

# Configuring BGP

Cisco Syntax

(C) Herbert Haas 2005/03/11

## Agenda



- **Basic Configuration Commands**
- **Next Hop Handling**
- **Route Reflectors**
- **BGP Policy Control: Route Maps**
- **BGP Policy Control: Prefix Lists**
- **Route Dampening**
- **Selected Topics**

Note:

This chapter provides an introduction and overview.  
Please consult the Cisco documentation for more details.

(C) Herbert Haas 2005/03/11

2

## Basic BGP Commands



- Enable BGP

```
Router(config)#  
router bgp AS-number
```

- Activate BGP session with another router (either IBGP or EBGP)

```
Router(config-router)#  
neighbor {ip-address|peer-group-name} remote-as AS-number
```

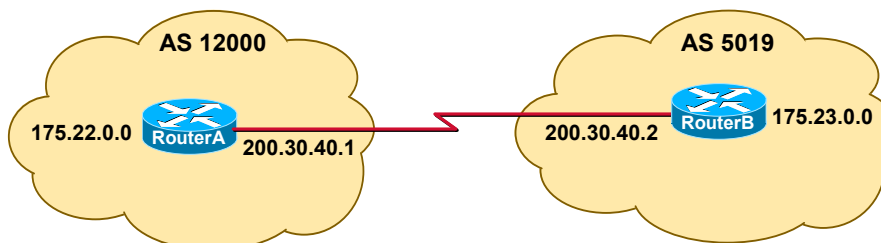
- Allow BGP to advertise IGP route if it is already in the IP table

```
Router(config-router)#  
network network-number mask network-mask
```

## Example



```
RouterA(config)# router bgp 12000  
RouterA(config-router)# neighbor 200.30.40.2 remote-as 5019  
RouterA(config-router)# network 175.22.0.0
```



```
RouterB(config)# router bgp 5019  
RouterB(config-router)# neighbor 200.30.40.1 remote-as 12000  
RouterB(config-router)# network 175.23.0.0
```

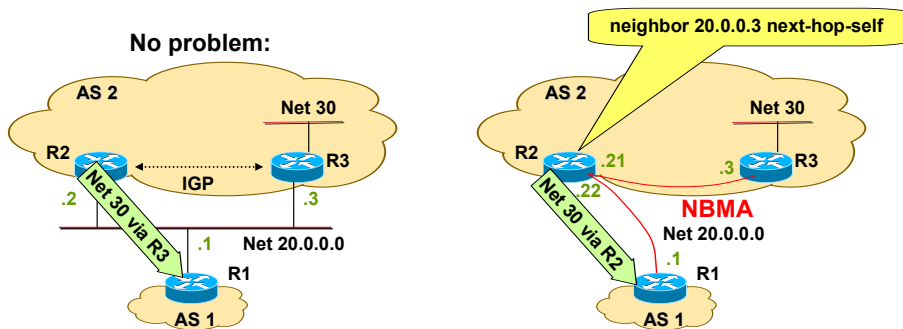
## Next Hop Handling



- In **partially meshed NBMA** topology the logical next hop might not be directly reachable
- Manipulate next-hop default behavior

```
Router (config-router) #
```

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



(C) Herbert Haas 2005/03/11

5

## Disabling Synchronization



- If there are only BGP routers inside the AS, disable synchronization for faster convergence

```
Router (config-router) #
```

```
no synchronization
```

- **Note:**  
Synchronization is on by default!

(C) Herbert Haas 2005/03/11

6

## Aggregate Routes



- Create a summary entry in the BGP table

```
Router(config-router)#
```

```
aggregate-address ip-address mask [summary-only] [as-set]
```

- ♦ "Summary only" to not advertise the specific routes
- If synchronization is performed, then the **summary route must occur in the routing table (!)**
  - ♦ Should be statically entered

## Resetting BGP



- After changing a BGP configuration the BGP connections should be reset

```
Router#
```

```
clear ip bgp {*|address} [soft[in|out]]
```

- This will disrupt routing, except using the **"soft"** keyword:
  - ♦ Sessions are not reset
  - ♦ Router sends all updates again

## Verifying BGP



- **Display information about BGP**

```
Router#  
show ip bgp [summary|neighbors]
```

- **Additional show ip bgp options:**

```
Router#  
show ip bgp [community|community-list|dampened-paths|  
filter-list|flap-statistics|inconsistent-as|paths|  
peer-group|regex]
```

- **Display BGP events in realtime**

```
Router#  
debug ip bgp
```

## Route Reflectors



- **Define this router as BGP route reflector and specify its clients**

```
Router(config-router)#  
neighbor ip-address route-reflector-client
```

- **Note: No configuration needed at clients**

## Policy Control: Route Maps



- Basic tool to configure policies and manipulate advertisements for BGP
- Route maps can be linked with the BGP neighbor command:

```
Router(config-router) #  
neighbor ip-address route-map map-tag in|out
```

- ♦ **in/out** specifies whether the route map should apply on incoming or outgoing BGP routing updates from/to that neighbor

## Basic Route Map Command



```
route-map map-tag [permit|deny] [sequence-number]  
! Match and set commands are following here...
```

- **map-tag** identifies route map
- **permit/deny** specifies whether this route should be propagated or not
- **sequence-number** indicates position of this entry within this route map

## Match Patterns and Set Policies



- **Match** statements are used if the policy statements should only apply on a subset of routes
  - ◆ Matching pattern defined with ACL
- The actual policy is defined with **set** commands

```
route-map MYMAP permit 10
match ip address 1
set local-preference 200
ip access-list 1 permit 1.2.3.0 0.0.0.255
```

## Important Match and Set Commands



(Arguments have been omitted)

match ip address

match community-list

match as-path

set metric

set local-preference

set ip [default] next-hop

set dampening

set as-path

set [default] interface

match length

match length

To remove communities from  
inbound or outbound updates

set comm-list delete

set community

set origin

set weight

set metric-type internal

To set MED value on EBG  
updates to match the IGP metric

## Route Map Structure



```
route-map MYMAP permit 10
  AND {
    match u v w
    match x
  } OR
  AND {
    set a
    set b
  } ELSE
route-map MYMAP permit 20
  match y
  set c
```

## Using Communities



- Communities are not propagated to internal or external neighbors by default
- Therefore `send-community` keyword required

```
Router(config-router) #
neighbor ip-address send-community
neighbor ip-address route-map map-tag out
```

- The community itself is specified in a route-map:

```
Router(config-router) #
route-map map-tag permit [sequence-number]
match ip address ACL-number
set community community
```

## Policy Control: Prefix List



- Recommended method since IOS version 12.0
- Create a prefix list:

```
Router(config)#  
ip prefix-list list-name [seq seq-value] {deny|permit}  
network/len [ge ge-value] [le le-value]
```

Sequence number

Prefix and length

Range of prefix-length for more specific matches: ge-value to 32

Range of prefix-length for more specific matches: len to le-value

- Describe prefix list (for admin only)

```
Router(config)#  
ip prefix-list list-name description text
```

- Attach to BGP process:

```
Router(config-router)#  
neighbor {ip-address|peer-group-name} prefix-list  
prefix-listname {in|out}
```

## Prefix List Example



```
router bgp 5  
neighbor 172.16.1.2 remote-as 5  
neighbor 178.22.16.1 remote-as 2  
neighbor 178.22.16.1 prefix-list 1 out  
!  
ip prefix-list 1 seq 10 deny 192.68.10.0/24  
ip prefix-list 1 seq 20 permit 20.12.4.0/24  
ip prefix-list 1 seq 30 permit 25.55.2.0/24  
ip prefix-list 1 seq 40 permit 178.23.2.0/24  
ip prefix-list 1 seq 50 deny 0.0.0.0/0 le 32
```

# Route Dampening



Router(config)#

```
bgp dampening [[route-map map-name] [half-life-time  
reuse-value suppress-value maximum-suppress-time]]
```

- Minimize instability caused by route flapping and oscillation
- Cisco IOS parameters and terms:
  - **Penalty**—An incremented numeric value counting the flaps per route
  - **Half-life**—Amount of time to elapse to reduce the penalty by ½
  - **Suppress limit**—If penalty > this threshold then route is suppressed
  - **Reuse limit**—If penalty < this threshold then suppressed route will be reused
  - **History entry**—To monitor route oscillation levels, flap information is stored in a database

```
show ip bgp flap-statistics
```

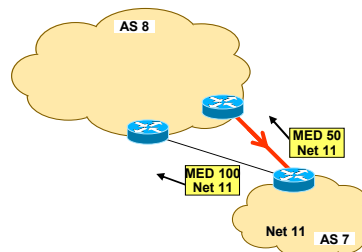
# MED Handling



- MED received by an AS does not leave this AS
- Comparing MED for the same route from different ASs is typically meaningless
- However, can be enforced by

Router(config)#

```
bgp always-compare-med
```



## MED: Cisco vs IETF



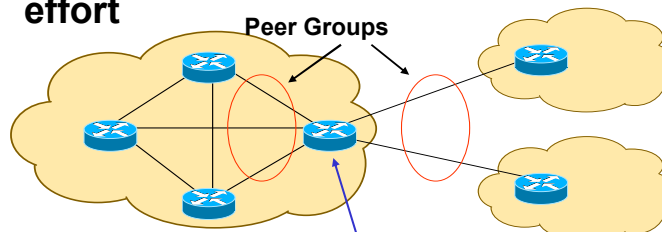
- If a route has no MED attached
  - ♦ IETF assumes MED=infinity (least preferred) per default
  - ♦ Cisco assumes a default MED of 0 (most preferred)
- To configure the router to conform to the IETF standard, use the command

```
Router(config)#  
bgp bestpath missing-as-worst
```

## Peer Groups



- Define a **template** with parameters set for group of neighbors instead of individually
  - ♦ Useful when members have **same outbound policy**
  - ♦ Members can have **different inbound policy**
- Reduced configuration and processing effort



This router has configured an internal and external peer group and applies the associated policies to them