 수 있습니다. route map set 명령의 구문은 다음과 같습니다.

```
<#root>
```

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

이 명령에서 사용할 수 있는 몇 가지 미리 정의된 잘 알려진 커뮤니티는 다음과 같습니다.

- 

**no-export** - eBGP 피어에 알리지 않습니다. 이 경로를 AS 내에 유지합니다.

- 

**no-advertise** - 이 경로를 피어, 내부 또는 외부에 알리지 않습니다.

- 

**internet** - 이 경로를 인터넷 커뮤니티에 광고합니다. 모든 라우터가 이 커뮤니티에 속합니다.

- 

**local-as** — 로컬 AS 외부의 패킷 전송을 방지하기 위해 연합 시나리오에서 사용합니다.

커뮤니티를 설정하는 경로 맵의 두 가지 예는 다음과 같습니다.

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

또는

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

additive 키워드를 설정하지 않으면 200이 이미 존재하는 이전 커뮤니티를 대체합니다. additive 키워드를 사용하면 커뮤니티에 200개가 추가됩니다. 커뮤니티 특성을 설정하더라도 이 특성은 기본적으로 네이버에 전송되지 않습니다. 인접 디바이스로 특성을 전송하려면 다음 명령을 사용해야 합니다.

<#root>

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

예를 들면 다음과 같습니다.

```
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
```

Cisco IOS Software Release 12.0 이상에서는 10진수, 16진수 및 AA:NN의 세 가지 형식으로 커뮤니티를 구성할 수 있습니다. 기본적으로 Cisco IOS Software는 이전 십진수 형식을 사용합니다. AA:NN에서 구성 및 표시하려면 **ip bgp-community new-formatcommand**를 global configuration 실행합니다. AA:NN의 첫번째 부분은 AS 번호를 나타내고, 두번째 부분은 2바이트 번호를 나타낸다.

예를 들면 다음과 같습니다.

글로벌 컨피그레이션에서 [ip bgp-community new-format 명령](#)이 없으면 **show ip bgp 10.6.0.0** 명령 문제는 커뮤니티 특성 값을 10진수 형식으로 표시합니다. 이 예에서는 커뮤니티 특성 값이 6553620으로 나타납니다.

<#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

이제 이 라우터에서 `ip bgp-community new-format` 명령을 전역적으로 실행합니다.

```
<#root>
```

```
Router#
```

```
configure terminal
```

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

```
ip bgp-community new-format
```

```
Router(config)#
```

```
exit
```

`bgp-community new-format` global 컨피그레이션 명령을 사용하면 커뮤니티 값이 AA:NN 형식으로 표시됩니다. 다음 예에서 `show ip bgp 10.6.0.0` 명령의 출력에는 값이 100:20으로 나타납니다.

```
<#root>
```

```
Router#
```

```
show ip bgp 10.6.0.0
```

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

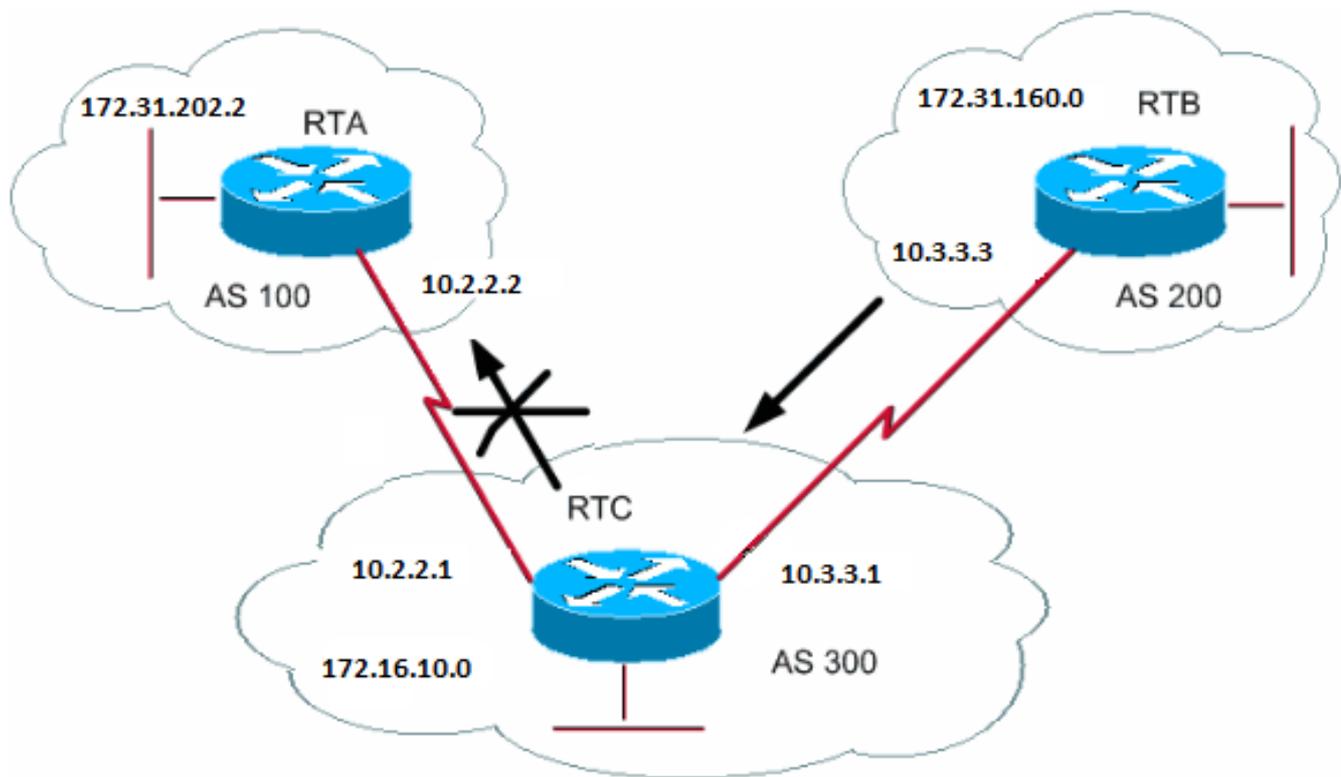
```
Community: 100:20
```

### BGP 사례 연구 3

#### BGP 필터

다양한 필터 방법을 통해 BGP 업데이트의 송수신을 제어할 수 있습니다. 경로 정보를 기반으로 하거나 경로 정보 또는 커뮤니티를 기반으로 BGP 업데이트를 필터링할 수 있습니다. 모든 방법은 같은 결과를 얻을 수 있습니다. 특정 네트워크 컨피그레이션에 따라 다른 방법 대신 한 방법을 선택할 수 있습니다.

#### 경로 필터



라우터가 학습하거나 광고하는 라우팅 정보를 제한하려면 특정 네이버에 대한 라우팅 업데이트를 사용하여 BGP를 필터링할 수 있습니다. 액세스 목록을 정의하고 인접 디바이스와의 업데이트에 액세스 목록을 적용합니다. 라우터 컨피그레이션 모드에서 다음 명령을 실행합니다.

```
<#root>
```

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

이 예에서 RTB는 네트워크 172.31.160.0을 시작하고 RTC에 업데이트를 전송합니다. RTC가 AS100에 대한 업데이트 전파를 중지하려면 액세스 목록을 정의하여 해당 업데이트를 필터링하고 RTA와의 통신 중에 액세스 목록을 적용해야 합니다.

```
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
```

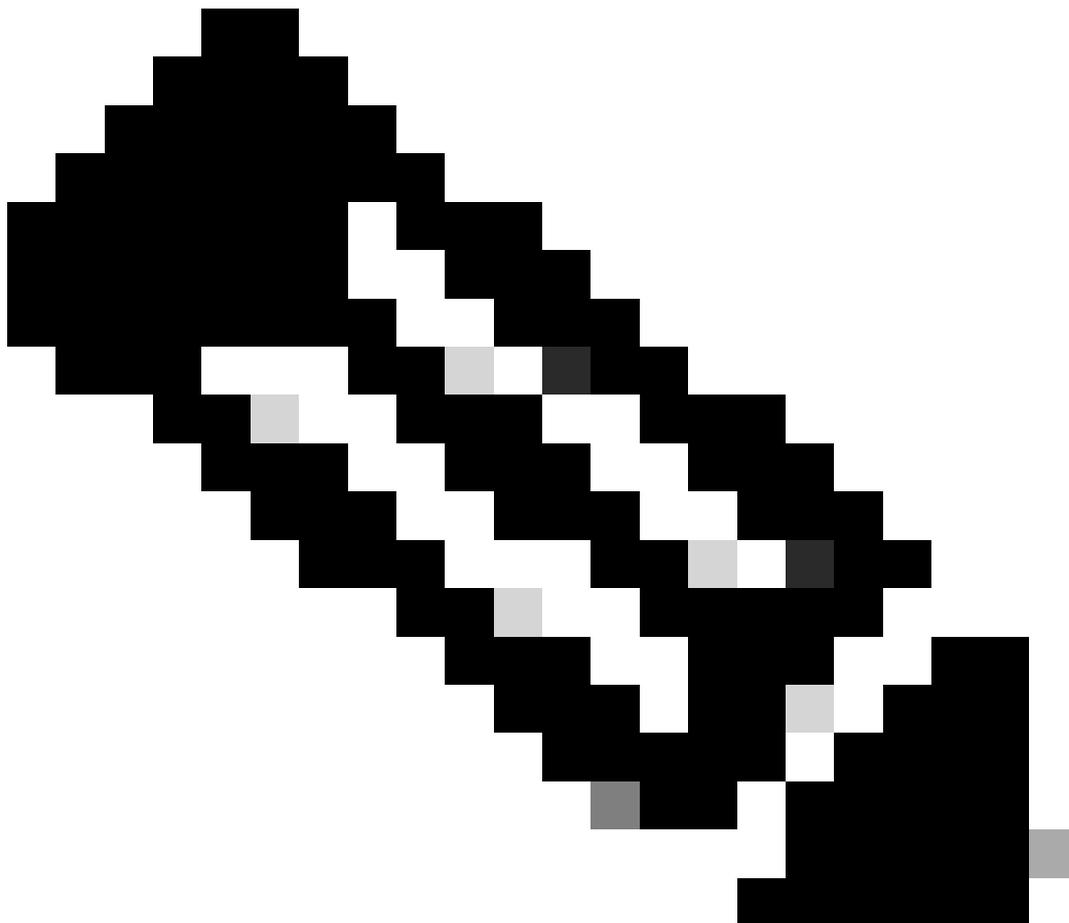
```
access-list 1 permit 0.0.0.0 255.255.255.255
```

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

일부 충돌을 일으킬 수 있는 수퍼넷을 다룰 때는 액세스 목록의 사용이 약간 까다롭습니다.

이 섹션의 예에서 RTB의 서브넷은 160.10.x.x로 다르다고 가정합니다. 목표는 업데이트를 필터링하고 192.168.160.0/8만 광고하는 것입니다.

---



---

**참고:** /8 표기법이란 IP 주소의 가장 왼쪽부터 시작되는 8비트의 서브넷 마스크를 사용한다는 의미입니다. 이 주소는 192.168.160.0 255.0.0.0에 해당합니다.

---

이 명령은 `access-list 1 permit 192.168.160.0 0.255.255.25 5192.168.160.0/8, 192.168.160.0/9` 등을 허용합니다. 업데이트를 192.168.160.0/8으로만 제한하려면 다음 형식의 확장 액세스 목록을 사용해야 합니다.

<#root>

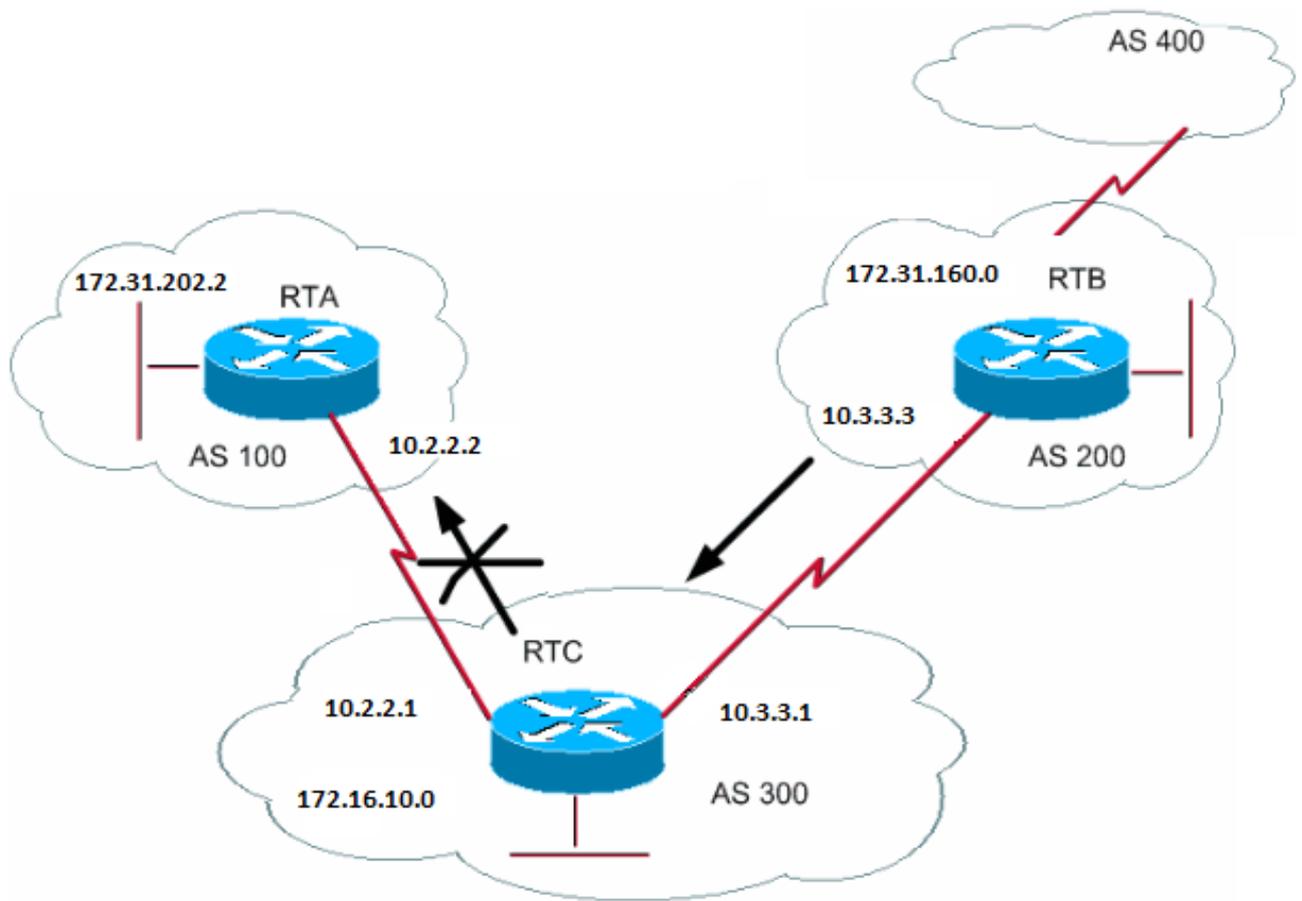
```
access-list 101 permit ip 192.168.160.0 0.255.255.255 255.0.0.0 0.0.0.0.
```

이 목록은 192.168.160.0/8만 허용합니다.

BGP 피어에서 [네트워크를 필터링하는 방법에 대한](#) 샘플 컨피그레이션은 BGP 피어에서 [하나](#) 이상의 네트워크 차단을 참조하십시오. 이 방법은 표준 및 확장 ACL(Access Control List)과 함께 `distribute-list` 명령을 사용하고 접두사 목록을 필터링하는 기능을 사용합니다.

경로 필터

경로를 필터링할 수도 있습니다.



BGP AS 경로 정보를 사용하여 수신 및 발신 업데이트 모두에 대한 액세스 목록을 지정할 수 있습니다. 이 섹션의 다이어그램에서는 AS100으로 이동하지 않도록 172.31.160.0에 대한 업데이트를 차단할 수 있습니다. 업데이트를 차단하려면 AS200에서 시작된 업데이트의 AS100으로의 전송을 차단하는 RTC에서 액세스 목록을 정의합니다. 다음 명령을 실행합니다.

<#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}
```

다음 예에서는 RTC에서 RTA로 172.31.160.0에 대한 업데이트를 전송하지 않습니다.

```
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 .*
```

이 access-list 1의 명령은 200으로 시작하고 200으로 끝나는 경로 정보로 모든 업데이트를 거부합니다. 명령에서 ^200\$는 "정규식"입니다. 여기서 ^은 "시작 문자"를 의미하고 \$는 "끝 문자"를 의미합니다. RTB는 200으로 시작하고 200으로 끝나는 경로 정보와 함께 172.31.160.0에 대한 업데이트를 전송하므로 업데이트는 액세스 목록과 일치합니다. 액세스 목록은 이러한 업데이트를 거부합니다.

the .\*는 "."는 "임의의 문자"를 의미하고 \*는 "그 문자의 반복"을 의미하는 다른 정규식입니다. So, .\*는 다른 모든 업데이트의 전송을 허용하는 데 필요한 모든 경로 정보를 나타냅니다.

^200\$를 사용하지 않고 ^200을 사용하면 어떻게 됩니까? 이 절의 다이어그램에서와 같이 AS400을 사용하면 AS400이 생성하는 업데이트에 양식(200, 400)의 경로 정보가 포함됩니다. 이 경로 정보에서 200이 먼저이고 400이 끝입니다. 이러한 업데이트는 액세스 목록 ^200과 일치합니다. 경로 정보가 200으로 시작되기 때문입니다. 액세스 목록은 요구 사항이 아닌 RTA로의 이러한 업데이트 전송을 방지합니다.

올바른 정규식을 구현했는지 확인하려면 [show ip bgp regexpregular-expression 명령](#)을 실행합니다. 이 명령은 정규식 컨피그레이션과 일치하는 모든 경로를 표시합니다.

## AS 정규식

이 섹션에서는 정규식 생성에 대해 설명합니다.

정규식은 입력 문자열과 일치하는 패턴입니다. 정규식을 빌드할 때 입력이 일치해야 하는 문자열을 지정합니다. BGP의 경우 입력이 일치해야 하는 경로 정보로 구성된 문자열을 지정합니다.

Path Filter 섹션의 예에서는 문자열 ^200\$를 지정했습니다. 결정을 위해 업데이트 내부에 제공되는 경로 정보가 문자열과 일치하기를 원했습니다.

정규식은 다음과 같습니다.

- 

### 범위

범위는 왼쪽 및 오른쪽 대괄호 안의 문자 시퀀스입니다. 예를 들면 [abcd]입니다.

- 

### 아톰

원자는 단일 문자입니다. 몇 가지 예를 들면 다음과 같습니다.

- 

- 

the. 는 단일 문자와 일치합니다.

- 

- 

^은 입력 문자열의 시작과 일치합니다.

- 

- 

\$\$는 입력 문자열의 끝과 일치합니다.

\

이\는 문자와 일치합니다.

-

The\_는 쉼표(,), 왼쪽 중괄호({}), 오른쪽 중괄호()}, 입력 문자열의 시작, 입력 문자열의 끝 또는 공백과 일치합니다

•

**조각**

한 조각은 원자 뒤에 오는 다음 기호 중 하나입니다.

\*

\*는 원자의 0개 이상의 시퀀스와 일치합니다.

+

◦  
+는 원자의 1개 이상의 시퀀스와 일치합니다.

?

◦  
the?는 원자 또는 null 문자열과 일치합니다.

•  
지사

브랜치는 0개 이상의 연결된 조각입니다.

다음은 정규식의 몇 가지 예입니다.

a\*

•  
이 표현식은 문자 "a"가 발생했음을 나타내며, 여기에는 아무 것도 포함되지 않습니다.

a+

- 

이 식은 문자 "a"가 적어도 한 번 이상 있어야 함을 나타냅니다.

ab?a

- 

이 식은 "aa" 또는 "aba"와 일치합니다.

\_100\_

- 

이 표현식은 AS100을 통해 표현됩니다.

\_100\$

- 

이 표현식은 AS100의 출처를 나타냅니다.

^100 .\*

이 표현식은 AS100으로부터의 전송을 나타냅니다.

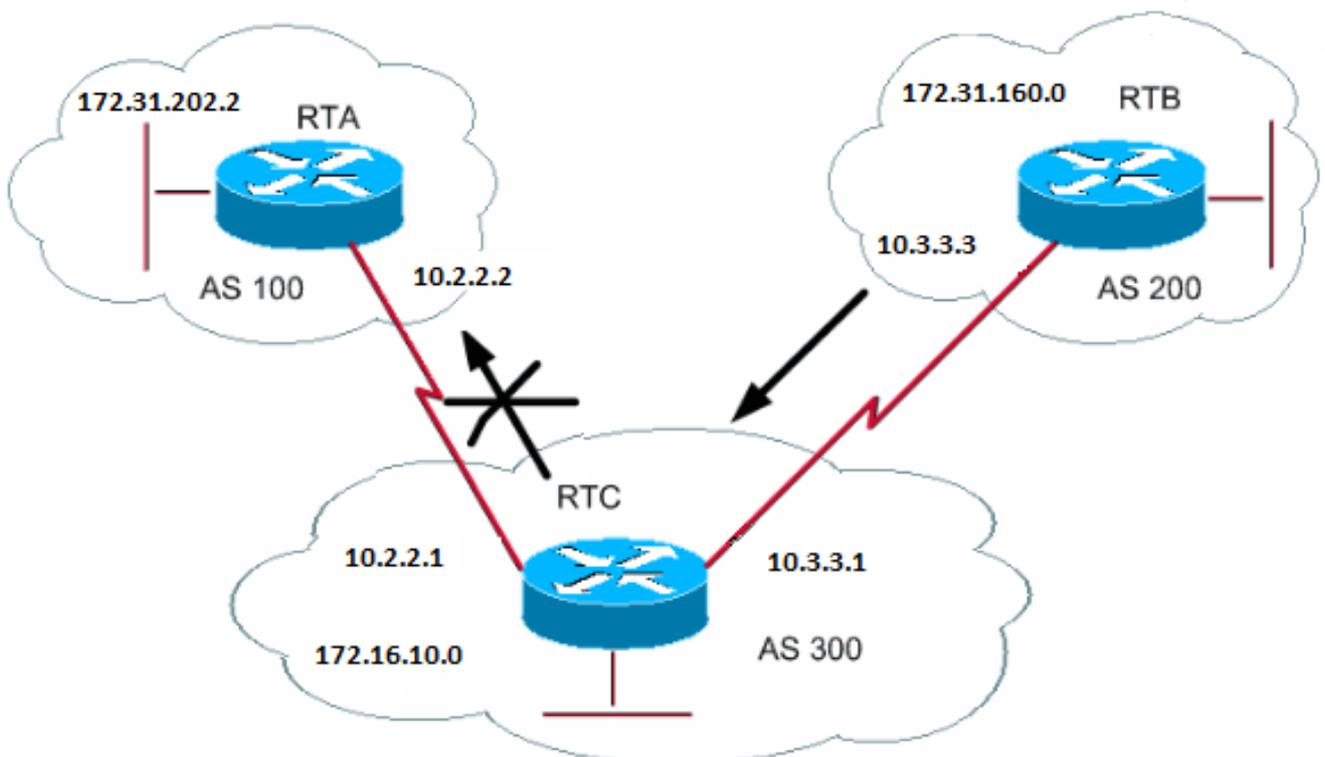
AS

이 표현식은 이 AS의 출처를 나타냅니다.

정규식 필터링의 샘플 [컨피그레이션은 BGP](#)에서 정규식 사용을 참조하십시오.

#### BGP 커뮤니티 필터

이 문서에서는 경로 필터링 및 AS 경로 필터링에 대해 다룹니다. 또 다른 방법은 커뮤니티 필터링입니다. Community Attribute 절에서는 커뮤니티에 대해 설명하고, 이 절에서는 커뮤니티 사용 방법에 대한 몇 가지 예를 제공합니다.

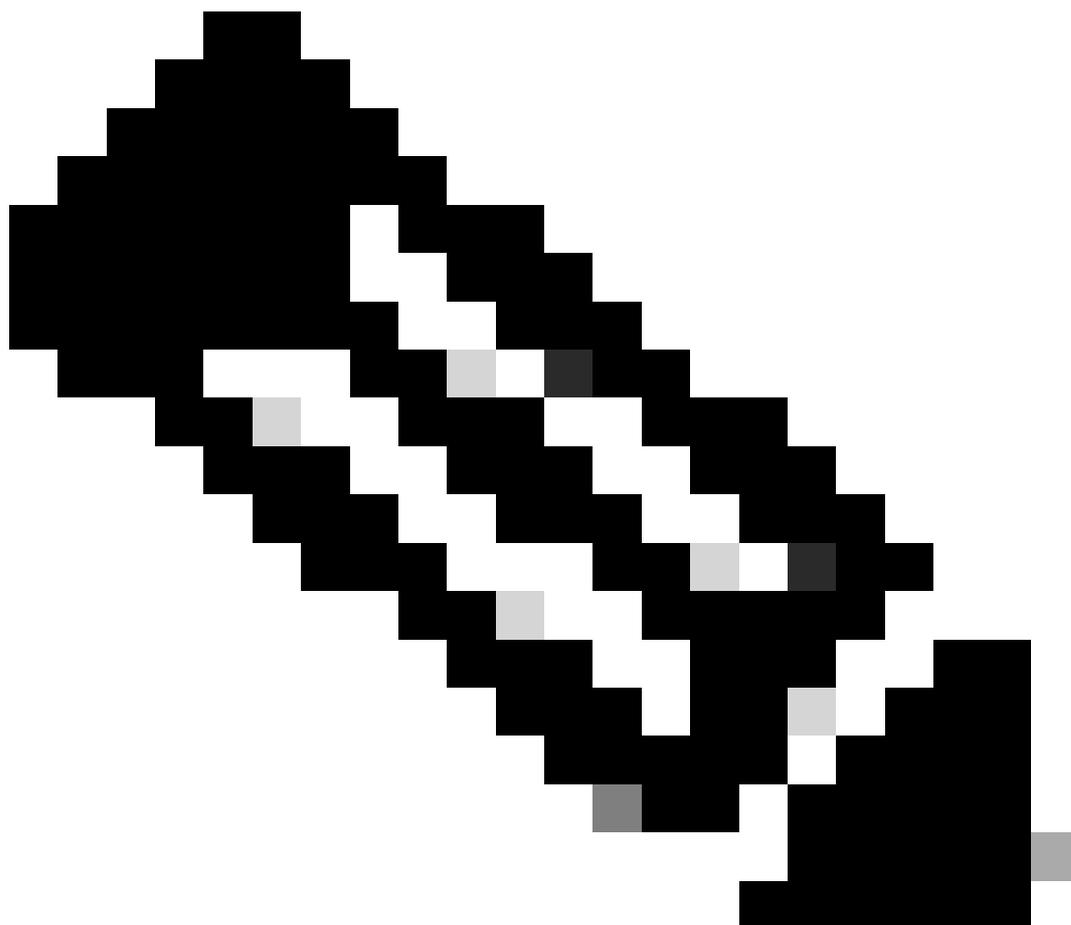


이 예에서는 RTC가 이러한 경로를 외부 피어에 전파하지 않도록 RTB가 RTB가 광고하는 BGP 경로에 대한 커뮤니티 특성을 설정하도록 합니다. community 특성을 no-export사용합니다.

```
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
```

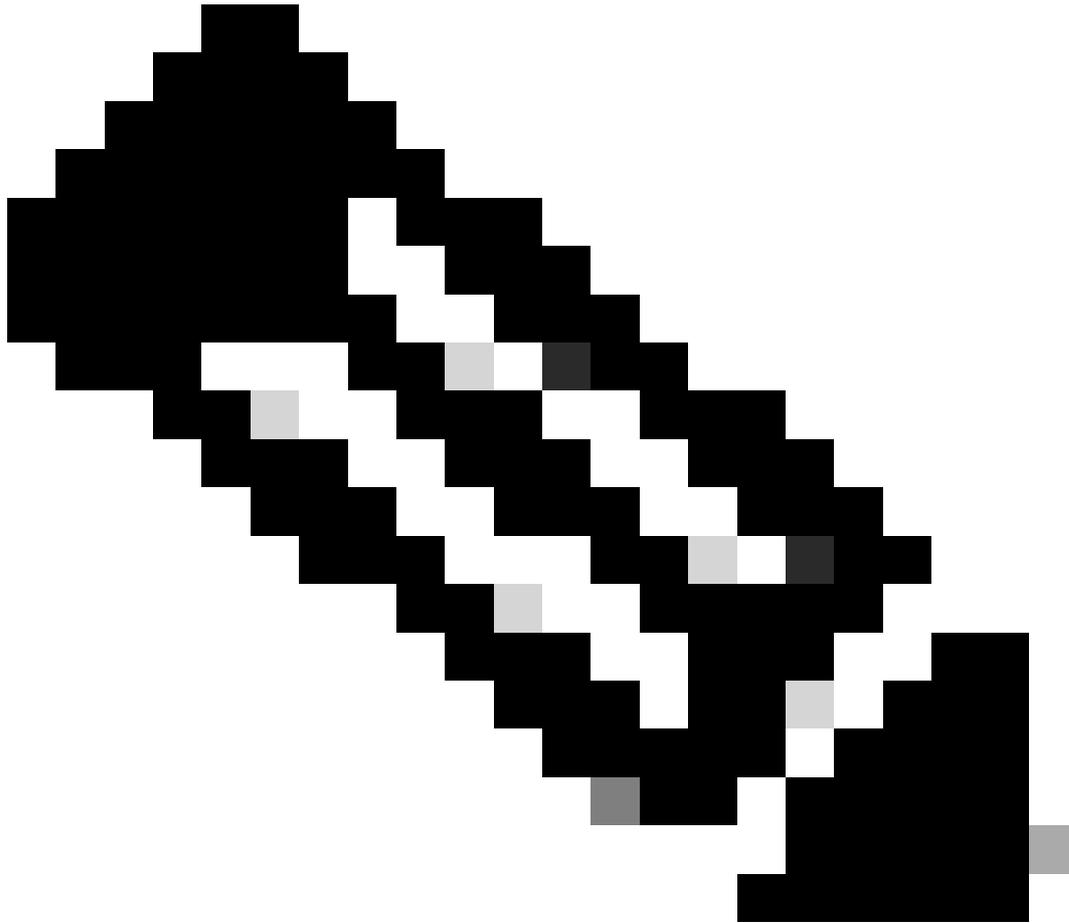


---

참고: 이 예에서는 `route-map setcommunity` 명령을 사용하여 커뮤니티를 `no-export`로 설정합니다.

---

---



참고: `neighbor send-community` 이 특성을 RTC에 보내려면 이 명령이 필요합니다.

---

RTC가 NO\_EXPORT 속성으로 업데이트를 가져오면 RTC는 외부 피어 RTA에 업데이트를 전파하지 않습니다.

이 예에서 RTB는 커뮤니티 특성을 로100 200 additive 설정했습니다. 이 작업은 RTC로 전송되기 전에 현재 커뮤니티 값에 값 100 200을 추가합니다.

```
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
```

커뮤니티 목록은 경로 맵의 일치 절에서 사용하는 커뮤니티 그룹입니다. 커뮤니티 목록을 사용하면 다른 커뮤니티 번호 목록을 기준으로 특성을 필터링하거나 설정할 수 있습니다.

<#root>

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

예를 들어, 이 경로 맵인 match-on-community를 정의할 수 있습니다.

```
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.

커뮤니티 목록을 사용하여 커뮤니티 값을 기준으로 특정 업데이트에서 가중치 및 메트릭과 같은 특정 매개변수를 필터링하거나 설정할 수 있습니다. 이 섹션의 두 번째 예에서 RTB는 커뮤니티 100 200으로 RTC에 업데이트를 보냈습니다. RTC가 이러한 값을 기준으로 가중치를 설정하려는 경우 다음을 수행할 수 있습니다.

```
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

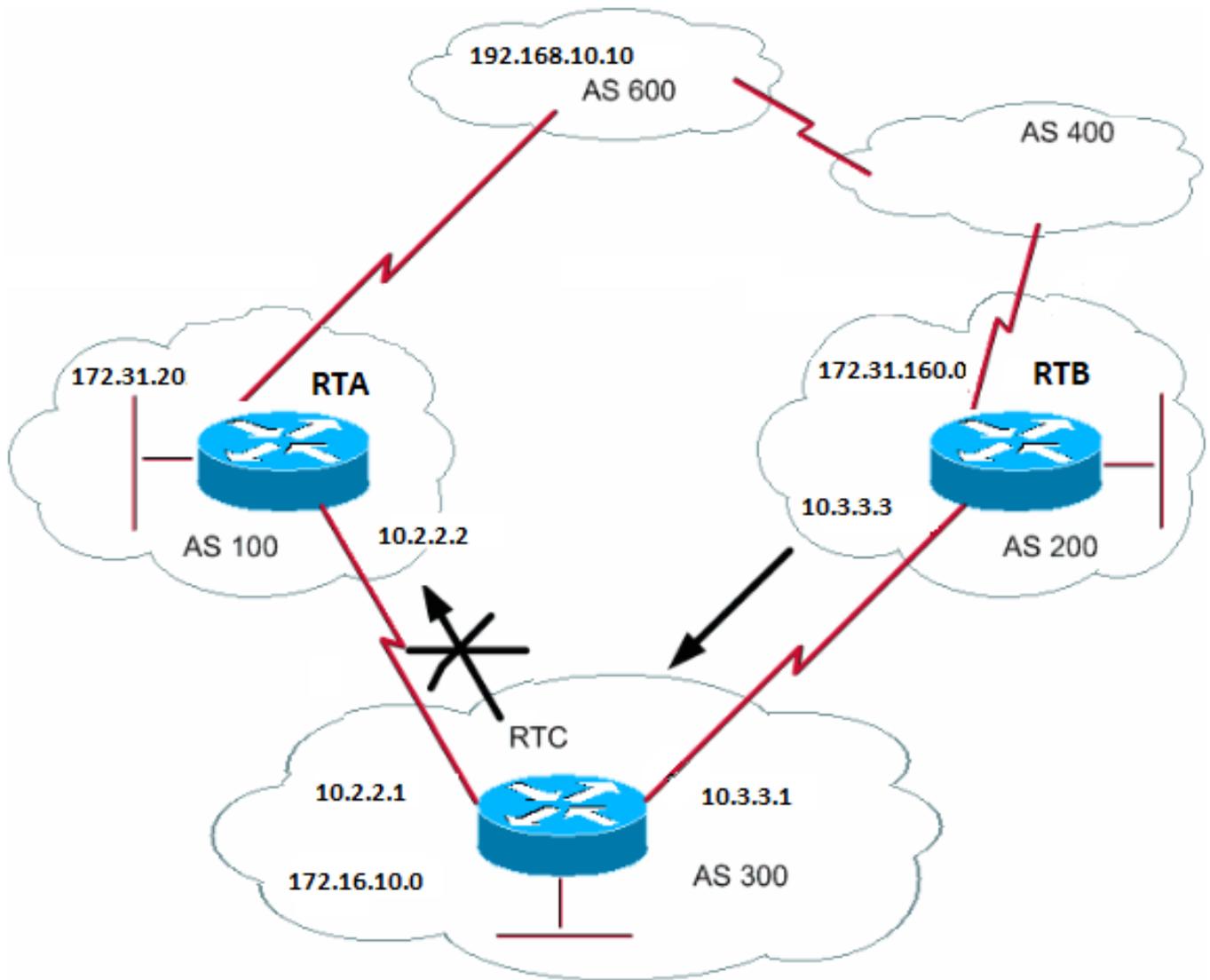
route-map check-community permit 30
 match community 3

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

이 예에서는 community 특성에 100이 있는 경로는 list 1과 일치합니다. 이 경로의 가중치는 20으로 설정됩니다. 커뮤니티 일치 항목 목록 2로 200만 있고 가중치가 20인 모든 경로. 키워드 **exact**는 커뮤니티가 200개로만 구성되고 다른 것은 없음을 나타냅니다. 마지막 커뮤니티 목록은 다른 업데이트가 삭제되지 않도록 하기 위한 것입니다. 기본적으로 일치하지 않는 모든 항목은 삭제된다는 것을 기억하십시오. internet 키워드는 모든 경로가 인터넷 커뮤니티의 구성원이므로 모든 경로를 나타냅니다.

자세한 내용은 [내용은 BGP 커뮤니티 값을 사용하여 업스트림 제공자 네트워크 구성 및 제어](#)를 참조하십시오.

BGP 네이버 및 경로 맵



경로 맵과 함께 **neighbor** 명령을 사용하여 수신 및 발신 업데이트에 대한 매개변수를 필터링하거나 설정할 수 있습니다.

IP 주소를 기준으로 일치하는 경우 neighbor 문과 연결된 경로 맵은 수신 업데이트에 영향을 미치지 않습니다.

<#root>

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

이 섹션의 다이어그램에서 RTC가 AS200에서 AS200에 로컬인 네트워크와 다른 네트워크에 대해 학습하도록 한다고 가정합니다. 또한 허용되는 경로의 가중치를 20으로 설정합니다. 네이버 및 **as-path** 액세스 목록을 조합하여 사용합니다.

```
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$
```

AS200에서 시작된 모든 업데이트에는 200으로 시작하고 200으로 끝나는 경로 정보가 있습니다. 이러한 업데이트는 허용됩니다. 다른 모든 업데이트가 삭제됩니다.

다음과 같이 가정합니다.

- AS200에서 시작되고 가중치가 20인 업데이트 수락
- AS400에서 시작된 업데이트 삭제
- 다른 업데이트의 경우 가중치 10

```
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 .*
```

이 문은 AS200에 로컬인 업데이트의 가중치를 20으로 설정합니다. 또한 이 명령문은 AS400 뒤에 있는 업데이트의 가중치를 10으로 설정하고 AS400에서 오는 업데이트를 삭제합니다.

#### Set As-path Prepend 명령 사용

BGP 결정 프로세스를 조작하려면 경로 정보를 조작해야 하는 경우도 있습니다. 경로 맵과 함께 사용하는 명령은 다음과 같습니다.

<#root>

[set as-path prepend](#) <as-path#> <as-path#>

BGP Neighbors and Route Maps 절의 다이어그램에서 RTC는 자체 네트워크 172.16.10.0을 두 개의 다른 AS, AS100 및 AS200에 광고한다고 가정합니다. 정보가 AS600으로 전파되면 AS600의 라우터에는 서로 다른 두 경로를 통해 약 172.16.10.0에 대한 네트워크 연결 정보가 있습니다. 첫 번째 경로는 경로(100, 300)가 있는 AS100을 거치고 두 번째 경로는 경로(400, 200, 300)가 있는 AS400을 거칩니다. 다른 모든 속성이 동일한 경우 AS600은 최단 경로를 선택하고 AS100을 통해 경로를 선택합니다.

AS300은 AS100을 통해 모든 트래픽을 가져옵니다. AS300 끝에서의 결정에 영향을 주려는 경우 AS100을 통과하는 경로가 AS400을 통과하는 경로보다 긴 것처럼 보일 수 있습니다. AS100으로 광고되는 현재 경로 정보에 AS 번호 앞에 추가하는 경우 이 작업을 수행할 수 있습니다. 일반적인 방법은 다음과 같이 자신의 AS 번호를 반복하는 것입니다.

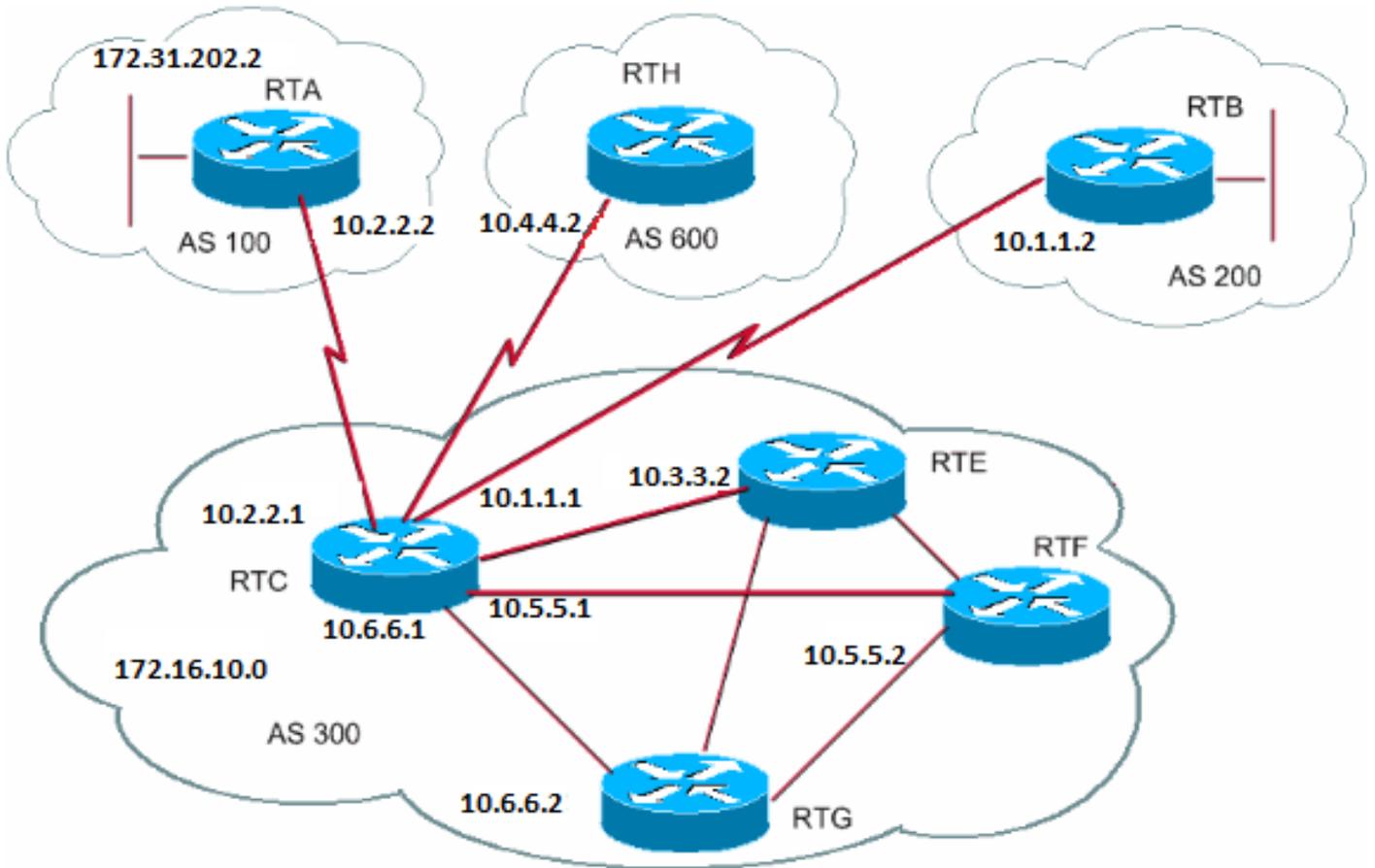
```
RTC#
router bgp 300
network 172.16.10.0
neighbor 10.2.2.2 remote-as 100
neighbor 10.2.2.2 route-map SETPATH out

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

이러한 구성으로 인해 AS600은 AS100을 통해 (100, 300, 300, 300)의 경로 정보로 172.16.10.0에 대한 업데이트를 수신합니다. 이 경

로 정보는 AS600이 AS400으로부터 받은 (400, 200, 300)보다 겁니다.

### BGP 피어 그룹



BGP 피어 그룹은 동일한 업데이트 정책을 사용하는 BGP 인접 디바이스 그룹입니다. 경로 맵, 배포 목록 및 필터 목록은 일반적으로 업데이트 정책을 설정합니다. 각 개별 네이버에 대해 동일한 정책을 정의하지 않습니다. 대신 피어 그룹 이름을 정의하고 이러한 정책을 피어 그룹에 할당합니다.

피어 그룹의 멤버는 피어 그룹의 모든 컨피그레이션 옵션을 상속합니다. 옵션이 아웃바운드 업데이트에 영향을 주지 않는 경우 이러한 옵션을 재정의하도록 멤버를 구성할 수도 있습니다. 인바운드에 설정된 옵션만 재정의할 수 있습니다.

피어 그룹을 정의하려면 다음 명령을 실행합니다.

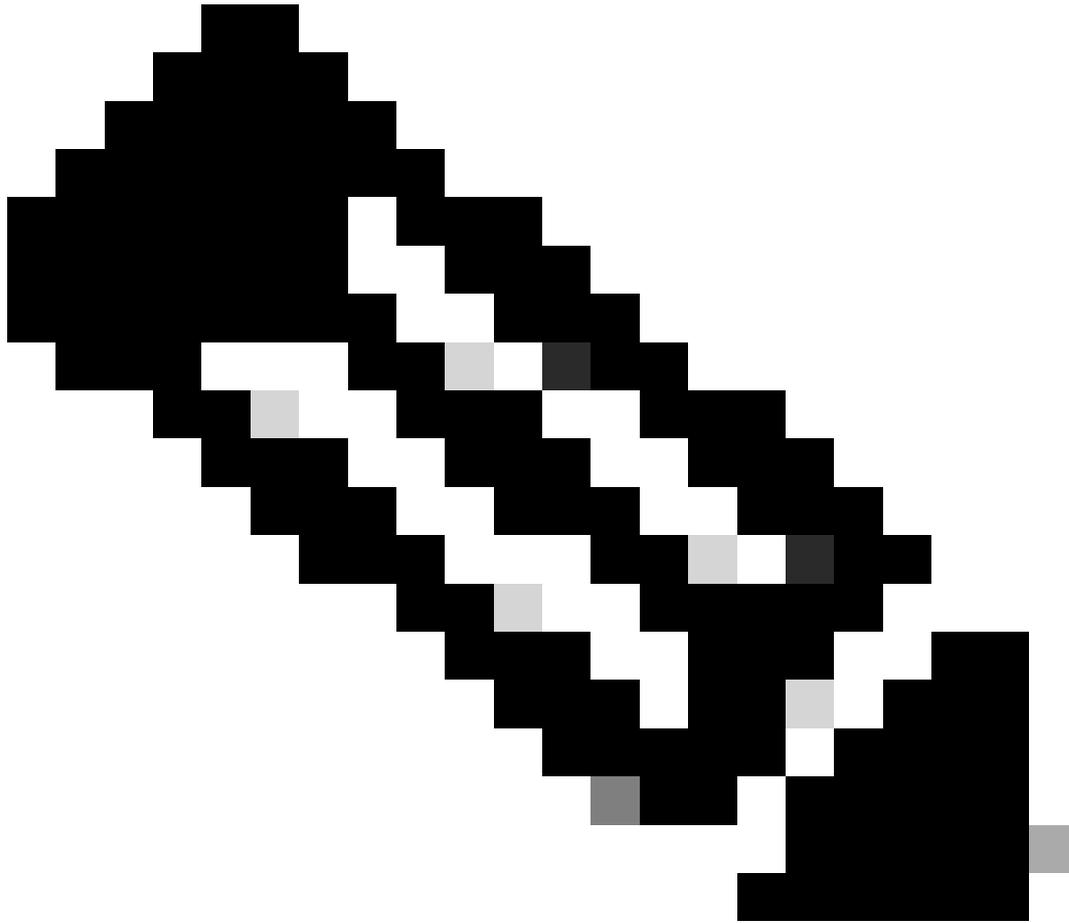
```
<#root>
```

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

이 예에서는 내부 및 외부 BGP 인접 디바이스에 피어 그룹을 적용합니다.

```
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
```

이 컨피그레이션은 이름이 internalmap인 피어 그룹을 정의합니다. 컨피그레이션에서는 메트릭을 5로 설정하고 2개의 서로 다른 필터 목록(1 및 2)을 설정하는 경로 맵 SETMETRIC과 같은 그룹에 대한 일부 정책을 정의합니다. 이 컨피그레이션은 피어 그룹을 모든 내부 네이버, RTE, RTF 및 RTG에 적용합니다. 또한 인접 RTE에 대해 별도의 필터 목록 3을 정의합니다. 이 필터 목록은 피어 그룹 내의 필터 목록 2를 재정의합니다.



참고: 인바운드 업데이트에 영향을 주는 옵션만 재정의할 수 있습니다.

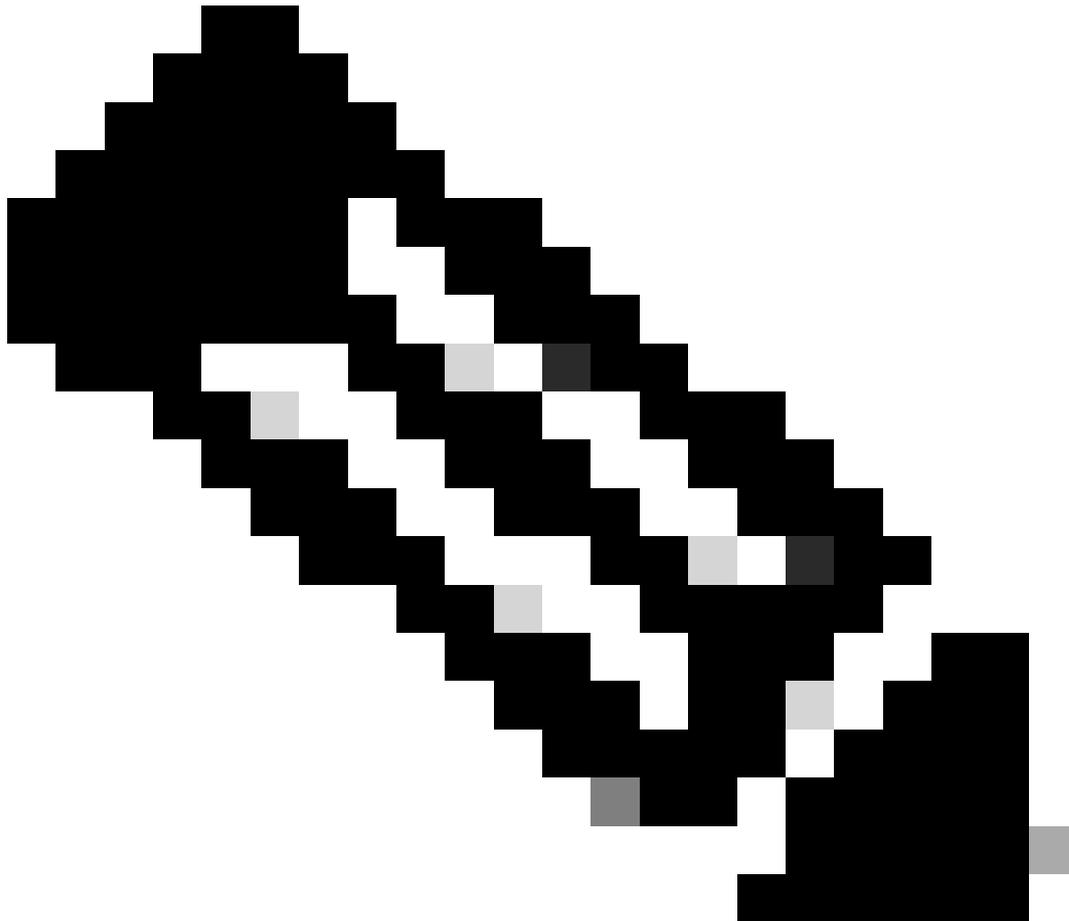
---

이제 외부 네이버와 함께 피어 그룹을 사용하는 방법을 살펴보십시오. 이 절의 다이어그램과 동일하게 피어 그룹 externalmap으로 RTC를 구성하고 피어 그룹을 외부 네이버에 적용합니다.

```
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
```

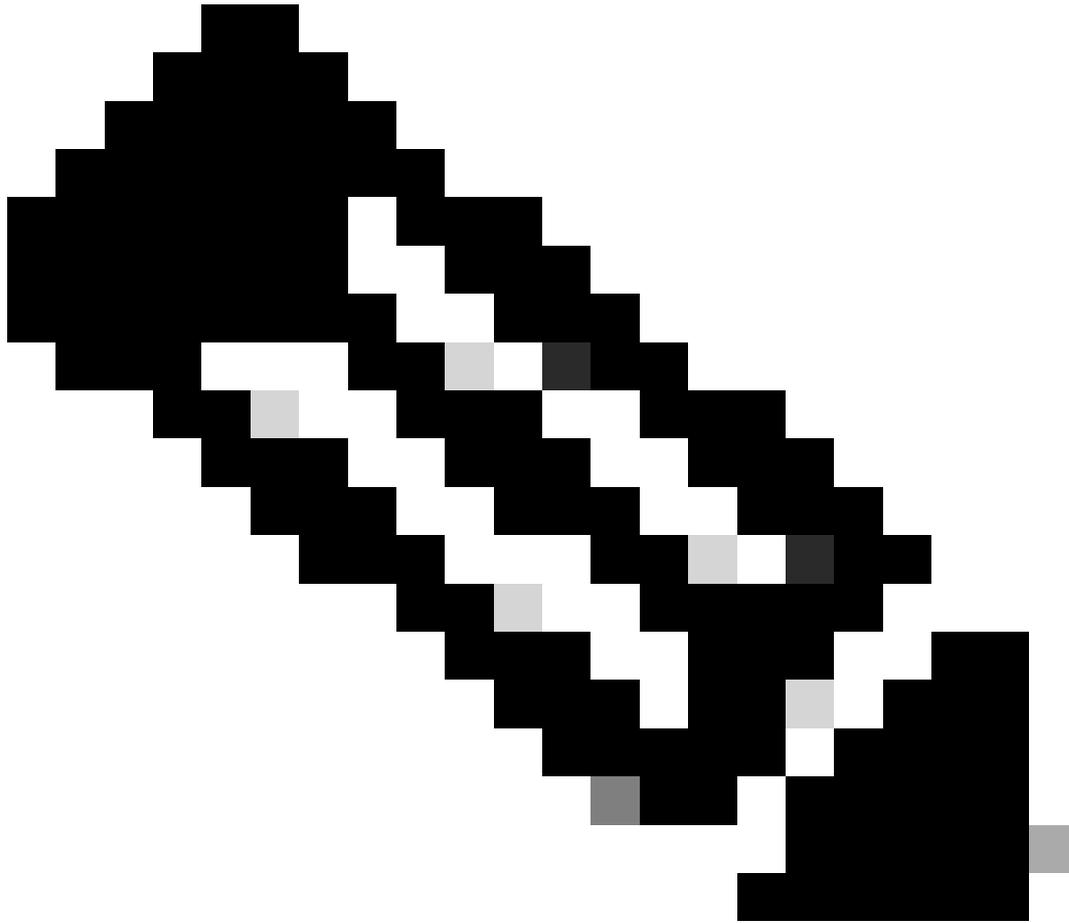
```
neighbor 10.2.2.2 peer-group externalmap
neighbor 10.4.4.2 remote-as 600
neighbor 10.4.4.2 peer-group externalmap
neighbor 10.1.1.2 remote-as 200
neighbor 10.1.1.2 peer-group externalmap
neighbor 10.1.1.2 filter-list 3 in
```

---

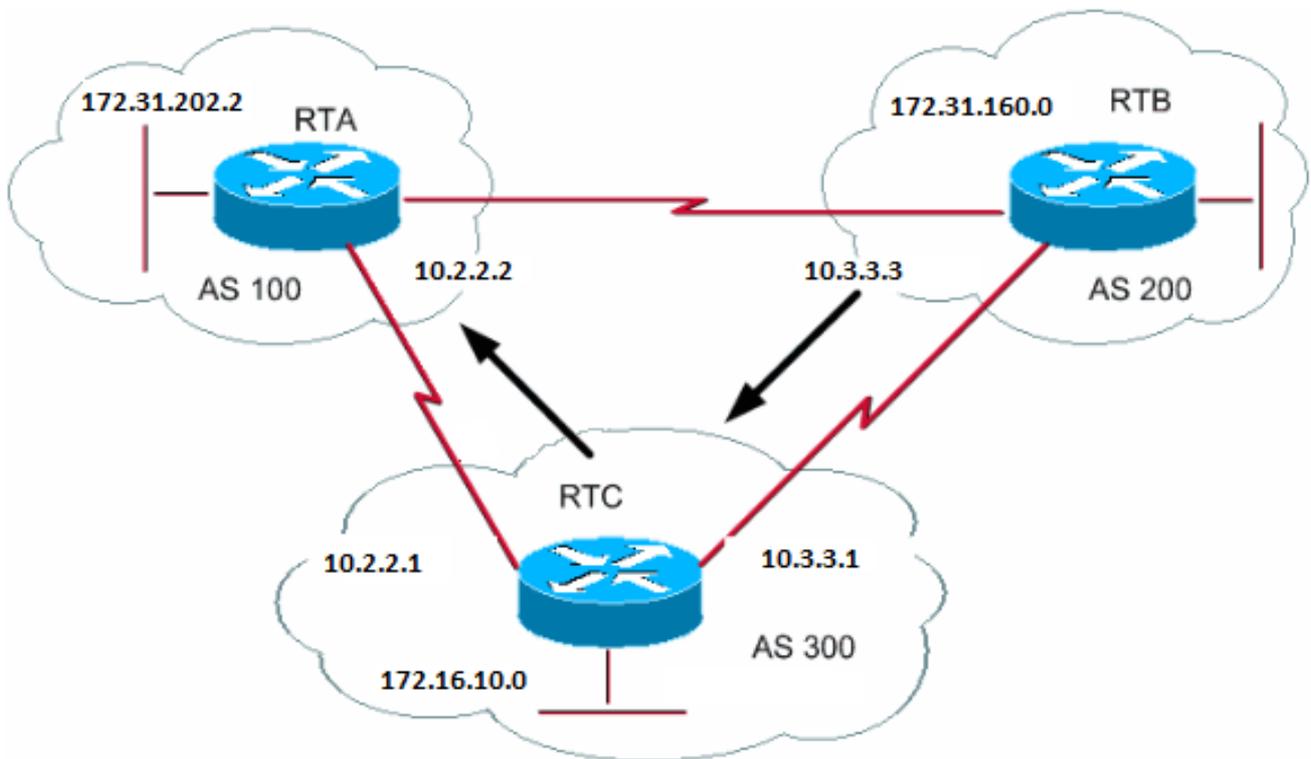


**참고:** 이러한 컨피그레이션에서는 다른 외부 AS를 정의해야 하므로 피어 그룹 외부에서 `remote-as` 문을 정의합니다. 또한 인접 디바이스 10.1.1.2의 인바운드 업데이트를 필터 목록 3의 할당으로 재정의합니다. 피어 그룹에 대한 자세한 내용은 BGP 피어 그룹을 참조하십시오.

---



**참고:** Cisco IOS Software Release 12.0(24)S에서 Cisco는 BGP Dynamic Update Peer Groups 기능을 도입했습니다. 이 기능은 이후 Cisco IOS Software 릴리스에서도 사용할 수 있습니다. 이 기능은 동일한 아웃바운드 정책을 공유하는 네이버의 업데이트 그룹을 동적으로 계산하고 최적화하는 새로운 알고리즘을 제공합니다. 이러한 인접 디바이스는 동일한 업데이트 메시지를 공유할 수 있습니다. Cisco IOS Software의 이전 릴리스에서는 피어 그룹 컨피그레이션을 기반으로 BGP 업데이트 메시지 그룹을 만들었습니다. 이 방법으로 그룹 업데이트가 제한된 아웃바운드 정책 및 특정 세션 컨피그레이션을 업데이트할 수 있습니다. BGP 동적 업데이트 피어 그룹 기능은 업데이트 그룹 복제를 피어 그룹 컨피그레이션과 분리합니다. 이러한 분리는 컨버전스 시간 및 인접 디바이스 컨피그레이션의 유연성을 향상합니다. 자세한 내용은 BGP 동적 업데이트 피어 그룹을 참조하십시오.



BGP4 over BGP3의 주요 개선 사항 중 하나는 CIDR(Classless Interdomain Routing)입니다. CIDR 또는 수퍼네팅은 IP 주소를 새롭게 확인하는 방법입니다. CIDR에서는 클래스 A, B 또는 C와 같은 클래스 개념이 없습니다. 예를 들어 네트워크 192.168.213.0은 한때 불법 클래스 C 네트워크였습니다. 이제 네트워크는 192.168.213.0/16의 법적 슈퍼넷입니다. 16은 IP 주소의 맨 왼쪽에서 셀 때 서브넷 마스크의 비트 수를 나타냅니다. 이 표현은 192.168.213.0 255.255.0.0과 유사합니다.

라우팅 테이블의 크기를 최소화하기 위해 집계를 사용합니다. 어그리게이션은 하나의 경로에 대한 광고가 가능하도록 여러 다른 경로의 특성을 결합하는 과정입니다. 이 예에서 RTB는 네트워크 172.31.160.0을 생성합니다. 해당 경로 192.168.160.0의 슈퍼넷을 RTA로 전파하도록 RTC를 구성합니다.

```

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

```

RTC는 종합 주소 192.168.160.0을 RTA에 전파합니다.

집계 명령

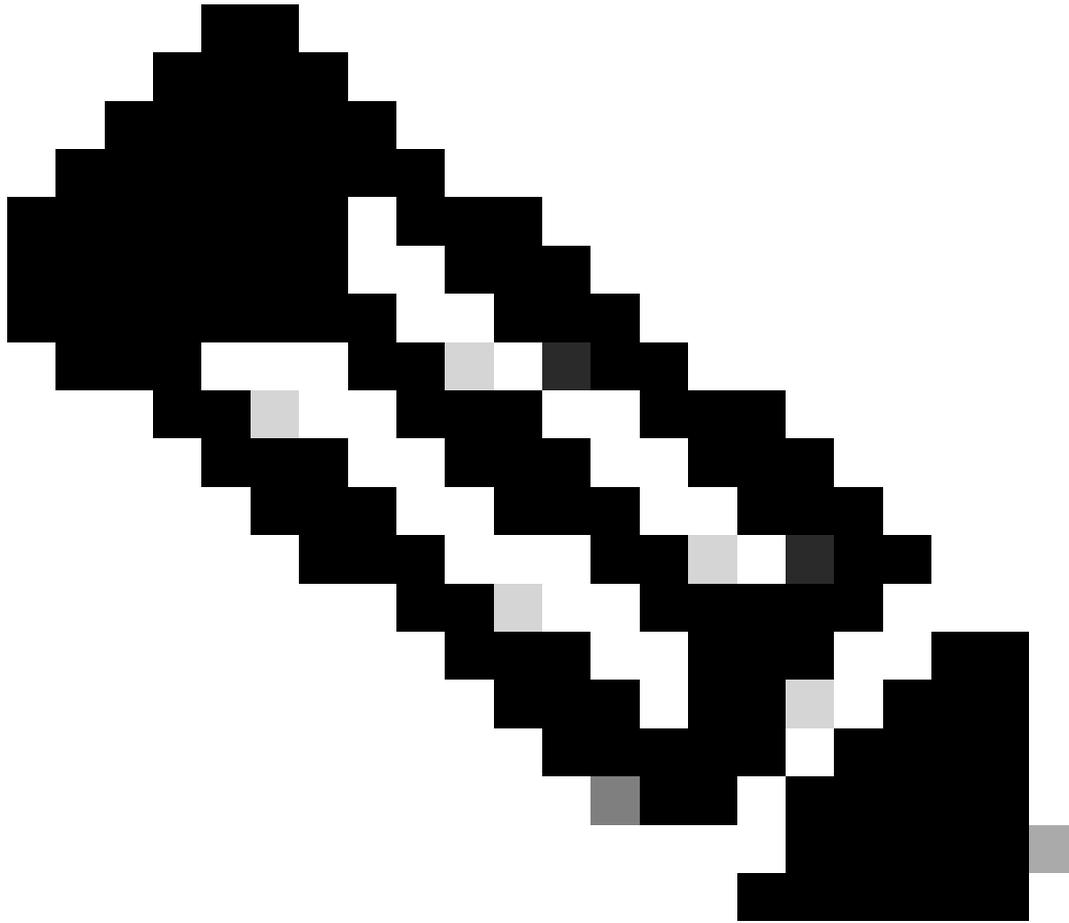
광범위한 aggregate 명령이 있습니다. 각자 자신이 원하는 집성 행동을 하기 위해서는 어떻게 작용하는지 이해해야 한다.

첫 번째 명령은 CIDR 및 종합 주소 섹션의 예에 있는 명령입니다.

```
<#root>
```

```
aggregate-address address-mask
```

이 명령은 접두사 경로 및 모든 더 구체적인 경로를 광고합니다. **aggregate-address 192.168.160.0** 명령은 추가 네트워크 192.168.160.0을 전파하지만 172.31.160.0이 RTA로 전파되는 것을 막지는 않습니다. 그 결과 두 네트워크 192.168.160.0 및 172.31.160.0이 RTA로 전파되며, 이는 접두사와 더 구체적인 경로를 모두 광고하는 것입니다.



**참고:** BGP 라우팅 테이블에 해당 주소의 더 구체적인 경로가 없는 경우 주소를 집계할 수 없습니다.

---

예를 들어, RTB에 BGP 테이블에 더 구체적인 192.168.160.0 항목이 없으면 RTB는 192.168.160.0에 대한 집계를 생성할 수 없습니다. BGP 테이블에 더 구체적인 경로를 삽입할 수 있습니다. 경로 삽입은 다음을 통해 발생할 수 있습니다.

- 다른 AS에서 들어오는 업데이트

-

IGP 또는 고정에서 BGP로 재배포

- 

network 명령(예: **network 172.31.160.0**)

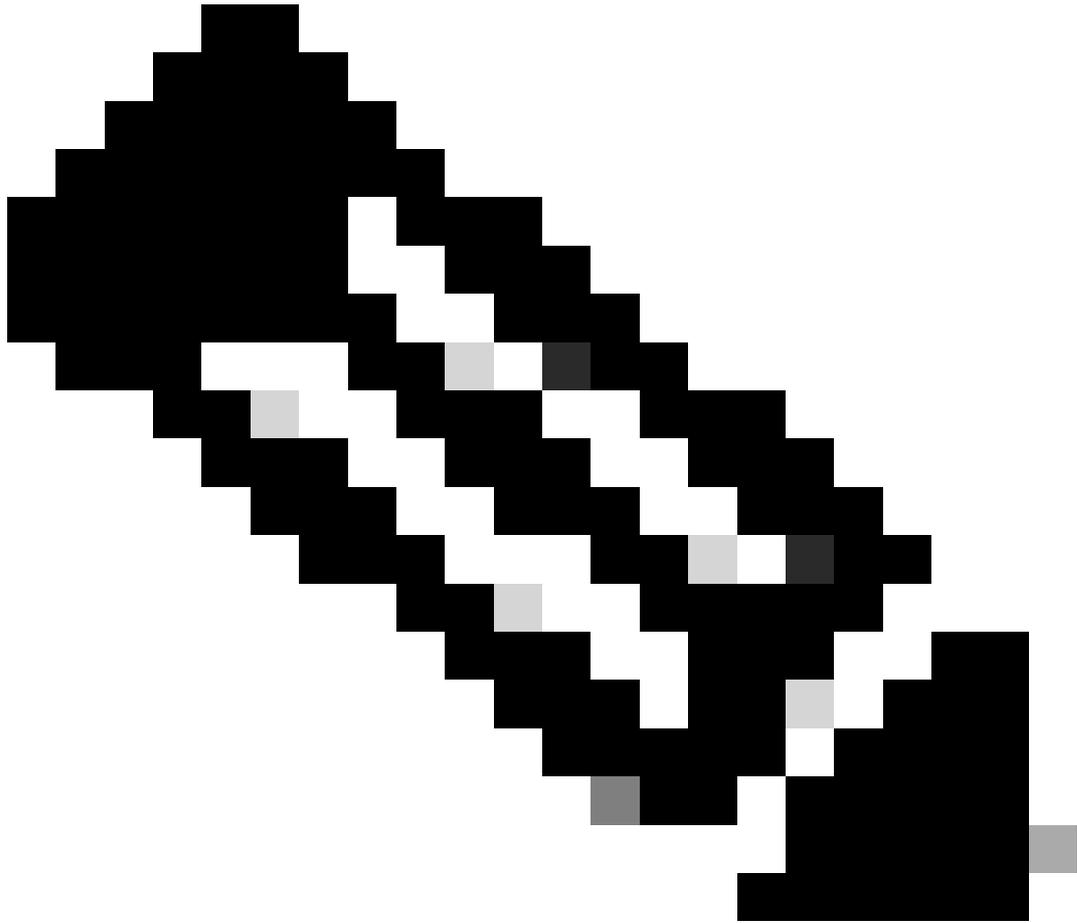
RTC가 네트워크 192.168.160.0만 전파하고 더 구체적인 경로만 전파하도록 하려면 다음 명령을 실행합니다.

```
<#root>
```

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

이 명령은 접두사만 광고합니다. 이 명령은 더 구체적인 경로를 모두 억제합니다.

명령 집계 **192.168.160.0 255.0.0.0**은 네트워크 192.168.160.0을 요약 전용으로 전파하고 더 구체적인 경로 172.31.160.0을 억제합니다.



**참고:** network 문을 통해 BGP에 삽입된 네트워크를 종합하면 네트워크 항목은 항상 BGP 업데이트에 삽입됩니다. 이 주입은 aggregate summary-only 명령을 사용하는 경우에도 발생합니다. CIDR 단원의 예시 예 1에서는 이 상황을 설명합니다.

---

<#root>

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

이 명령은 접두사와 더 구체적인 경로를 광고합니다. 그러나 이 명령은 라우팅 업데이트의 경로 정보에 as-set 정보를 포함합니다.

```
<#root>
```

```
aggregate 192.168.0.0 255.0.0.0 as-set
```

sectionCIDR 예 2(as-set)에서는 이 명령에 대해 설명합니다.

어그리게이션을 수행할 때 더 구체적인 경로를 억제하려면 경로 맵을 정의하고 어그리게이션에 경로 맵을 적용합니다. 이 작업을 통해 억제할 더 구체적인 경로를 선택할 수 있습니다.

```
<#root>
```

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

이 명령은 접두사와 더 구체적인 경로를 광고합니다. 그러나 이 명령은 경로 맵 기반으로 광고를 억제합니다. CIDR 및 Aggregate Addresses 섹션의 다이어그램에서 192.168.160.0을 집계하고, 더 구체적인 경로 192.168.160.20을 억제하고, 172.31.160.0의 전파를 허용한다고 가정합니다. 이 경로 맵 사용:

```
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
```

suppress-map의 정의에 따라 액세스 목록이 허용하는 모든 패킷의 업데이트에서 제외됩니다.

그런 다음 경로 맵을 aggregate 문에 적용합니다.

```
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
```

다음은 또 다른 변형입니다.

```
<#root>
```

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

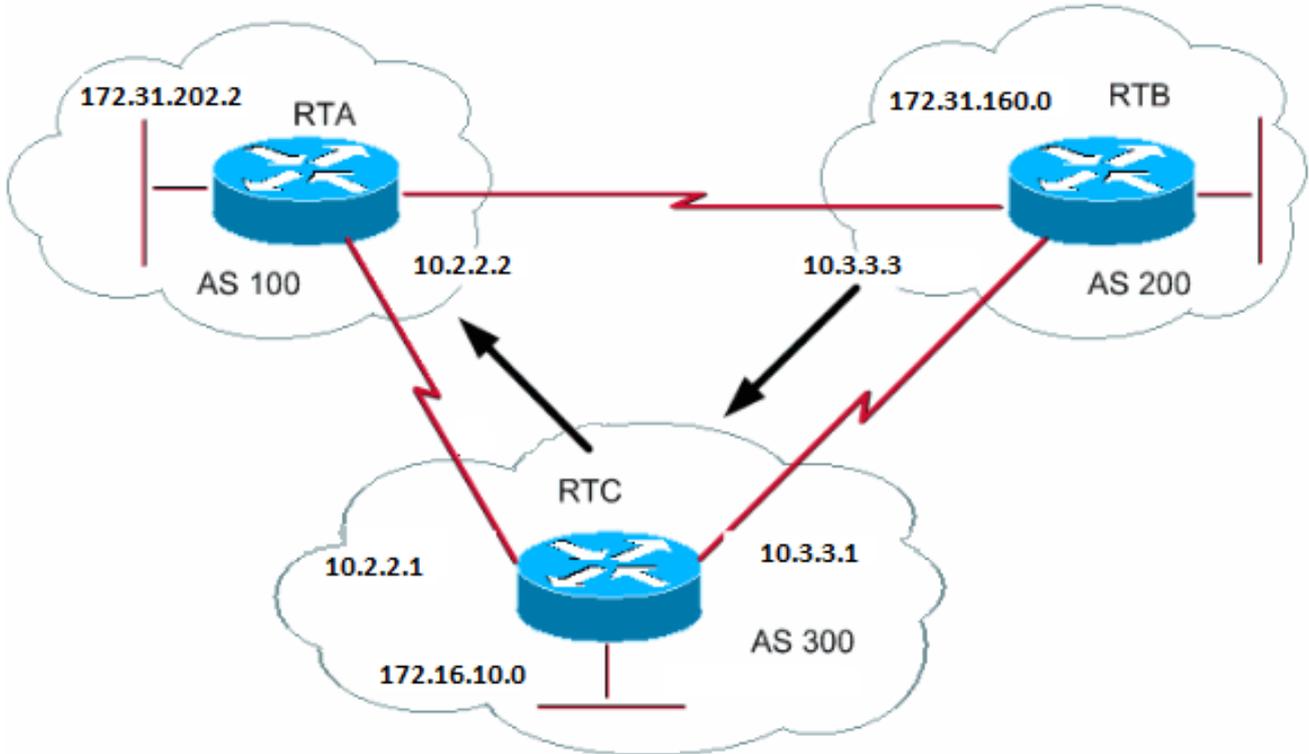
이 명령을 사용하면 집계를 전송할 때 메트릭 등의 특성을 설정할 수 있습니다. 집계의 원점을 IGP로 설정하려면 이 경로 맵을 aggregate attribute-map 명령에 적용합니다.

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

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

자세한 내용은 [BGP에서 경로 어그리게이션 이해를 참조하십시오.](#)

CIDR 예 1



요청: RTB에서 접두사 192.168.160.0을 알리고 더 구체적인 모든 경로를 억제하도록 허용합니다. 이 요청의 문제는 네트워크 172.31.160.0이 AS200에 로컬이라는 것입니다. 즉, AS200이 172.31.160.0의 발신자입니다. aggregate summary-only 명령을 사용하더라도 172.31.160.0에 대한 항목을 생성하지 않고 RTB가 192.168.160.0에 대한 접두사를 생성하도록 할 수 없습니다. RTB는 172.31.160.0의 발신자이므로 두 네트워크를 모두 생성합니다. 이 문제에 대한 두 가지 해결책이 있습니다.

첫 번째 솔루션은 고정 경로를 사용하고 BGP로 재배포하는 것입니다. 결과는 RTB가 불안정한 기원(?)이 있는 집계를 광고한다는 것입니다.

```
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
```

두 번째 솔루션에서는 고정 경로 외에 **network** 명령에 대한 엔트리를 추가합니다. 이 엔트리는 업데이트의 출처를 IGP로 설정하는 것을 제외하고 동일한 영향을 미칩니다.

```

RTB#
router bgp 200
network 192.168.160.0 mask 255.0.0.0

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

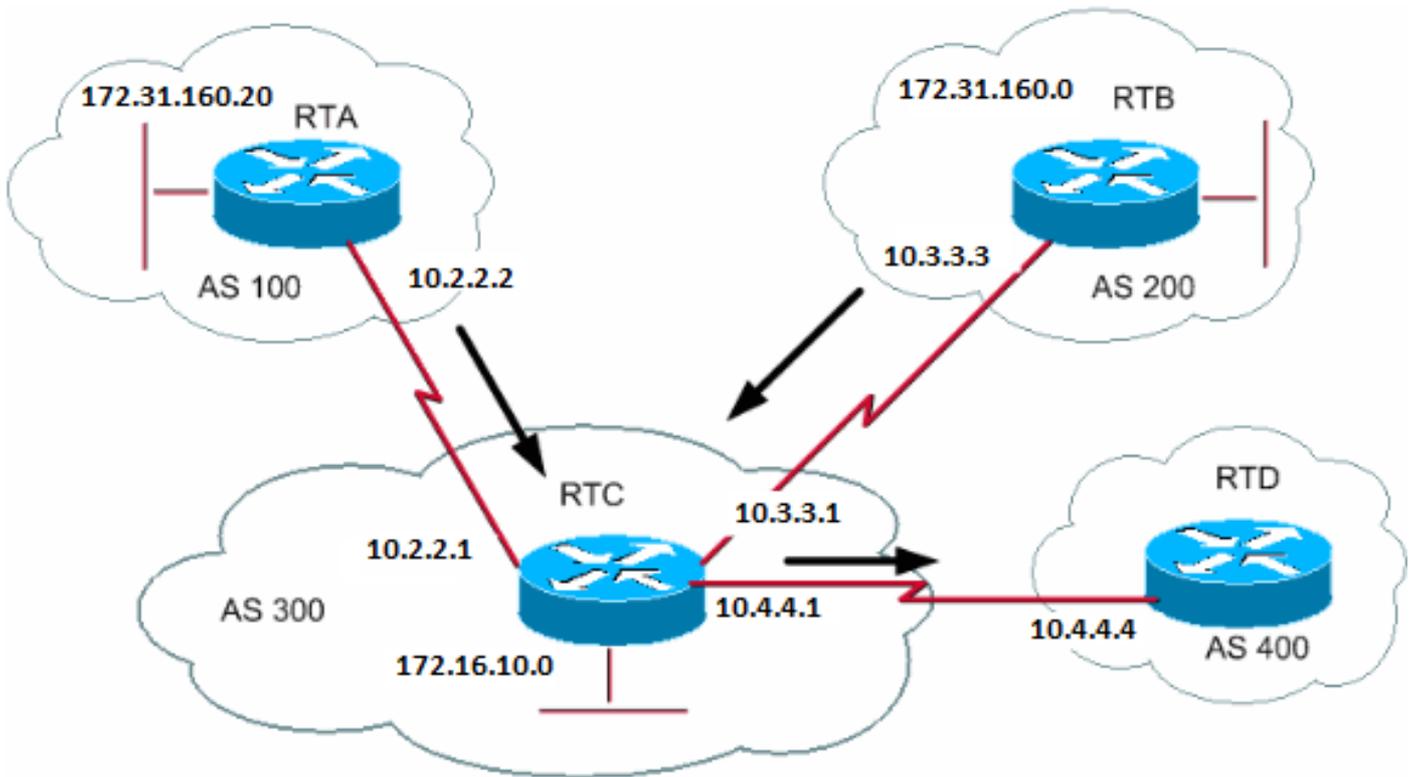
neighbor 10.3.3.1 remote-as 300
redistribute static

ip route 192.168.160.0 255.0.0.0 null0

```

CIDR 예 2(as-set)

경로 정보의 크기를 줄이려면 어그리게이션에서 명령문을 set로 사용합니다. as가 설정된 경우, 집계된 여러 경로에 AS 번호가 나타난 횟수에 관계없이 AS 번호는 한 번만 나열됩니다. 정보를 종합하면 path 특성과 관련된 정보가 손실되는 상황에서 aggregate as-set 명령을 사용합니다. 이 예에서 RTC는 RTA에서 약 192.168.160.20의 업데이트를 받고 RTB에서 약 172.31.160.0의 업데이트를 받습니다. RTC가 네트워크 192.168.160.0/8을 집계하여 RTD로 전송하려고 한다고 가정합니다. RTD는 해당 경로의 출처를 알지 못합니다. aggregate as-set 문을 추가하면 RTC가 set{}의 형식으로 경로 정보를 생성하도록 강제합니다. 그 집합은 어떤 경로가 먼저였는지와 상관없이 모든 경로 정보를 포함한다.



```

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
```

사례 1:

RTC에 as-set **문**이 없습니다. RTC는 경로가 AS300에서 시작된 것처럼 경로 정보(300)와 함께 업데이트 192.168.160.0/8을 RTD에 보냅니다.

```
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.
```

사례 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}.
```

다음 두 주제인 BGP Federation 및 Route Reflector는 AS 내에서 iBGP 피어링 폭증을 추가로 제어하려는 ISP(Internet Service Provider)를 위한 것입니다.

## BGP 연합

BGP 연합을 구현하면 AS 내부의 iBGP 메시지가 줄어듭니다. 하나의 AS를 여러 AS로 나누고 그룹 전체를 하나의 연합에 할당하는 게 요령이다. 각 AS만 iBGP를 완전히 매쉬하며 연합 내부의 다른 AS에 연결됩니다. 이러한 AS에는 연합 내의 AS에 대한 eBGP 피어가 있지만 AS는 iBGP를 사용한 것처럼 라우팅을 교환합니다. 이러한 방식으로, 연합은 다음 홉, 메트릭, 그리고 로컬 선호도 정보를 보존한다. 겉으로는 민노총은 단일 AS로 보인다.

BGP 연합을 구성하려면 다음 명령을 실행합니다.

```
<#root>
```

```
bgp confederation identifier <autonomous-system>
```

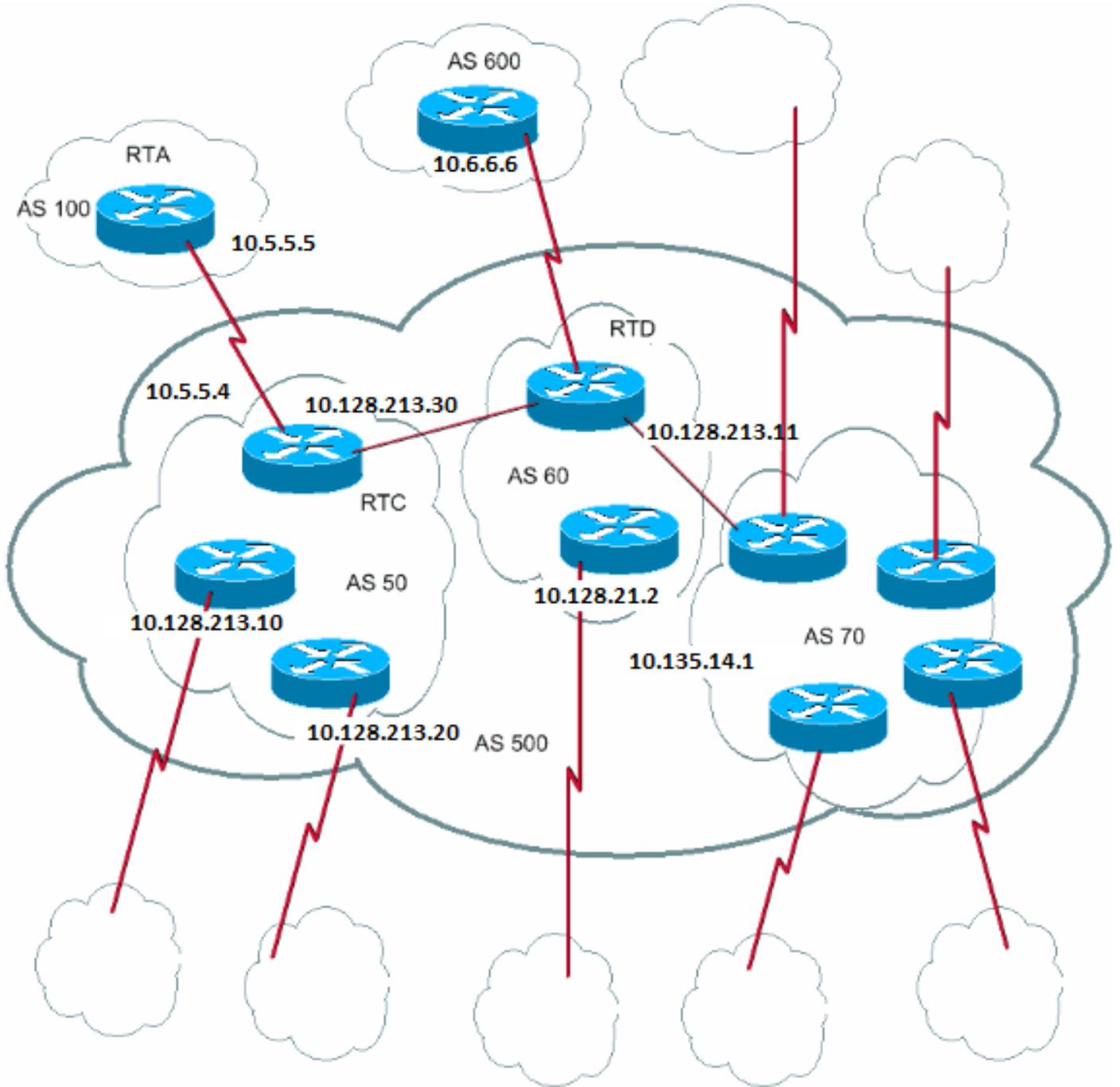
페더레이션 식별자는 페더레이션 그룹의 AS 번호입니다.

이 명령의 문제는 연합 내의 여러 AS 간에 피어링을 수행합니다.

```
<#root>
```

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

다음은 연합의 예입니다.



BGP 스피커 9개로 구성된 AS500이 있다고 가정합니다. 다른 비 BGP 스피커도 있지만 다른 AS에 대한 eBGP 연결이 있는 BGP 스피커만 관심을 갖습니다. AS500 내에서 전체 iBGP 메쉬를 만들려면 각 라우터에 대해 9개의 피어 연결이 필요합니다. iBGP 피어 8개와 eBGP 피어 1개를 외부 AS에 연결해야 합니다.

연합을 사용하는 경우 AS500을 여러 AS(AS50, AS60 및 AS70)로 나눌 수 있습니다. AS에 연합 식별자 500을 지정합니다. 외부에서는 AS, AS500 하나만 봅니다. 각 AS50, AS60 및 AS70에 대해 iBGP 피어의 전체 메시지를 정의하고 `bgp federation peers` 명령으로 연합 피어의 목록을 정의합니다.

다음은 라우터 RTC, RTD, RTA의 샘플 컨피그레이션입니다.

---

참고: RTA에는 AS50, AS60 또는 AS70에 대한 지식이 없습니다. RTA는 AS500에 대해서만 알고 있습니다.

---

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

```

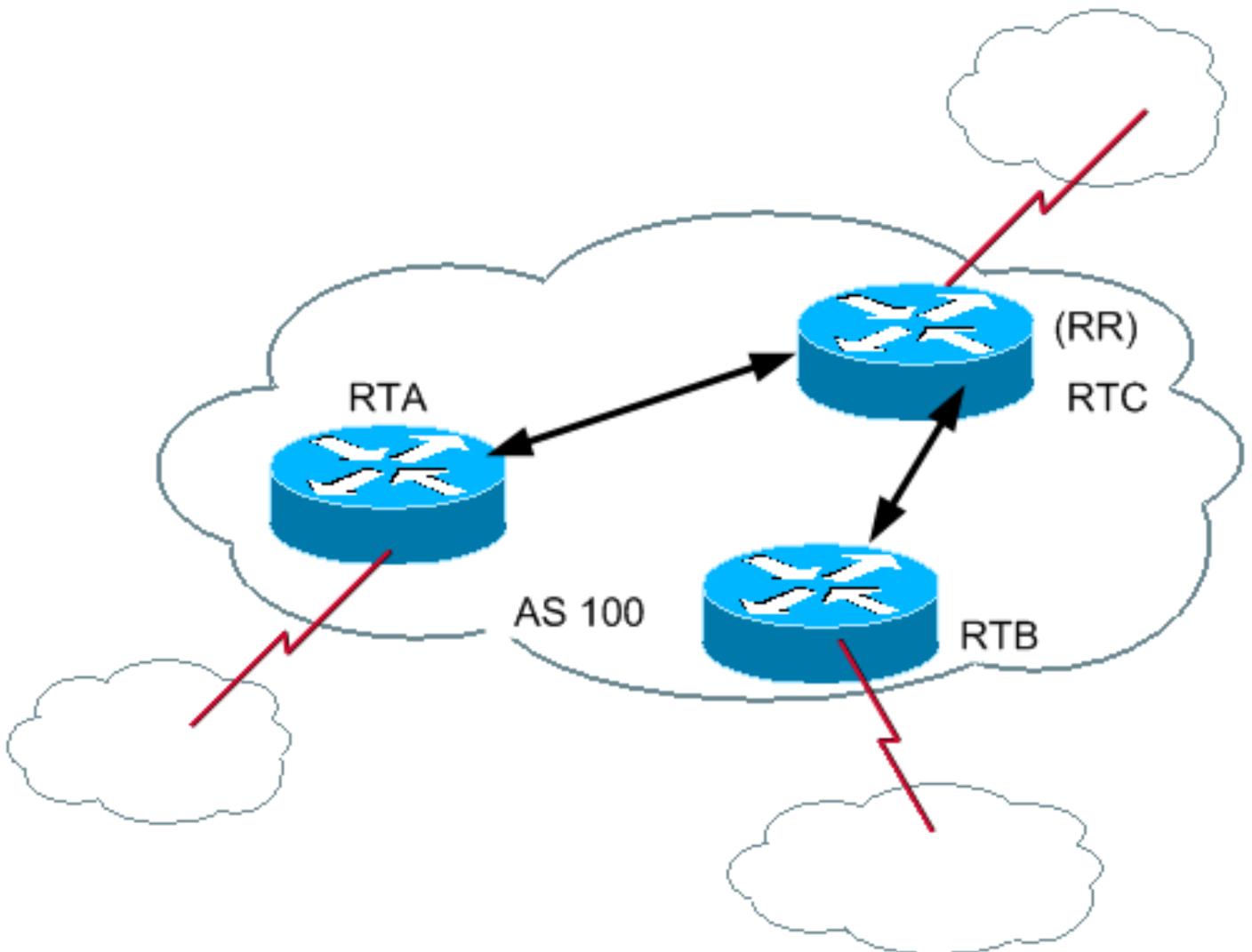
bgp confederation identifier 500
bgp confederation peers 50 70
neighbor 10.128.210.2 remote-as 60 (IBGP connection within AS60)
neighbor 10.128.213.30 remote-as 50(BGP connection with confederation peer 50)
neighbor 10.128.213.14 remote-as 70 (BGP connection with confederation peer 70)
neighbor 10.6.6.16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

```

### 경로 리플렉터

AS 내에서 iBGP 피어링 급증에 대한 또 다른 해결책은 RR(Route Reflector)입니다. iBGP 섹션에서 보여주는 것처럼, BGP 스피커는 BGP 스피커가 다른 iBGP 스피커를 통해 학습한 경로를 세 번째 iBGP 스피커에 알리지 않습니다. 이 제한을 약간 완화하고 라우터가 iBGP에서 학습한 경로를 다른 iBGP 스피커에 알리거나 반영할 수 있는 추가 제어를 제공할 수 있습니다. 이러한 경로 반영은 AS 내의 iBGP 피어 수를 줄입니다.



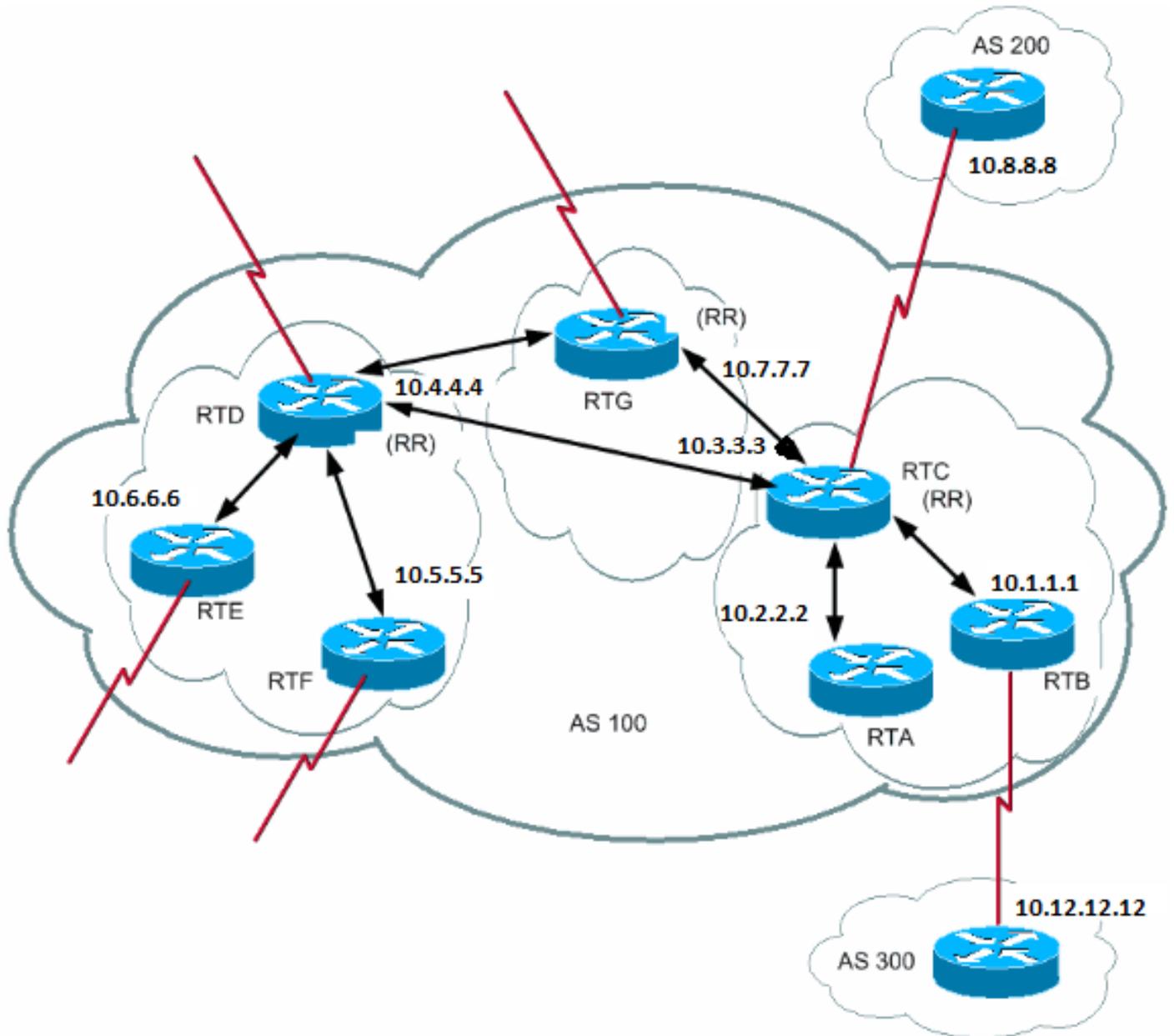
일반적인 경우 AS100 내에서 RTA, RTB, RTC 간 전체 iBGP 메쉬를 유지합니다. RR 개념을 활용하면 RTC를 RR로 선택할 수 있습니다. 이러한 방식으로 RTC는 RTA 및 RTB와의 부분 iBGP 피어링을 갖습니다. RTC는 RTA 및 RTB에서 제공되는 업데이트에 대한 RR이므로 RTA와 RTB 간의 피어링이 필요하지 않습니다.

<#root>

[neighbor <ip address> route-reflector-client](#)

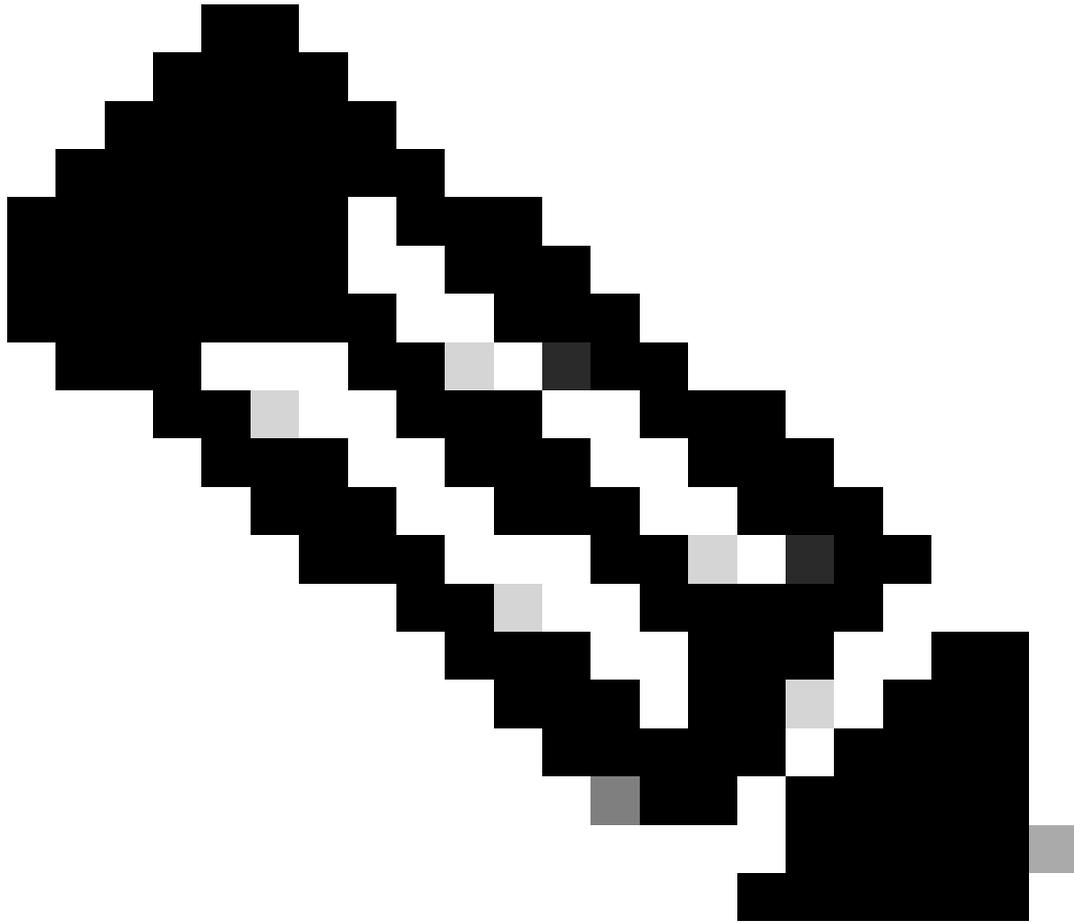
이 명령을 사용하는 라우터는 RR이며, 명령점이 있는 인접 디바이스가 해당 RR의 클라이언트입니다. 이 예에서 RTC 컨피그레이션에는 RTA 및 **RTB** IP 주소를 가리키는 neighbor route-reflector-client 명령이 있습니다. RR과 클라이언트의 조합이 "클러스터"입니다. 이 예에서 RTA, RTB, RTC는 AS100 내에서 단일 RR로 클러스터를 구성합니다.

클라이언트가 아닌 RR의 다른 iBGP 피어는 비클라이언트입니다.



AS는 둘 이상의 RR을 가질 수 있습니다. 이 경우 RR은 다른 iBGP 스피커와 마찬가지로 다른 RR을 처리합니다. 다른 RR은 동일한 클러스터(클라이언트 그룹) 또는 다른 클러스터에 속할 수 있습니다. 간단한 컨피그레이션에서는 AS를 여러 클러스터로 나눌 수 있습니다. 풀 메시 토폴로지에서는 다른 RR로 각 RR을 비클라이언트 피어로 구성합니다. 클라이언트는 클라이언트 클러스터 외부의 iBGP 스피커와 피어링해서는 안 됩니다.

앞의 다이어그램에서 RTA, RTB, RTC는 하나의 클러스터를 구성한다. RTC는 RR입니다. RTC의 경우 RTA와 RTB는 클라이언트이고 다른 것은 클라이언트가 아닙니다. `neighbor route-reflector-client` 명령은 RR의 클라이언트를 가리킵니다. 동일한 RTD는 클라이언트 RTE 및 RTF에 대한 RR입니다. RTG는 세 번째 클러스터의 RR입니다.



참고: RTD, RTC 및 RTG는 완전히 메싱되지만 클러스터 내의 라우터는 그렇지 않습니다.

---

RR이 경로를 수신하면 이 목록에 표시된 대로 RR이 경로를 라우팅합니다. 그러나 이 활동은 피어 유형에 따라 달라집니다.

- 

Routes from a nonclient peer(비클라이언트 피어의 경로) - 클러스터 내의 모든 클라이언트에 반영됩니다.

- 

Routes from a client peer(클라이언트 피어에서 경로) - 모든 비클라이언트 피어 및 클라이언트 피어에도 반영됩니다.

- Routes from an eBGP peer(eBGP 피어의 경로) - 모든 클라이언트 및 비 클라이언트 피어에 업데이트를 전송합니다.

라우터 RTC, RTD 및 RTB의 상대적 BGP 컨피그레이션은 다음과 같습니다.

```
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
```

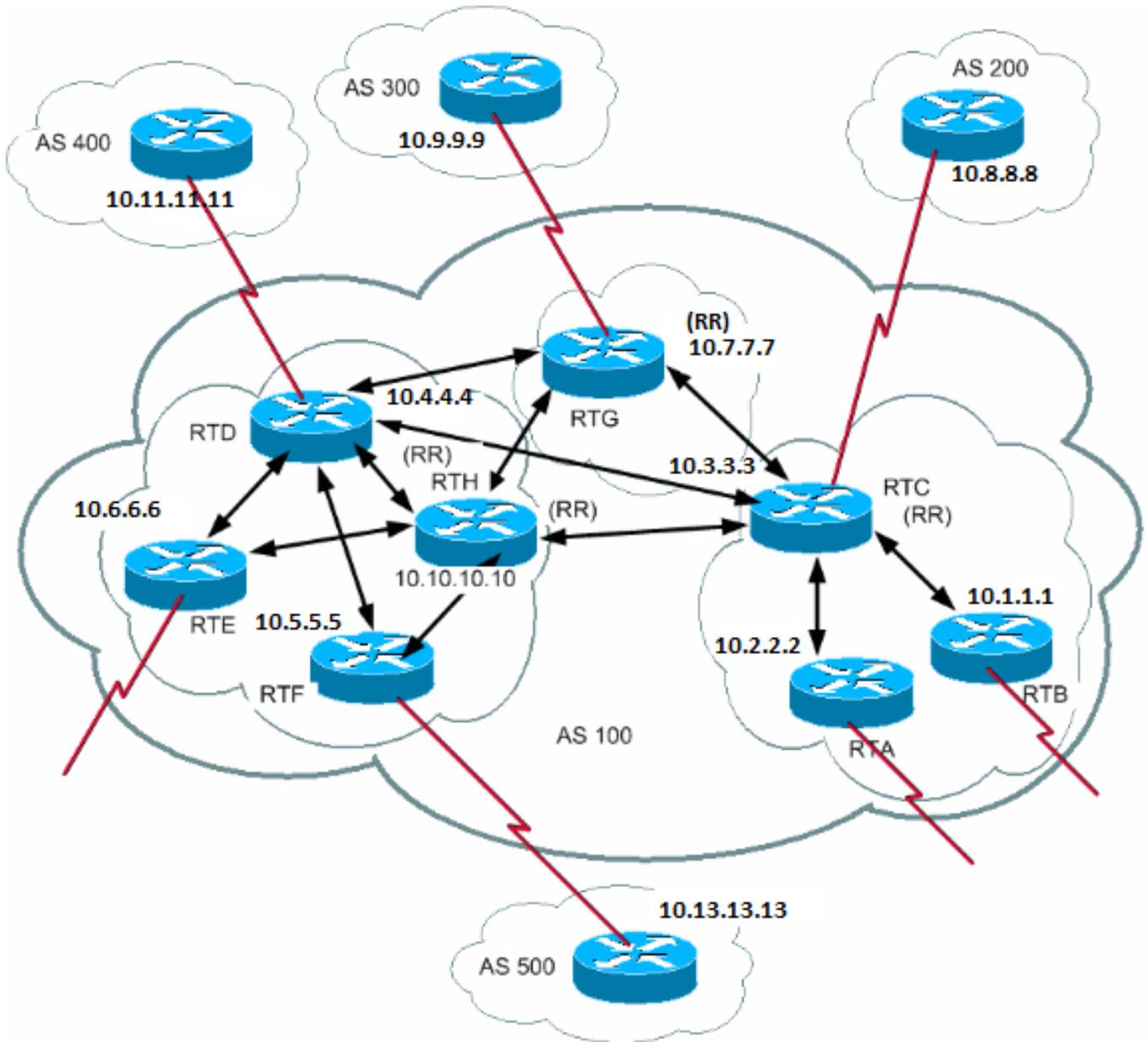
```
RTD#
router bgp 100
neighbor 10.6.6.16 remote-as 100
neighbor 10.6.6.16 route-reflector-client
neighbor 10.5.5.5 remote-as 100
neighbor 10.5.5.5 route-reflector-client
neighbor 10.7.7.7 remote-as 100
neighbor 10.3.3.3 remote-as 100
```

iBGP에서 학습한 경로가 반영되므로 라우팅 정보 루프가 있을 수 있습니다. RR 체계에는 이 루프를 방지하는 몇 가지 방법이 있습니다.

- **originator-id** — 길이가 4바이트인 선택적 비전이성 BGP 특성입니다. RR은 이 특성을 생성합니다. 이 특성은 로컬 AS에서 경로 발신자의 라우터 ID(RID)를 전달합니다. 컨피그레이션 불량으로 인해 라우팅 정보가 발신자에게 반환되면 정보가 무시됩니다.

- **cluster-list** - 한 클러스터 내의 여러 RR이 클러스터 목록을 다룹니다.

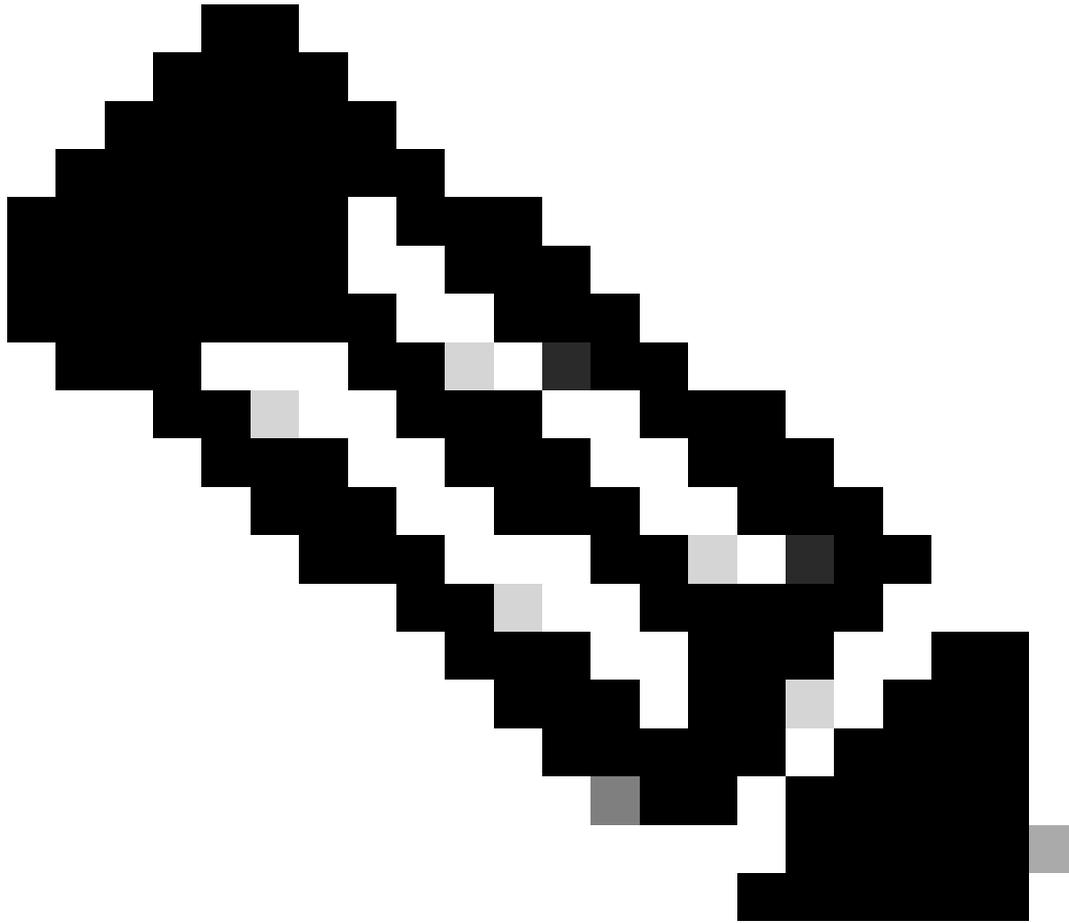
클러스터 내의 여러 RR



일반적으로 클라이언트 클러스터에는 단일 RR이 있습니다. 이 경우 RR의 라우터 ID는 클러스터를 식별합니다. 이중화를 높이고 단일 장애 지점을 방지하기 위해 클러스터에는 둘 이상의 RR이 있을 수 있습니다. RR이 동일한 클러스터에 있는 RR의 업데이트를 인식할 수 있도록 동일한 클러스터에 있는 모든 RR을 4바이트 클러스터 ID로 구성해야 합니다.

클러스터 목록은 경로가 통과한 클러스터 ID의 시퀀스입니다. RR이 RR 클라이언트에서 클러스터 외부의 비클라이언트로의 경로를 반영하면 RR은 로컬 클러스터 ID를 클러스터 목록에 추가합니다. 이 업데이트에 빈 클러스터 목록이 있으면 RR에서 이를 생성합니다. 이 특성을 사용하면 RR은 컨피그레이션이 잘못되어 라우팅 정보가 동일한 클러스터로 다시 루프백되었는지 확인할 수 있습니다. 로컬 클러스터 ID가 클러스터 목록에 있으면 알림이 무시됩니다.

이 절의 다이어그램에서 RTD, RTE, RTF, RTH는 하나의 클러스터에 속한다. RTD와 RTH는 모두 동일한 클러스터에 대한 RR입니다.



참고: RTH가 모든 RR과 완전히 연결된 피어링을 수행하므로 이중화가 있습니다. RTD가 다운되면 RTD 대신 RTH가 됩니다.

---

RTH, RTD, RTF 및 RTC의 구성은 다음과 같습니다.

```
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#

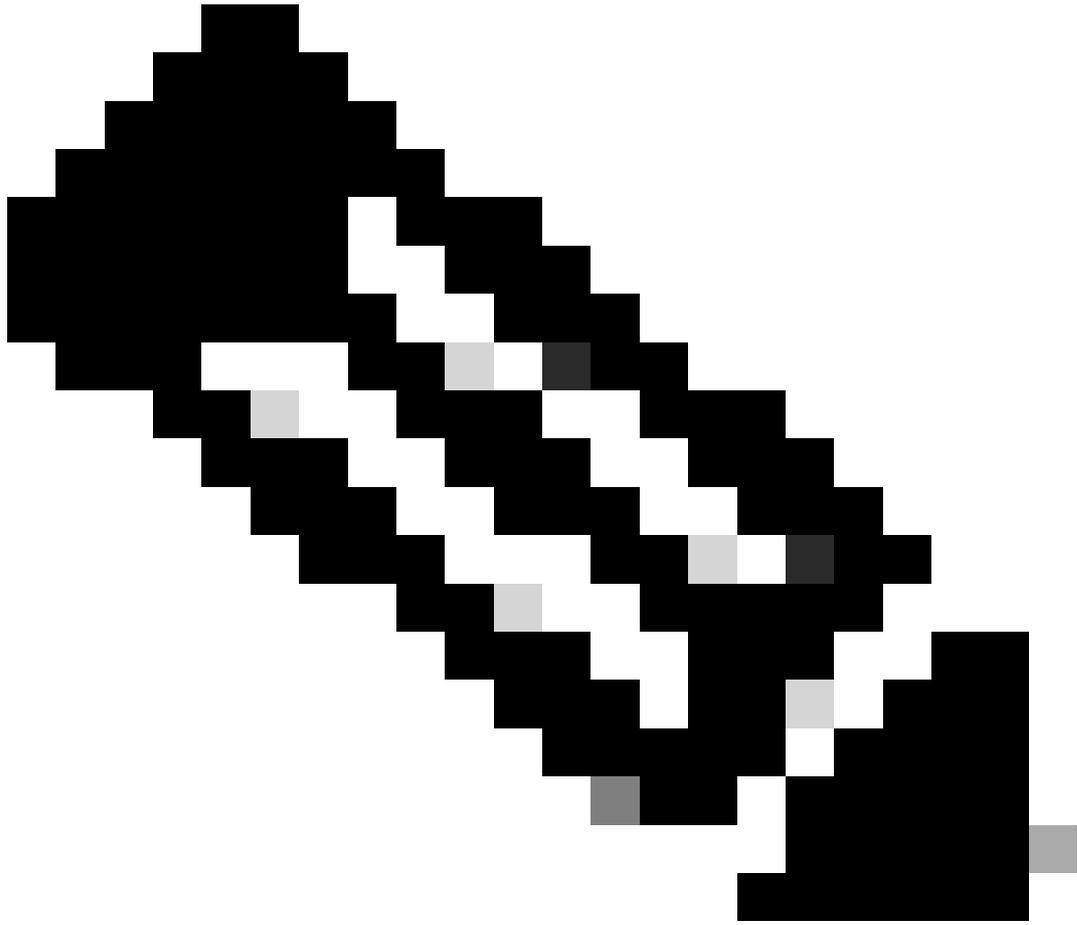
```
router bgp 100
neighbor 10.10.10.10 remote-as 100
neighbor 10.5.5.5 remote-as 100
neighbor 10.5.5.5 route-reflector-client
neighbor 10.6.6.16 remote-as 100
neighbor 10.6.6.16 route-reflector-client
neighbor 10.7.7.7 remote-as 100
neighbor 10.3.3.3 remote-as 100
neighbor 10.11.11.11 remote-as 400
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
```

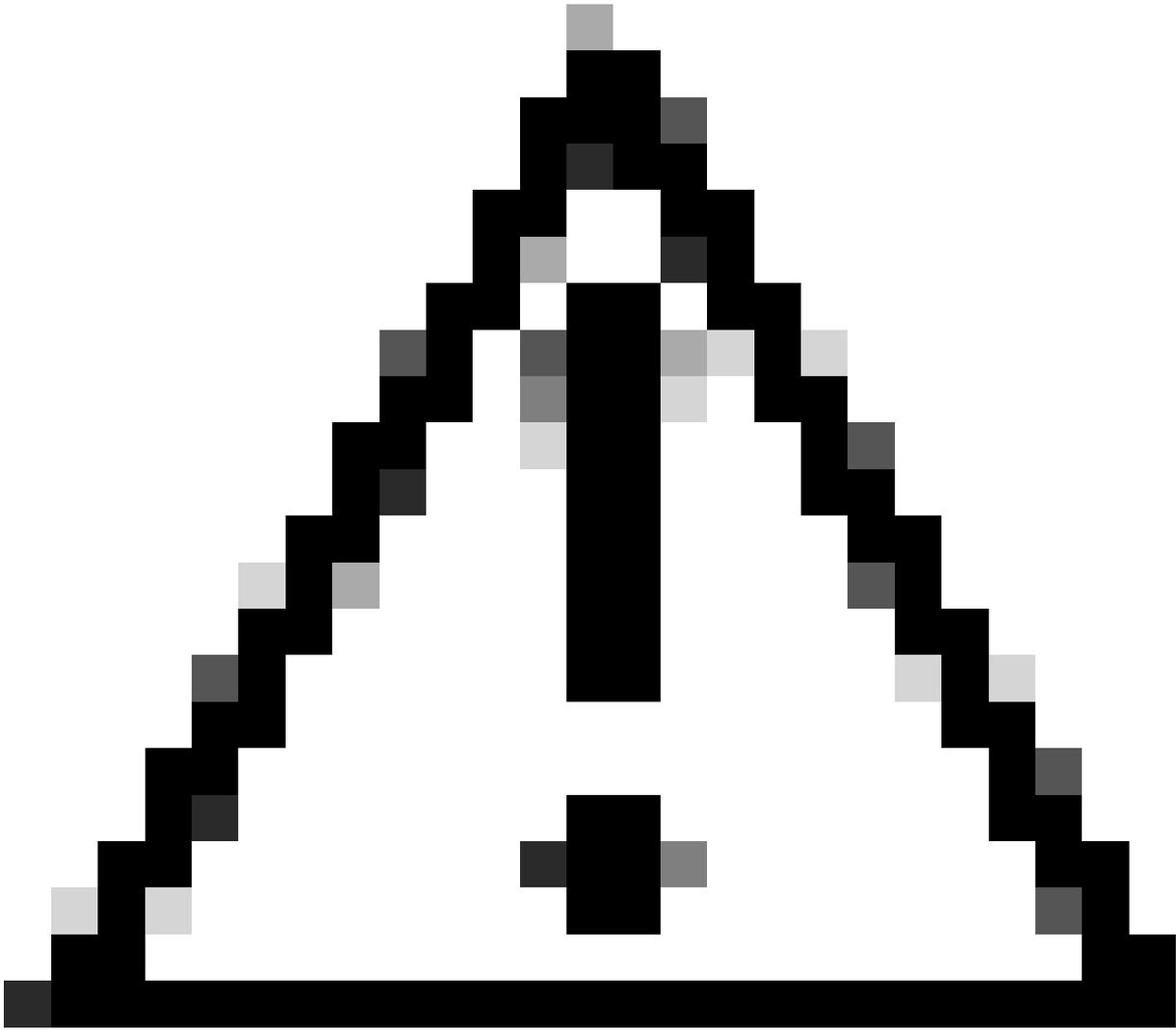
RTC#

```
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
```



참고: 해당 클러스터에 RR이 하나만 있으므로 RTC에 대해 `bgp cluster-id` 명령이 필요하지 않습니다.

---



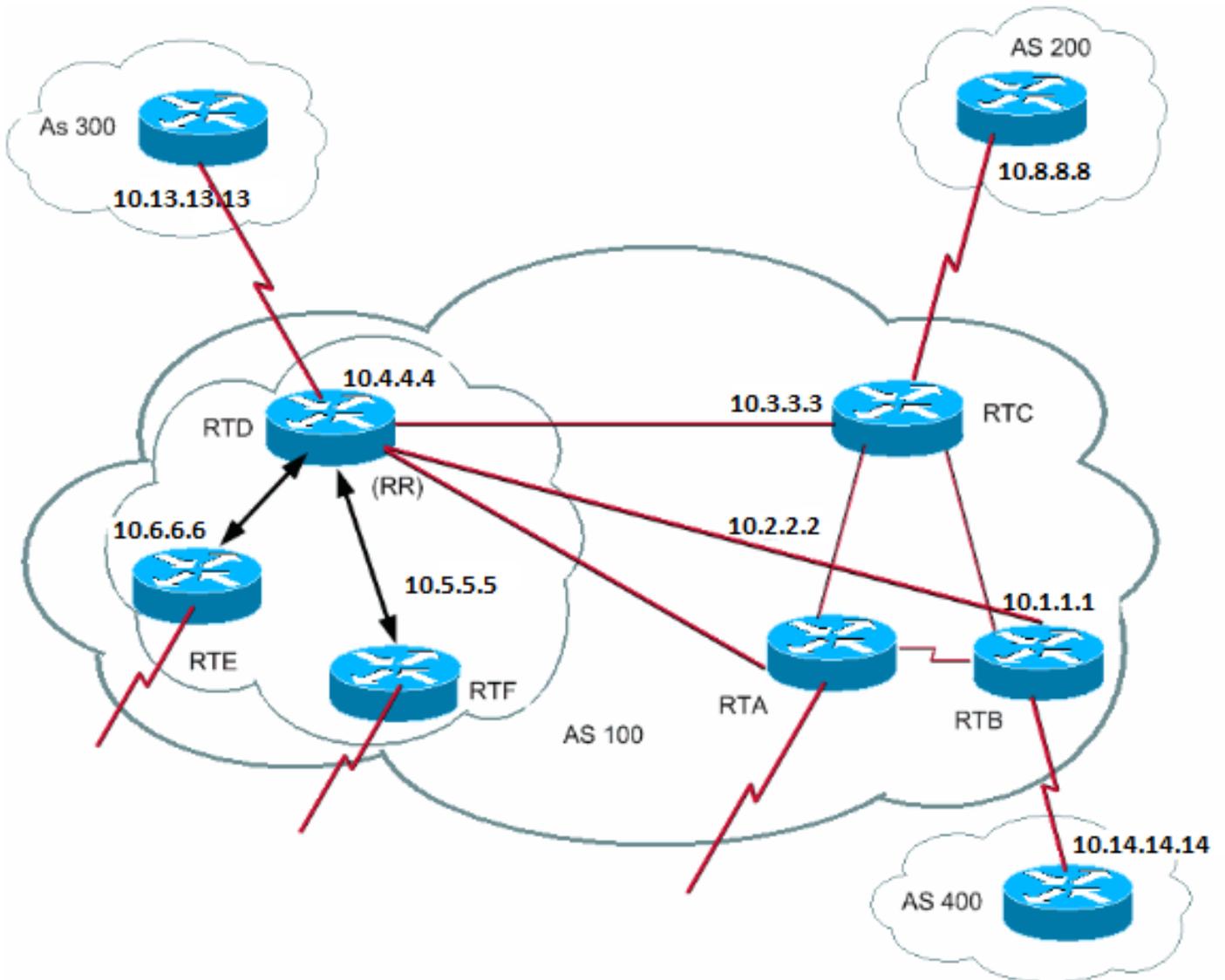
주의: 이 컨피그레이션에서는 피어 그룹을 사용하지 않습니다. 클러스터 내의 클라이언트에 서로 직접적인 iBGP 피어가 없고 클라이언트가 RR을 통해 업데이트를 교환하는 경우 피어 그룹을 사용하지 마십시오. 피어 그룹을 구성하는 경우 RR의 경로 소스에 대한 잠재적 철회가 클러스터 내의 모든 클라이언트에 전송됩니다. 이 전송으로 인해 문제가 발생할 수 있습니다.

라우터 하위 명령 `bgp client-to-client 리플렉션` RR에서 기본적으로 활성화되어 있습니다. RR에서 BGP 클라이언트-클라이언트 리플렉션을 해제한 상태에서 클라이언트 간에 이중화된 BGP 피어링을 만드는 경우 피어 그룹을 안전하게 사용할 수 있습니다. 자세한 내용은 [피어 그룹](#)의 제한을 참조하십시오.

RR 및 일반 BGP 스피커

AS에는 RR 개념을 모르는 BGP 스피커가 있을 수 있습니다. 이 문서에서는 이러한 라우터를 일반 BGP 스피커라고 합니다. RR 방식

은 이러한 기존 BGP 스피커가 공존할 수 있게 합니다. 이러한 라우터는 클라이언트 그룹 또는 비클라이언트 그룹의 멤버가 될 수 있습니다. 이러한 라우터가 있으면 현재 iBGP 모델에서 RR 모델로 쉽고 점진적으로 마이그레이션할 수 있습니다. 단일 라우터를 RR로 구성하고 다른 RR 및 RR 클라이언트를 정상적인 iBGP 피어로 만드는 경우 클러스터 생성을 시작할 수 있습니다. 그런 다음 점차적으로 클러스터를 더 많이 생성할 수 있습니다.



이 다이어그램에서 RTD, RTE, RTF는 경로 반사의 개념을 가지고 있다. RTC, RTA, RTB는 기존 라우터입니다. 이러한 라우터는 RR로 구성할 수 없습니다. 이러한 라우터와 RTD 간에 일반적인 iBGP 메시지를 수행할 수 있습니다. 나중에 업그레이드할 준비가 되면 RTC를 클라이언트 RTA 및 RTB와 함께 RR로 만들 수 있습니다. 클라이언트는 경로 반사 체계를 이해할 필요가 없습니다. RR만 업그레이드하면 됩니다.

다음은 RTD 및 RTC의 구성입니다.

```

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
```

RTC를 업그레이드하고 RTC를 RR로 만들 준비가 되면 iBGP 풀 메시지를 제거하고 RTA 및 RTB가 RTC의 클라이언트가 되도록 합니다

라우팅 정보의 루프 방지

지금까지 이 문서에서는 잠재적 정보 루프를 방지하는 데 사용할 수 있는 두 가지 특성(**originator-id** 및 **cluster-list**)에 대해 언급했습니다.

루프를 제어하기 위한 또 다른 수단은 아웃바운드 경로 맵의 **set** 절에 더 많은 제한을 두는 것이다. 아웃바운드 경로 맵에 대한 **set** 절은 iBGP 피어에 반영되는 경로에 영향을 주지 않습니다.

또한 neighbor별 컨피그레이션 옵션인 **next-hop-self**에 더 많은 제한을 둘 수 있습니다. **next-hop-self** n RR을 사용할 경우, 이 절은 eBGP에서 학습한 경로의 다음 홉에만 영향을 줍니다. 반영된 경로의 다음 홉은 변경하지 않아야 하기 때문입니다.

경로 플랩 댐핑

Cisco IOS Software 릴리스 11.0에는 경로 댐핑이 도입되었습니다. 경로 댐핑은 경로 플래핑이 유발하는 불안정성을 최소화하기 위한 메커니즘이다. 경로 댐핑은 또한 네트워크를 통한 진동을 줄입니다. 잘못된 동작 경로를 식별하기 위한 기준을 정의합니다. 플랩 경로는 각 플랩에 대해 1000의 페널티를 받습니다. 누적 페널티가 미리 정의된 억제 제한에 도달하는 즉시 경로 알림의 억제가 발생합니다. 벌금은 미리 구성된 반감기를 기준으로 기하급수적으로 감소합니다. 사전 정의된 재사용 제한에서 페널티가 감소하면 경로 알림이 더 이상 억제되지 않습니다.

AS의 외부에 있고 iBGP를 통해 학습된 경로에는 경로 댐핑이 적용되지 않습니다. 이러한 방식으로 경로 댐핑은 AS 외부 경로의 iBGP 피어에 대해 더 높은 페널티를 방지합니다.

벌금은 5초 단위로 줄어듭니다. 경로는 10초 단위로 억제되지 않습니다. 라우터는 페널티가 재사용 제한의 절반 미만이 될 때까지 댐핑 정보를 보관합니다. 이 시점에서 라우터는 정보를 삭제합니다.

초기에는 댐핑이 기본적으로 해제되어 있습니다. 필요한 경우 이 기능을 향후 기본 활성화로 지정할 수 있습니다. 다음 명령은 경로 댐핑을 제어합니다.

•

**bgp dampening** - 댐핑을 설정합니다.

•

**no bgp dampening** - 댐핑을 해제합니다.

- 

**bgp dampeninhalf-life-time** - 반감기 시간을 변경합니다.

모든 매개변수를 동시에 설정하는 명령은 다음과 같습니다.

- 

**bgp 댐핑 Half-life-time reuse suppress maximum-suppress-time**

이 목록에서는 구문에 대해 자세히 설명합니다.

- 

**half-life-time** - 범위는 1~45분이며 현재 기본값은 15분입니다.

- 

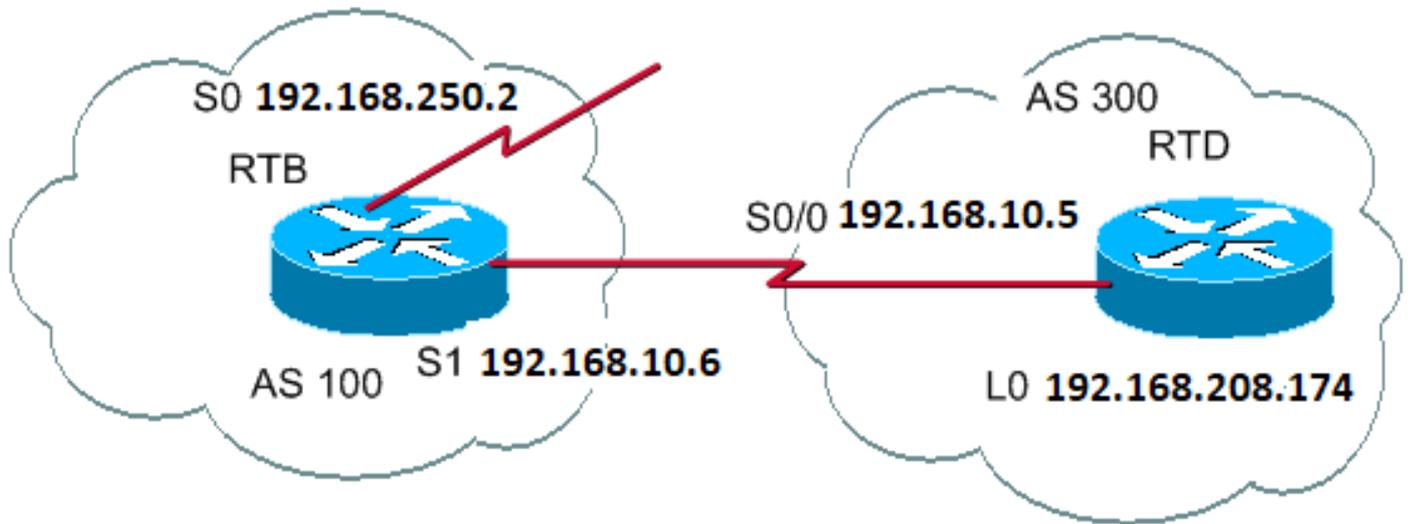
**reuse-value** - 범위는 1~20,000이고 기본값은 750입니다.

- 

**suppress-value** - 범위는 1~20,000이고 기본값은 2000입니다.

- 

**max-suppress-time** — 경로를 억제할 수 있는 최대 기간입니다. 범위는 1~255분이며 기본값은 반감기 시간의 4배입니다.



```

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

```

RTB의 컨피그레이션은 기본 매개변수로 경로 댐핑을 위한 것입니다. RTD에 대한 eBGP 링크가 안정적이라고 가정하면 RTB BGP 테이블은 다음과 같습니다.

<#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

경로 플랩을 시뮬레이션하려면 RTD에서 `clear ip bgp 192.168.10.6` 명령을 실행합니다. RTB BGP 테이블은 다음과 같습니다.

```
<#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

192.168.10.0에 대한 BGP 항목이 `ahistorystate`입니다. 이렇게 배치하면 경로에 대한 최상의 경로가 없지만 경로 플래핑에 대한 정보는 여전히 존재합니다.

```
<#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
```

노선이 플라핑으로 페널티를 받았지만, 페널티킥은 여전히 억제 한계 아래에 있다. 기본값은 2000입니다. 경로 삭제가 아직 발생하지 않았습니다. 경로가 몇 번 더 플랩하면 다음을 확인할 수 있습니다.

<#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

경로가 줄어들거나 억제되었습니다. 페널티가 "재사용 값"에 도달하면 경로가 재사용됩니다. 이 경우 재사용 값은 기본값인 750입니다. 댐핑 정보는 페널티가 재사용 제한의 절반 미만이 될 때 삭제됩니다. 이 때, 퍼지는 페널티가 375가 될 때(750/2=375) 발생합니다. 다음 명령은 플랩 통계 정보를 표시하고 지웁니다.

- 

**show ip bgp flap-statistics** - 모든 경로에 대한 플랩 통계를 표시합니다.

- 

**show ip bgp flap-statistics regexregular-expression** - 정규식과 일치하는 모든 경로에 대한 플랩 통계를 표시합니다.

- 

**show ip bgp flap-statistics filter-listlist** — 필터를 통과한 모든 경로에 대한 플랩 통계를 표시합니다.

- 

**show ip bgp flap-statisticsA.B.C.D m.m.m.m** - 단일 항목에 대한 플랩 통계를 표시합니다.

- 

**show ip bgp flap-statisticsA.B.C.D m.m.m.mlonger-prefix** — 보다 구체적인 항목에 대한 플랩 통계를 표시합니다.

- 

**show ip bgp neighbor [감쇠 경로] | [flap-statistics]** - 네이버의 모든 경로에 대한 플랩 통계를 표시합니다.

- 

**clear ip bgp flap-statistics** - 모든 경로에 대한 플랩 통계를 지웁니다.

- 

**clear ip bgp flap-statistics regexregular-expression** - 정규식과 일치하는 모든 경로에 대한 플랩 통계를 지웁니다.

- 

`clear ip bgp flap-statistics filter-listlist` — 필터를 통과한 모든 경로에 대한 플랩 통계를 지웁니다.

- 

`clear ip bgp flap-statisticsA.B.C.D m.m.m.m` - 단일 항목에 대한 플랩 통계를 지웁니다.

- 

`clear ip bgpA.B.C.Dflap-statistics` - 네이버의 모든 경로에 대한 플랩 통계를 지웁니다.

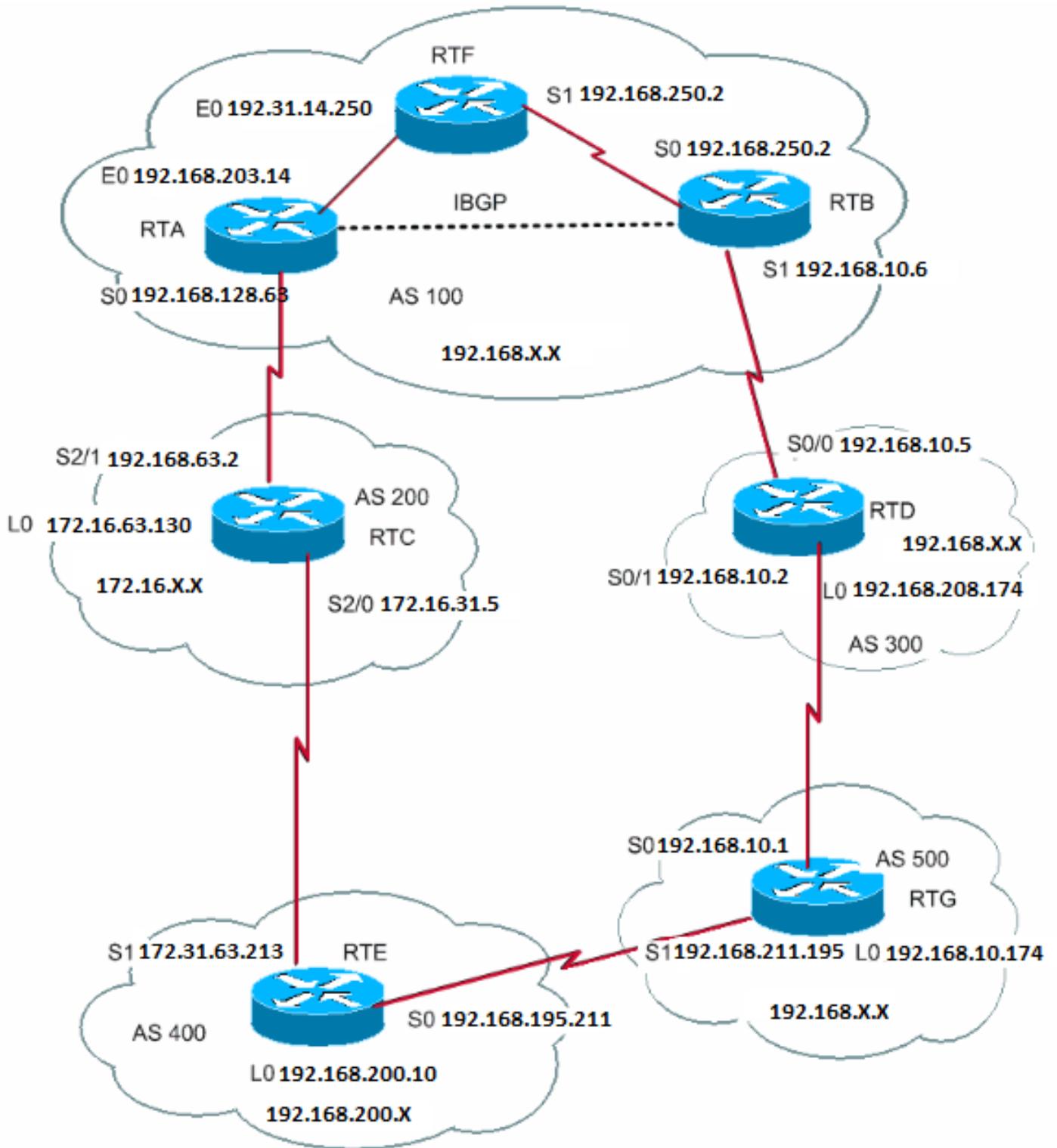
BGP가 경로를 선택하는 방법

BGP 특성 및 용어에 대해 잘 알고 있으므로 BGP Best [Path Selection Algorithm](#)을 참조하십시오.

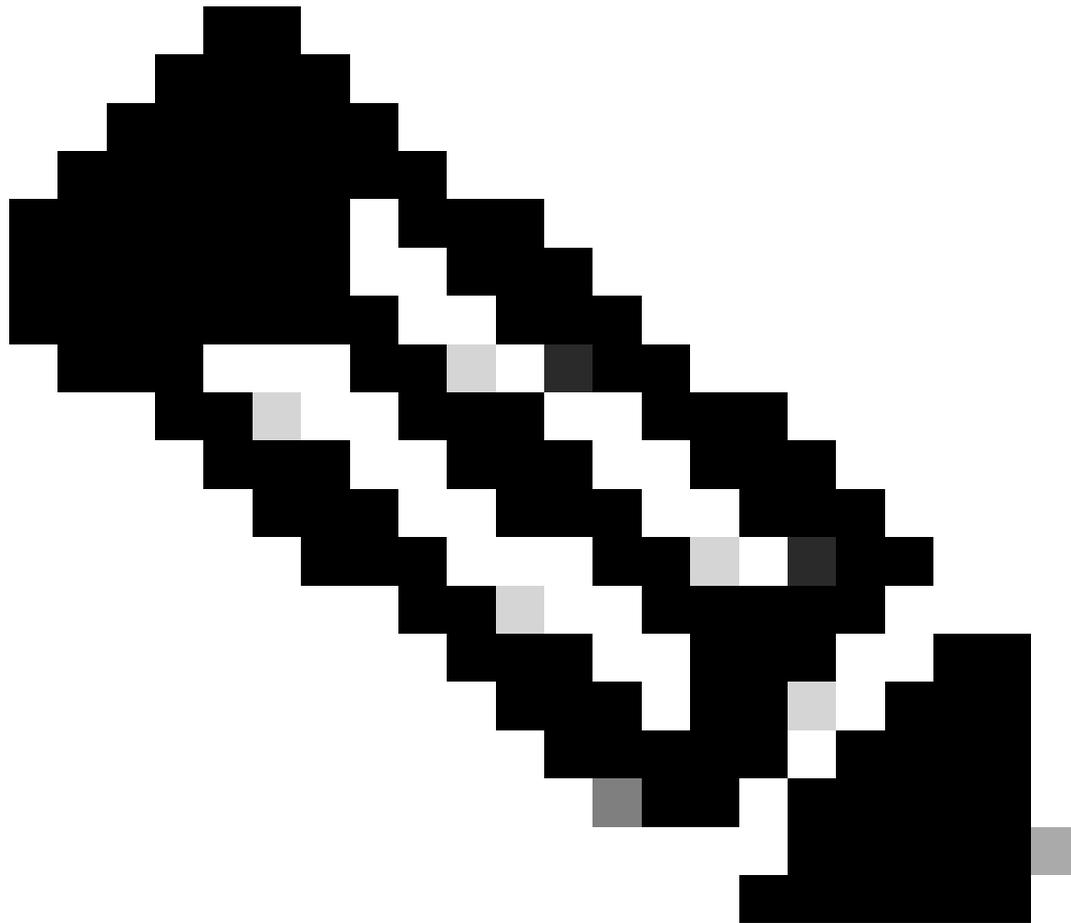
BGP 사례 연구 5

실용적인 디자인 예시

이 섹션에는 Cisco 라우터에 테이블이 실제로 나타나는 컨피그레이션 및 라우팅 테이블을 보여 주는 디자인 예제가 포함되어 있습니다.



이 섹션에서는 이러한 컨피그레이션을 단계적으로 구축하는 방법과 그 과정에서 발생할 수 있는 문제를 보여줍니다. eBGP를 통해 두 ISP에 연결하는 AS가 있을 때마다 AS 내에서 iBGP를 실행하여 경로를 더 효과적으로 제어할 수 있습니다. 이 예에서 iBGP는 AS100 내에서 RTA와 RTB 사이에서 실행되고 OSPF는 IGP로 실행됩니다. 두 ISP(AS200 및 AS300)에 연결한다고 가정합니다. 이는 모든 라우터에 대한 컨피그레이션의 첫 번째 실행입니다.



참고: 이러한 컨피그레이션은 최종 컨피그레이션이 아닙니다.

---

```
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
```

네트워크를 광고하려면 항상 network 이 명령을 사용하거나 고정 엔트리를 BGP로 재배포합니다. 이 방법은 IGP를 BGP로 재배포하는 것보다 우수합니다. 이 예에서는 명령을 network 사용하여 BGP에 네트워크를 삽입합니다.

여기서는 RTB와 RTD 간의 링크가 없는 것처럼 RTB 종료 시 s1 인터페이스로 시작합니다. 다음은 RTB BGP 테이블입니다.

```
<#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
```

이 표에는 다음과 같은 표기가 표시됩니다.

- 

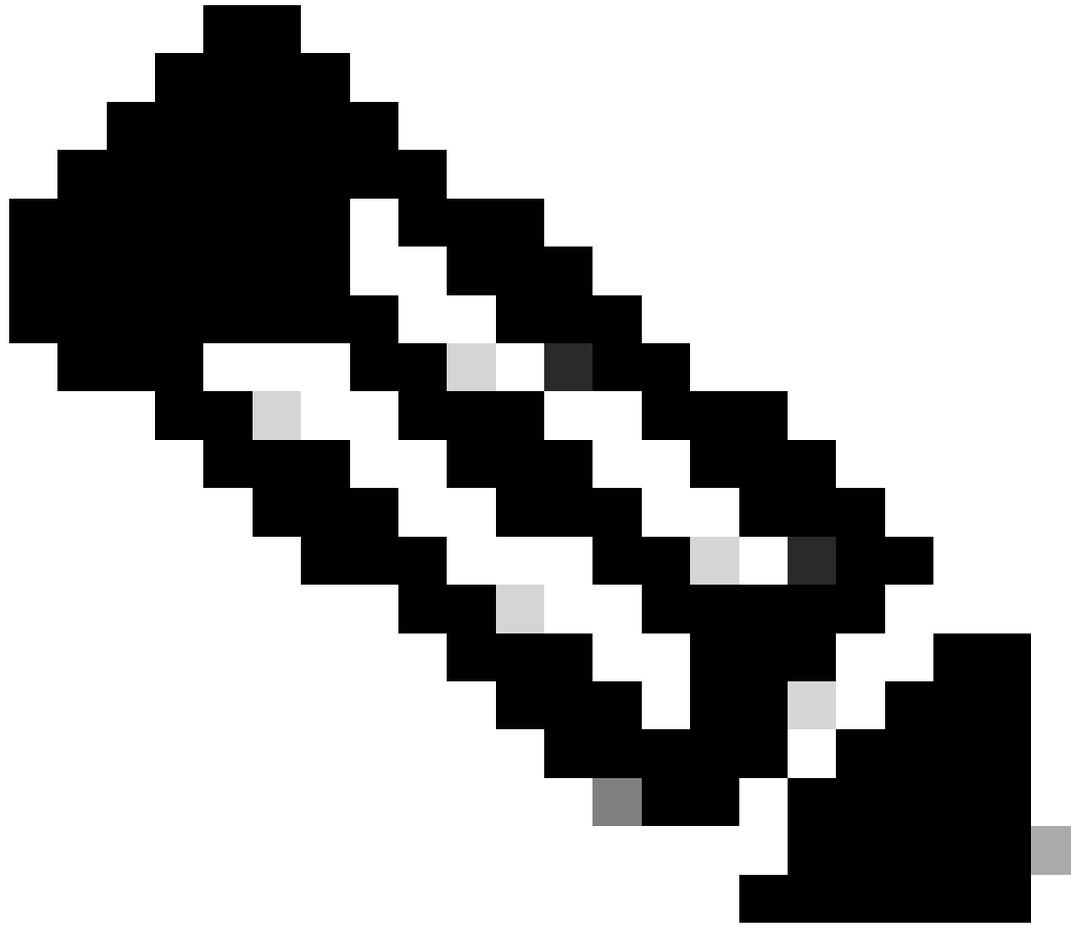
Aniat the beginning - iBGP 피어를 통해 항목을 학습했음을 나타냅니다.

- 

Aniat the end - 경로 정보의 원본이 IGP임을 나타냅니다.

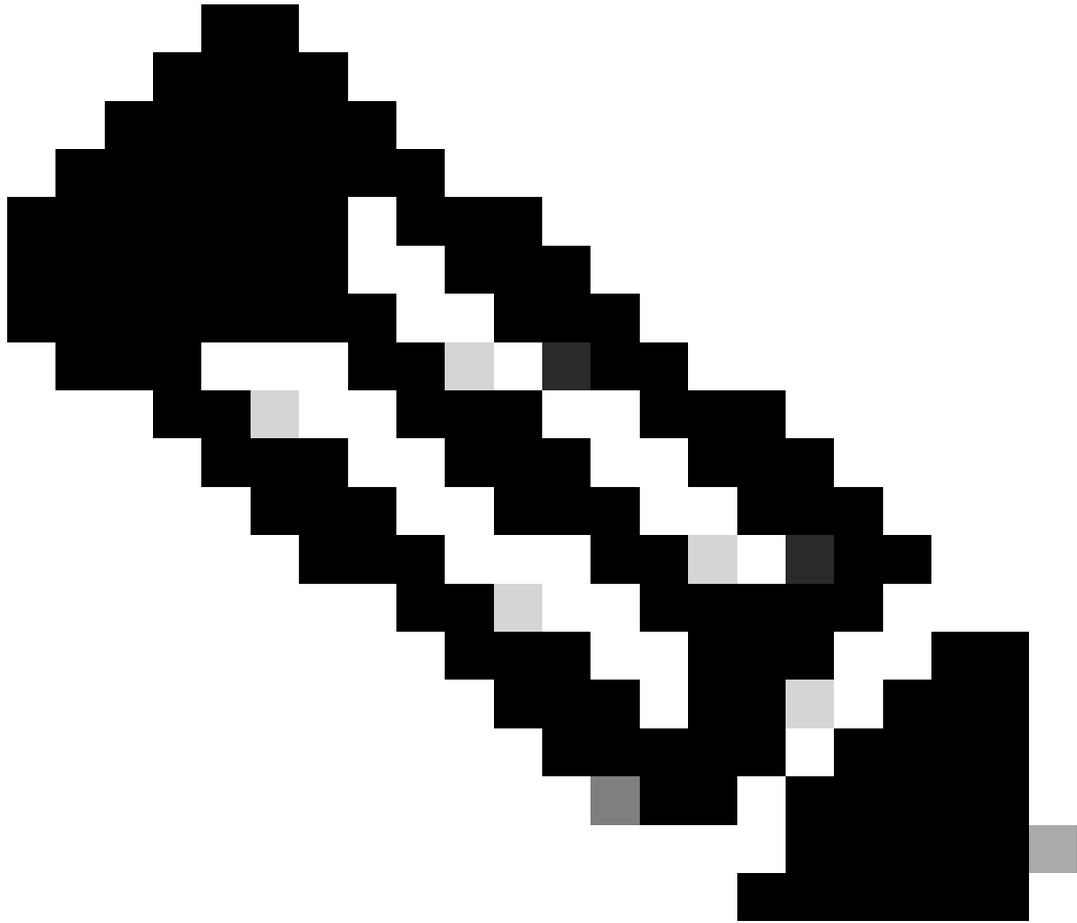
- 

Pathinformation(경로 정보) - 이 정보는 직관적입니다. 예를 들어, 네트워크 172.31.10.0은 경로 200을 통해 학습되며 다음 홉은 172.31.63.250입니다.



**참고:** 로컬에서 생성된 모든 항목(예: 192.168.250.15)에는 다음 홉 0.0.0.0이 있습니다.

- 
- An>symbol - BGP가 최상의 경로를 선택했음을 나타냅니다. BGP는 문서 BGP 최적 경로 선택 알고리즘이 [설명하는](#) 결정 단계를 사용합니다. BGP는 대상에 도달하는 데 가장 적합한 경로를 하나 선택하고, IP 라우팅 테이블에 경로를 설치하고, 다른 BGP 피어에 경로를 광고합니다.



**참고:** Next Hop 특성을 확인합니다. RTB는 iBGP로 전달되는 eBGP 다음 홉인 172.31.63.250의 다음 홉을 통해 172.31.10.0에 대해 알고 있습니다.

---

IP 라우팅 테이블을 확인합니다.

<#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
```

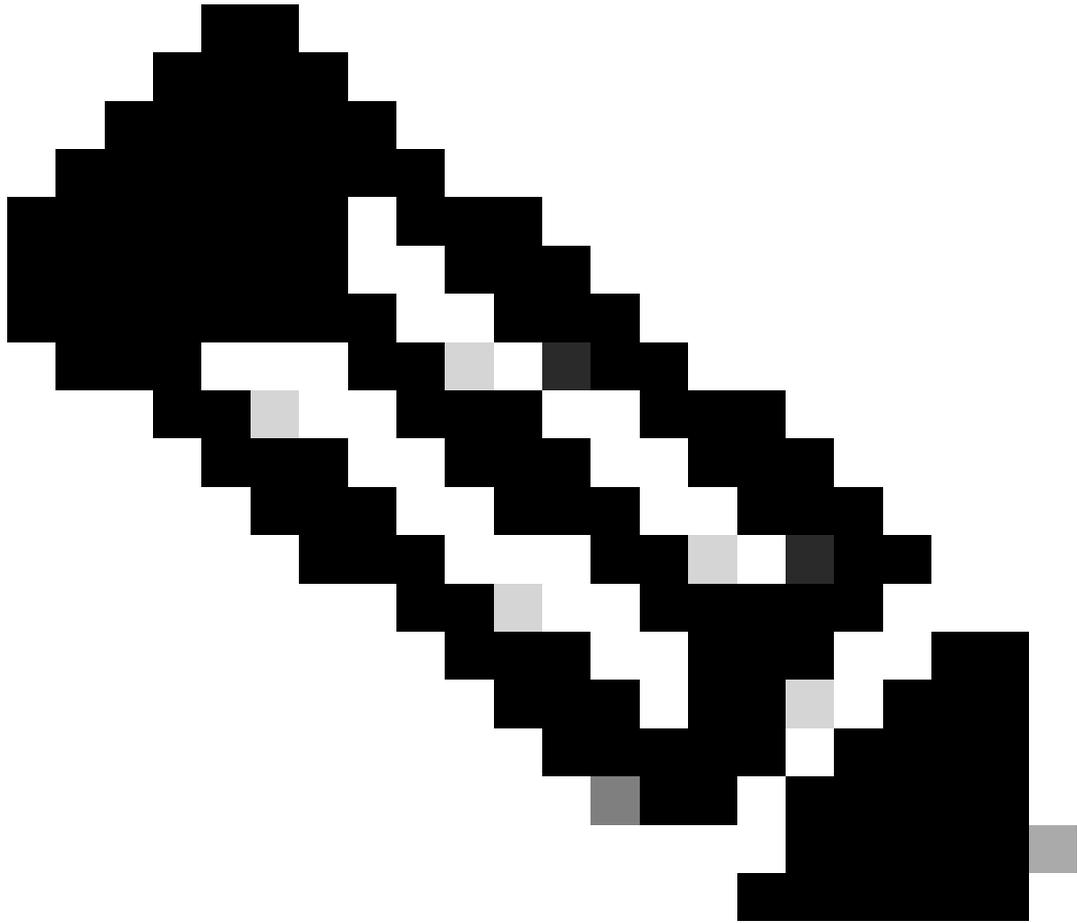
```
Gateway of last resort is not set
```

```
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
```

BGP 항목 중 라우팅 테이블에 도달한 항목이 없습니다. 여기에는 두 가지 문제가 있습니다.

첫 번째 문제는 이러한 항목의 다음 홉인 172.31.63.250에 연결할 수 없다는 것입니다. 이 IGP, 즉 OSPF를 통해 다음 홉에 도달할 수 있는 방법은 없습니다. RTB는 OSPF를 통해 192.168.213.63에 대해 학습하지 않았습니다. RTA s0 인터페이스에서 OSPF를 실행하여 패시브로 설정할 수 있습니다. 이 방법으로 RTB는 다음 홉 172.31.63.250에 도달하는 방법을 알 수 있습니다. 이 RTA 컨피그레이션은 여기에 표시됩니다.

```
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
```



참고: 다음 흡을bgp nexthop self 변경하기 위해 RTA와 RTB 간에 명령을 실행할 수 있습니다.

---

RTB의 새 BGP 테이블은 다음과 같습니다.

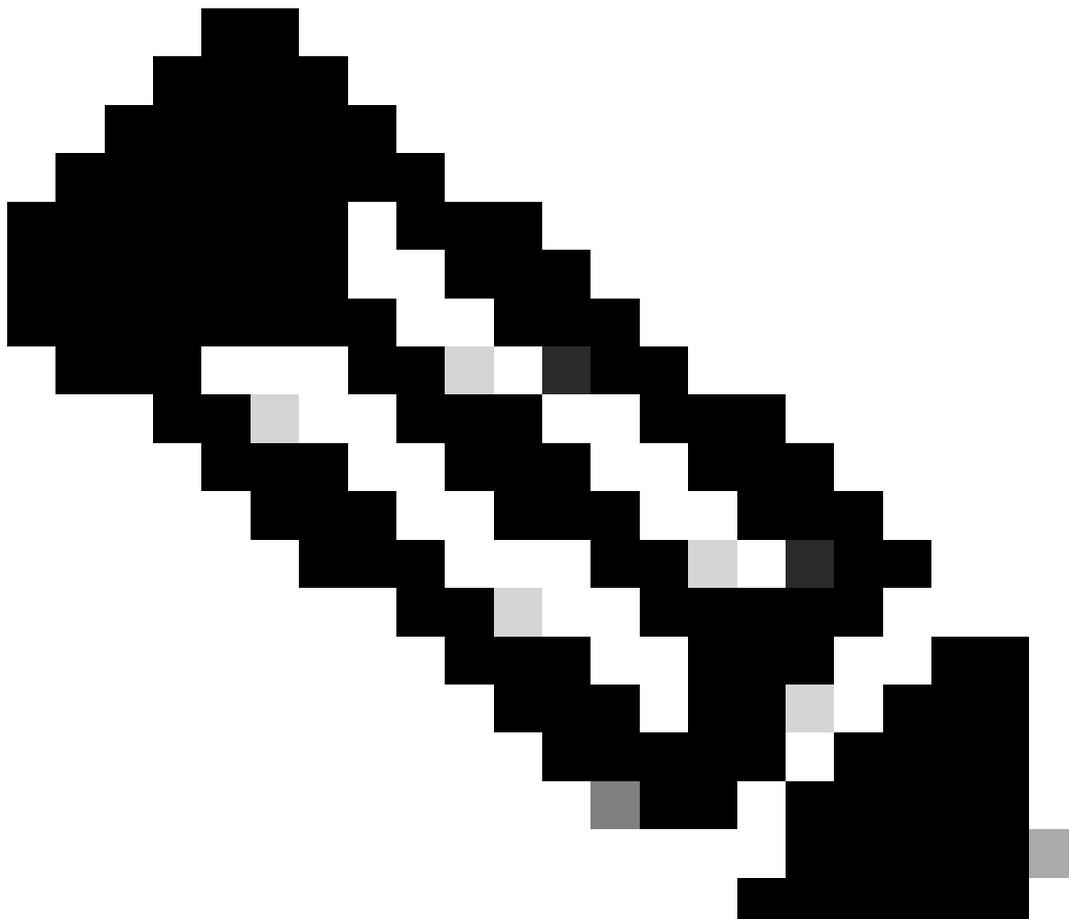
<#root>

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



---

참고: 모든 항목에는 >이(가) 있습니다. 즉, BGP가 다음 hops에 도달할 수 있습니다.

---

라우팅 테이블을 확인합니다.

```
<#root>
```

```
RTB#
```

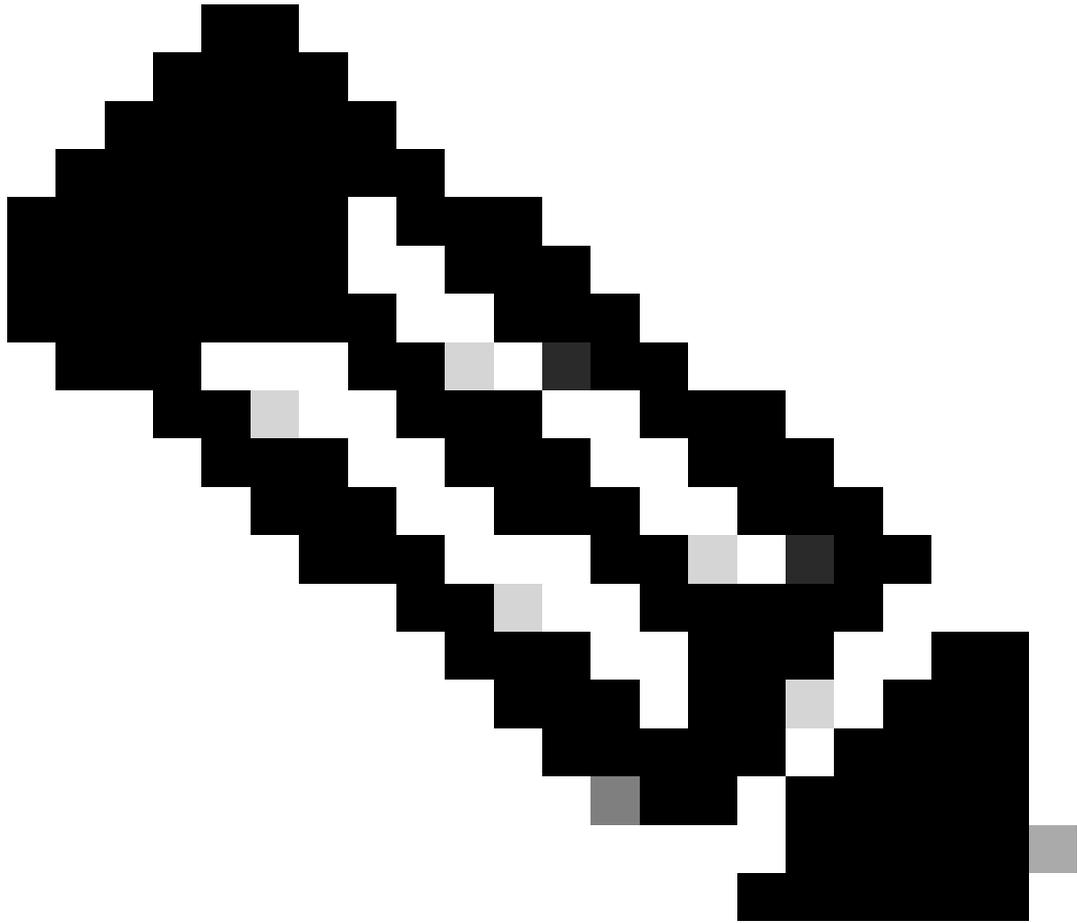
```
show ip route
```

```
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
```

두 번째 문제는 라우팅 테이블에 BGP 항목이 여전히 표시되지 않는다는 것입니다. 유일한 차이점은 이제 OSPF를 통해 192.168.213.63에 연결할 수 있다는 것입니다. 이 문제는 동기화 문제입니다. BGP는 이러한 엔트리를 라우팅 테이블에 배치하지 않으며 IGP와의 동기화 부족으로 인해 BGP 업데이트의 엔트리를 전송하지 않습니다.



참고: 아직 OSPF에 BGP를 재배포하지 않았으므로 RTF에는 네트워크 192.168.10.0 및 192.168.211.10에 대한 개념이 없습니다.

---

이 시나리오에서 동기화를 해제하면 항목이 라우팅 테이블에 나타납니다. 하지만 여전히 연결이 끊어졌습니다.

RTB에서 동기화를 해제하면 다음과 같이 됩니다.

```
<#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
```

라우팅 테이블은 정상적으로 보이지만 해당 네트워크에 연결할 수 있는 방법이 없습니다. 중간에 있는 RTF는 네트워크에 연결하는 방법을 모릅니다.

<#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
```

이 상황에서 동기화를 해제해도 문제가 여전히 존재합니다. 하지만 나중에 다른 문제로 인해 동기화가 필요합니다. RTA의 OSPF에 BGP를 재배포합니다(메트릭 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
```

라우팅 테이블은 다음과 같습니다.

```
<#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
```

OSPF가 iBGP보다 거리가 더 멀기 때문에 BGP 엔트리가 사라졌습니다. OSPF 거리는 110이고, iBGP 거리는 200입니다.

RTA에서 192.168.250.15를 알릴 수 있도록 RTA에서 동기화를 끕니다. 마스크 차이로 인해 RTA가 OSPF와 동기화되지 않으므로 이 작업이 필요합니다. RTB에서 동기화를 해제하여 RTB가 192.168.203.13을 알릴 수 있도록 합니다. 이 작업은 RTB에서도 같은 이유로 필요합니다.

이제 RTB s1 인터페이스를 불러와 경로의 모양을 확인합니다. 또한 RTB의 직렬 1에서 OSPF를 활성화하여 패시브로 만듭니다. 이 단계를 통해 RTA는 IGP를 통해 다음 홉 192.168.10.5에 대해 알 수 있습니다. 이 단계를 수행하지 않으면 라우팅 루프가 발생합니다. 다음 홉 192.168.10.5에 도달하려면 eBGP를 통해 반대 방향으로 이동해야 하기 때문입니다. 다음은 RTA 및 RTB의 새로운 컨피그레이션입니다.

```
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
```

BGP 테이블은 다음과 같습니다.

<#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

네트워크를 설계하여 두 개의 서로 다른 ISP(AS200 및 AS300)와 통신하는 방법에는 여러 가지가 있습니다. 한 가지 방법은 기본 ISP와 백업 ISP를 갖는 것입니다. ISP 중 하나에서 부분 경로를 학습하고 두 ISP에 대한 기본 경로를 학습할 수 있습니다. 이 예에서는 AS200에서 부분 경로를 수신하고 AS300에서 로컬 경로만 수신합니다. RTA와 RTB는 모두 OSPF로 기본 경로를 생성하며, 메트릭이 낮기 때문에 RTB를 기본 설정으로 사용합니다. 이 방법으로 두 ISP 간에 발신 트래픽의 균형을 맞출 수 있습니다.

RTA를 떠나는 트래픽이 RTB를 통해 다시 돌아오는 경우 잠재적인 비대칭이 발생할 수 있습니다. 두 ISP와 통신할 때 동일한 IP 주소 풀(동일한 주 네트워크)을 사용하는 경우 이러한 상황이 발생할 수 있습니다. 집계 때문에 전체 AS는 외부 세계에 대해 하나의 전체 엔티티처럼 보일 수 있습니다. 네트워크에 대한 진입 지점은 RTA 또는 RTB를 통해 발생할 수 있습니다. 인터넷에 여러 개의 포인트가 있는 경우에도 AS로 수신되는 모든 트래픽이 단일 포인트를 통해 수신됨을 확인할 수 있습니다. 이 예에서는 두 ISP에 대해 이야기할 때 서로 다른 두 개의 주 네트가 있습니다.

비대칭의 또 다른 잠재적인 이유는 AS에 도달할 수 있는 다른 알려진 경로 길이입니다. 아마도 한 통신 사업자가 다른 서비스 사업자보다 특정 대상에 더 가까울 수 있습니다. 이 예에서는 경로가 짧기 때문에 네트워크를 목적지로 하는 AS400의 트래픽이 항상 RTA를

통해 들어옵니다. 당신은 그 결정을 내리려고 노력할 수 있습니다. **set as-path prepend** 명령을 사용하여 업데이트에 경로 번호를 추가하고 경로 길이를 더 길게 보이게 할 수 있습니다. 그러나 로컬 환경 설정, 메트릭 또는 가중치와 같은 속성을 사용하는 경우 AS400에서 종료 지점을 AS200으로 설정할 수 있습니다. 이 경우 당신이 할 수 있는 일은 아무것도 없습니다.

이 컨피그레이션은 모든 라우터의 최종 컨피그레이션입니다.

```
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
```

RTA에서는 AS200에서 오는 경로에 대한 로컬 선호도가 200으로 설정됩니다. 또한 네트워크 172.31.200.200이 후보 기본값으로 선택됩니다. **ip default-network** 명령을 사용하면 기본값을 선택할 수 있습니다.

또한 이 예에서는 **default-information originate** 명령을 **OSPF**와 함께 사용하면 OSPF 도메인 내에 기본 경로가 삽입됩니다. 이 예에서는 또한 IS-IS Protocol(Intermediate System-to-Intermediate System Protocol) 및 BGP와 함께 이 명령을 사용합니다. RIP의 경우 추가 컨피그레이션 없이 RIP 0.0.0.0으로 자동 재배포됩니다. IGRP 및 EIGRP의 경우 IGRP 도메인으로 기본 정보 주입은 BGP를 IGRP 및 EIGRP로 재배포한 후에 발생합니다. 또한 IGRP 및 EIGRP를 사용하면 고정 경로를 0.0.0.0으로 IGP 도메인으로 재배포할 수 있습니다.

```

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

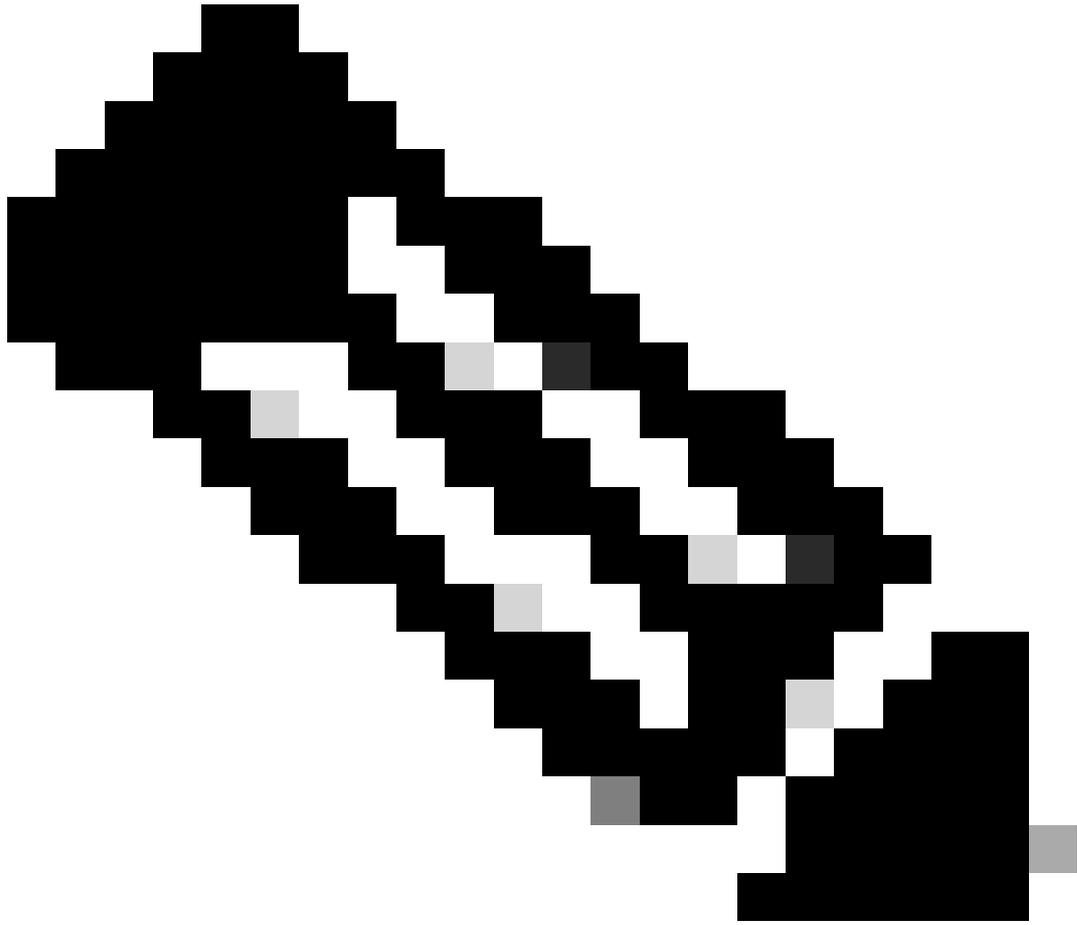
interface Serial0
 ip address 192.168.250.2 255.255.255.252
!
interface Serial1
 ip address 192.168.10.6 255.255.255.252

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

```

RTB의 경우 AS300에서 제공되는 업데이트에 대한 로컬 기본 설정이 300으로 설정됩니다. 이 값은 RTA에서 제공되는 iBGP 업데이트의 로컬 환경 설정 값보다 높습니다. 이러한 방식으로 AS100은 AS300의 로컬 경로에 대해 RTB를 선택합니다. RTB의 다른 모든 경로(다른 경로가 있는 경우)는 로컬 환경 설정 100으로 내부에서 전송합니다. 이 값은 RTA에서 제공되는 로컬 선호 값인 200보다 낮습니다. RTA가 기본 설정입니다.



참고: AS300 로컬 경로만 광고했습니다. ^300\$(과(와) 일치하지 않는 모든 경로 정보가 삭제됩니다. ISP의 고객인 로컬 경로와 인접 경로를 알려려면 ^300\_[0-9]\*를 사용합니다.

---

다음은 AS300 로컬 경로를 나타내는 정규식의 출력입니다.

<#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
```

RTC에서는 172.31.10.0/16을 집계하고 AS100에 삽입할 특정 경로를 나타냅니다. ISP가 이 작업을 거부하는 경우 AS100의 수신 끝을 필터링해야 합니다.

```
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 192.168.211.195 255.255.255.192

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

```

커뮤니티 필터링을 사용하는 방법에 대한 데모는 RTG에서 제공됩니다. RTD에 대한 no-export 192.168.211.0 업데이트에 커뮤니티를 추가합니다. 이러한 방식으로 RTD는 해당 경로를 RTB로 내보내지 않습니다. 그러나 이 경우 RTB는 어쨌든 이러한 경로를 수락하지 않습니다.

```

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
```

RTE는 172.31.200.200/16을 집계합니다. 다음은 RTA, RTF, RTB에 대한 최종 BGP 및 라우팅 테이블입니다.

```
<#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**

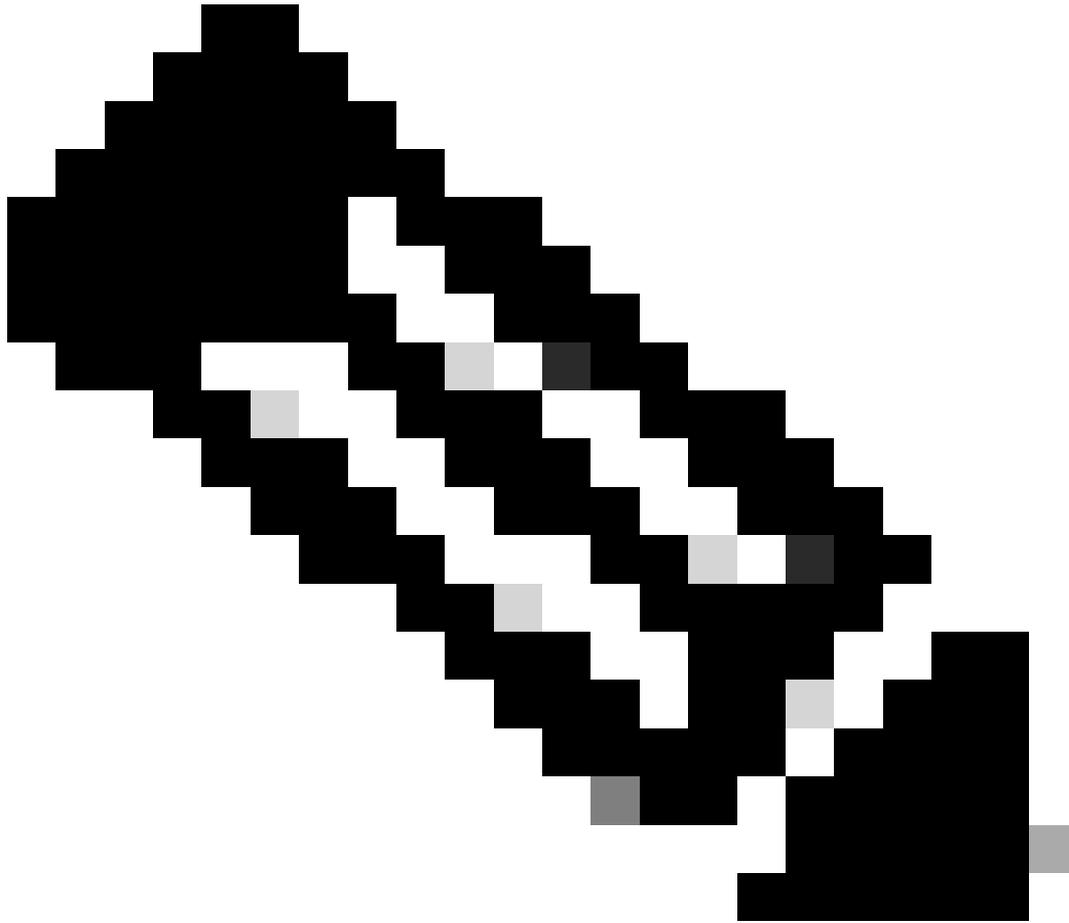
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP  
 D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
 E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP  
 i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, \* -  
 candidate default

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

```



**참고:** RTF 라우팅 테이블은 AS300 로컬 네트워크(예: 192.168.10.0)에 도달하는 방법이 RTB를 통하는 것임을 나타냅니다 . 172.31.200.200과 같은 다른 알려진 네트워크에 연결하는 방법은 RTA를 통하는 것입니다. 마지막 휴양지의 관문은 RTB로 설정되어 있습니다. RTB와 RTD 간의 연결에 어떤 일이 발생하면 RTA가 광고하는 기본값은 2000의 메트릭을 시작합니다.

---

<#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
```

관련 정보

- [BGP: FAQ\(자주 묻는 질문\)](#)
- [PIX 방화벽을 통한 BGP의 샘플 컨피그레이션](#)
- [HSRP를 사용하여 멀티홈 BGP 네트워크에서 이중화를 제공하는 방법](#)
- [Cat6000 MSFC에서 단일 라우터 모드 이중화 및 BGP 구성](#)
- [최적의 라우팅 달성 및 BGP 메모리 사용량 감소](#)
- [일반적인 BGP 문제 해결](#)
- [BGP 스캐너 또는 라우터 프로세스로 인한 높은 CPU 문제 해결](#)
- [단일 및 멀티 홈 환경에서 BGP를 사용한 로드 공유 이해](#)
- [BGP 지원 페이지](#)
- [Cisco 기술 지원 및 다운로드](#)

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.