



Configuring QoS Policy Propagation via Border Gateway Protocol

This chapter describes the tasks for configuring Policy Propagation via Border Gateway Protocol (BGP) on a router.

For complete conceptual information, see the section “QoS Policy Propagation via Border Gateway Protocol” in the chapter “Classification Overview” in this book.

For a complete description of the Policy Propagation via BGP commands in this chapter, refer to the *Cisco IOS Quality of Service Solutions Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index, or search online.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click Cancel at the login dialog box and follow the instructions that appear.

Policy Propagation via BGP Configuration Task Overview

The Policy Propagation via BGP feature allows you to classify packets by IP Precedence based on BGP community lists, BGP autonomous system paths, and access lists. After a packet has been classified, you can use other QoS features such as committed access rate (CAR) and Weighted Random Early Detection (WRED) to specify and enforce policies to fit your business model.

To configure Policy Propagation via BGP, perform the following basic tasks:

- Configure BGP and Cisco Express Forwarding (CEF) or distributed CEF (dCEF). To configure BGP, refer to the *Cisco IOS IP and IP Routing Configuration Guide*. To configure CEF and dCEF, refer to the *Cisco IOS Switching Services Configuration Guide*.
- Define the policy.
- Apply the policy through BGP.
- Configure the BGP community list, BGP autonomous system path, or access list and enable the policy on an interface. For information about this task, see the next section in this chapter.
- Enable CAR or WRED to use the policy. To enable CAR, see the chapter “Configuring Committed Access Rate” in this book. To configure WRED, see the chapter “Configuring Weighted Random Early Detection” in this book.

This chapter describes how to configure Policy Propagation based on BGP community list, BGP autonomous system path, or access list. It assumes you have already configured BGP and CEF or dCEF.

Policy Propagation via BGP Configuration Task List

To configure Policy Propagation via BGP, perform the tasks in the following sections. The first three sections are required; the remaining section is optional.

- Configuring Policy Propagation Based on Community Lists (Required)
- Configuring Policy Propagation Based on the Autonomous System Path Attribute (Required)
- Configuring Policy Propagation Based on an Access List (Required)
- Monitoring Policy Propagation via BGP (Optional)



Note

For the Policy Propagation via BGP feature to work, you must enable BGP and CEF/dCEF on the router. Subinterfaces on an ATM interface that have the **bgp-policy** command enabled must use CEF mode because dCEF is not supported. dCEF uses the Versatile Interface Processor (VIP) rather than the Route Switch Processor (RSP) to perform forwarding functions.

See the end of this chapter for the section “Policy Propagation via BGP Configuration Examples.”

Configuring Policy Propagation Based on Community Lists

This section describes how to configure Policy Propagation via BGP using community lists. The tasks listed in this section are required unless noted as optional. This section assumes you have already configured CEF/dCEF and BGP on your router.

To configure the router to propagate the IP Precedence based on the community lists, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	<code>route-map route-map-name [permit deny [sequence-number]]</code>	Defines a route map to control redistribution and enter route-map configuration mode.
Step 2	<code>match community-list community-list-number [exact]</code>	Matches a BGP community list.
Step 3	<code>set ip precedence [number name]</code>	Sets the IP Precedence field when the community list matches. You can specify either a precedence number or name.
Step 4	<code>router bgp autonomous-system</code>	Enters router configuration mode.
Step 5	<code>table-map route-map-name</code>	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 6	<code>ip community-list community-list-number {permit deny} community-number</code>	Creates a community list for BGP and controls access to it.
Step 7	<code>interface interface-type interface-number</code>	Specifies the interfaces (or subinterface) and enters interface configuration mode.
Step 8	<code>bgp-policy {source destination} ip-prec-map</code>	Classifies packets using IP Precedence.

	Command	Purpose
Step 9	<code>ip bgp-community new-format</code>	(Optional) Configures a new community format so that the community number is displayed in the short form.
Step 10	<code>end</code>	Exits configuration mode.

Configuring Policy Propagation Based on the Autonomous System Path Attribute

This section describes how to configure Policy Propagation via BGP based on the autonomous system path. This section assumes you have already configured CEF/dCEF and BGP on your router.

To configure the router to propagate the IP Precedence based on the autonomous system path attribute, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	<code>route-map route-map-name [permit deny [sequence-number]]</code>	Defines a route map to control redistribution and enter route-map configuration mode.
Step 2	<code>match as-path path-list-number</code>	Matches a BGP autonomous system path access list.
Step 3	<code>set ip precedence [number name]</code>	Sets the IP Precedence field when the autonomous system path matches. Specifies either a precedence number or name.
Step 4	<code>router bgp autonomous-system</code>	Enters router configuration mode.
Step 5	<code>table-map route-map-name</code>	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 6	<code>ip as-path access-list access-list-number {permit deny} as-regular-expression</code>	Defines an autonomous system path access list.
Step 7	<code>interface interface-type interface-number</code>	Specifies the interfaces (or subinterface) and enters interface configuration mode.
Step 8	<code>bgp-policy {source destination} ip-prec-map</code>	Classifies packets using IP Precedence.
Step 9	<code>end</code>	Exits configuration mode.

Configuring Policy Propagation Based on an Access List

This section describes how to configure Policy Propagation via BGP based on an access list. This section assumes you have already configured CEF/dCEF and BGP on your router.

To configure the router to propagate the IP Precedence based on an access list, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	<code>route-map route-map-name [permit deny [sequence-number]]</code>	Defines a route map to control redistribution and enter route-map configuration mode.
Step 2	<code>match ip address access-list-number</code>	Matches an access list.
Step 3	<code>set ip precedence [number name]</code>	Sets the IP Precedence field when the autonomous system path matches.
Step 4	<code>router bgp autonomous-system</code>	Enters router configuration mode.
Step 5	<code>table-map route-map-name</code>	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 6	<code>access-list access-list-number {permit deny} source</code>	Defines an access list.
Step 7	<code>interface interface-type interface-number</code>	Specifies the interfaces (or subinterface) and enters interface configuration mode.
Step 8	<code>bgp-policy {source destination} ip-prec-map</code>	Classifies packets using IP Precedence.
Step 9	<code>end</code>	Exits configuration mode.

Monitoring Policy Propagation via BGP

To monitor the Policy Propagation via BGP configuration, use one or more of the following commands in EXEC mode. The commands listed in this section are optional.

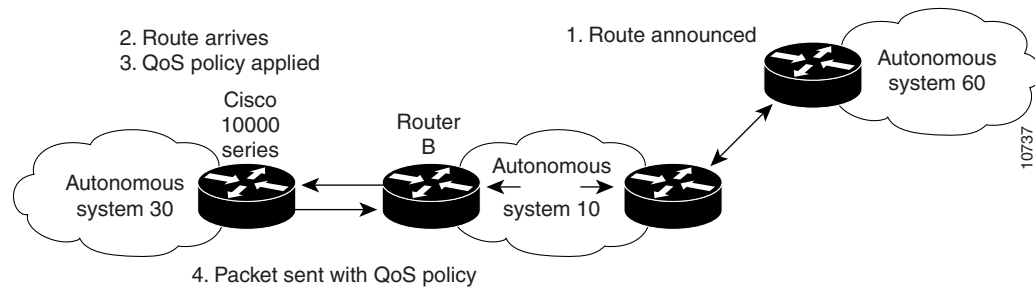
Command	Purpose
<code>show ip bgp</code>	Shows entries in the BGP routing table, to verify the correct community is set on the prefixes.
<code>show ip bgp community-list community-list-number</code>	Shows routes permitted by the BGP community list, to verify that the correct prefixes are selected.
<code>show ip cef network</code>	Shows entries in the Forwarding Information Base (FIB) table based on the IP address, to verify that CEF has the correct precedence value for the prefix.
<code>show ip interface</code>	Shows information about the interface.
<code>show ip route prefix</code>	Shows the current status of the routing table, to verify that the correct precedence values are set on the prefixes.

Policy Propagation via BGP Configuration Examples

The following example shows how to create route maps to match access lists, BGP community lists, and BGP autonomous system paths, and apply IP Precedence to routes learned from neighbors. For information on how to configure Policy Propagation via BGP, see the section “Policy Propagation via BGP Configuration Task Overview” in this chapter.

In Figure 2, Router A (Cisco10000 Series) learns routes from autonomous system 10 and autonomous system 60. QoS policy is applied to all packets that match the defined route maps. Any packets from Router A (Cisco 10000 Series) to autonomous system 10 or autonomous system 60 are sent the appropriate QoS policy.

Figure 2 Router Learns Routes and Applies QoS Policy



Router A (Cisco 10000 Series) Configuration

```
interface serial 5/0/0/1:0
ip address 200.28.38.2 255.255.255.0
bgp-policy destination ip-prec-map
no ip mroute-cache
no cdp enable
frame-relay interface-dlci 20 IETF

router bgp 30
 table-map precedence-map
 neighbor 20.20.20.1 remote-as 10
 neighbor 20.20.20.1 send-community
 !
ip bgp-community new-format
 !
 ! Match community 1 and set the IP Precedence to priority
route-map precedence-map permit 10
 match community 1
 set ip precedence priority
 !
 ! Match community 2 and set the IP Precedence to immediate
route-map precedence-map permit 20
 match community 2
 set ip precedence immediate
 !
 ! Match community 3 and set the IP Precedence to flash
route-map precedence-map permit 30
 match community 3
 set ip precedence flash
 !
 ! Match community 4 and set the IP Precedence to flash-override
route-map precedence-map permit 40
 match community 4
```

```

    set ip precedence flash-override
    !
    ! Match community 5 and set the IP Precedence to critical
    route-map precedence-map permit 50
    match community 5
    set ip precedence critical
    !
    ! Match community 6 and set the IP Precedence to internet
    route-map precedence-map permit 60
    match community 6
    set ip precedence internet
    !
    ! Match community 7 and set the IP Precedence to network
    route-map precedence-map permit 70
    match community 7
    set ip precedence network
    !
    ! Match ip address access list 69 or match AS path 1
    ! and set the IP Precedence to critical
    route-map precedence-map permit 75
    match ip address 69
    match as-path 1
    set ip precedence critical
    !
    ! For everything else, set the IP Precedence to routine
    route-map precedence-map permit 80
    set ip precedence routine
    !
    ! Define the community lists
    ip community-list 1 permit 60:1
    ip community-list 2 permit 60:2
    ip community-list 3 permit 60:3
    ip community-list 4 permit 60:4
    ip community-list 5 permit 60:5
    ip community-list 6 permit 60:6
    ip community-list 7 permit 60:7
    !
    ! Define the AS path
    ip as-path access-list 1 permit ^10_60
    !
    ! Define the access list
    access-list 69 permit 69.0.0.0

```

Router B Configuration

```

router bgp 10
 neighbor 30.30.30.1 remote-as 30
 neighbor 30.30.30.1 send-community
 neighbor 30.30.30.1 route-map send_community out
 !
 ip bgp-community new-format
 !
 ! Match prefix 10 and set community to 60:1
 route-map send_community permit 10
 match ip address 10
 set community 60:1
 !
 ! Match prefix 20 and set community to 60:2
 route-map send_community permit 20
 match ip address 20
 set community 60:2
 !
 ! Match prefix 30 and set community to 60:3
 route-map send_community permit 30

```

```
match ip address 30
set community 60:3
!
! Match prefix 40 and set community to 60:4
route-map send_community permit 40
match ip address 40
set community 60:4
!
! Match prefix 50 and set community to 60:5
route-map send_community permit 50
match ip address 50
set community 60:5
!
! Match prefix 60 and set community to 60:6
route-map send_community permit 60
match ip address 60
set community 60:6
!
! Match prefix 70 and set community to 60:7
route-map send_community permit 70
match ip address 70
set community 60:7
!
! For all others, set community to 60:8
route-map send_community permit 80
set community 60:8
!
! Define the access lists
access-list 10 permit 61.0.0.0
access-list 20 permit 62.0.0.0
access-list 30 permit 63.0.0.0
access-list 40 permit 64.0.0.0
access-list 50 permit 65.0.0.0
access-list 60 permit 66.0.0.0
access-list 70 permit 67.0.0.0
```

