

Table of Contents

<u>Using BGP Community Values to Control Routing Policy in an Upstream Provider Network</u>	1
<u>Introduction</u>	1
<u>Before You Begin</u>	1
<u>Conventions</u>	1
<u>Prerequisites</u>	1
<u>Components Used</u>	1
<u>Background Theory</u>	1
<u>Configure</u>	2
<u>Controlling the Routing Policy</u>	2
<u>Network Diagram</u>	3
<u>Configurations</u>	3
<u>Verify</u>	8
<u>Related Information</u>	13

Using BGP Community Values to Control Routing Policy in an Upstream Provider Network

Introduction

Before You Begin

- Conventions
- Prerequisites
- Components Used
- Background Theory

Configure

- Controlling the Routing Policy
- Network Diagram
- Configurations

Verify

Related Information

Introduction

This document demonstrates how the Border Gateway Protocol (BGP) community attribute can be used to control the routing policy in its upstream service provider network.

Before You Begin

Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

Prerequisites

There are no specific prerequisites for this document.

Components Used

This document is not restricted to specific software and hardware versions.

Background Theory

While communities themselves do not alter the BGP decision making process, they can be used as flags to mark a set of routes. Upstream service provider routers may then use these flags to apply specific routing policies (for example, local preference) within their network.

Providers establish a mapping between customer configurable community values and the corresponding local preference values within the providers network. The idea is that customers with specific policies that require the modification of LOCAL_PREF in the provider network sets the corresponding community values in their routing updates.

A community is a group of prefixes that share some common property and can be configured using BGP community attribute. The BGP community attribute is an optional transitive attribute of variable length. The

attribute consists of a set of four octet values, specifying a community. The community attribute values are encoded using an Autonomous System (AS) number in the first two octets, with the remaining two octets defined by the AS. A prefix can have more than one community attribute. A BGP speaker that sees multiple community attributes in a prefix can act based on one, some or all the attributes. A router has the option of adding or modifying community attribute before passing on to other peers. To learn more about the community attribute, refer to BGP Case Studies.

The local preference attribute is an indication to the AS which path is preferred in order to reach a certain network. When there are multiple paths to the same destination, the path with the higher preference is preferred (the default value of the local preference attribute is 100). For more information refer to Local Preference Attribute.

Configure

Controlling the Routing Policy

In this section, you are presented with the information to configure the features described in this document.

Note: To find additional information on the commands used in this document, use the Command Lookup Tool (registered customers only).

For simplification, the following mapping of the community attribute and local preference attribute is assumed to be established between the upstream service provider (AS 100) and the customer (AS 30).

Local Preference	Community Values
130	100:300
125	100:250

If the customer announces the prefixes with a community attribute equal to 100:300, then the upstream service provider sets the local preference of those routes to 130 and 125 if the community attribute equals 100:250.

This gives the potential for you to control the routing policy within the service provider network by changing the community values of the prefixes announced to the service provider.

In the network diagram below, customer AS 30 wishes to achieve the following routing policy using the community attributes.

- The traffic inbound from AS 100 destined to network 6.6.6.0/24 should come through the R1–R3 link. In case the R1–R3 link fails, all traffic should come in through R2–R3.
- The traffic inbound from AS 100 destined to network 7.7.7.0/24 should come through the R2–R3 link. In case the R2–R3 link fails, all traffic should come in through R1–R3.

To achieve this routing policy, R3 announces its prefixes as follows:

To R1:

- 6.6.6.0/24 with a community attribute 100:300
- 7.7.7.0/24 with a community attribute 100:250

To R2:

- 6.6.6.0/24 with a community attribute 100:250
- 7.7.7.0/24 with a community attribute 100:300

Once BGP neighbors R1 and R2 receive the prefixes from R3, they apply the preconfigured policy based on mapping between the community and local preference attributes (shown in the table above), and thus achieving the routing policy dictated by customer (AS 30). R1 installs the prefixes in the BGP table.

- 6.6.6.0/24 with a local preference of 130
- 7.7.7.0/24 with a local preference of 125

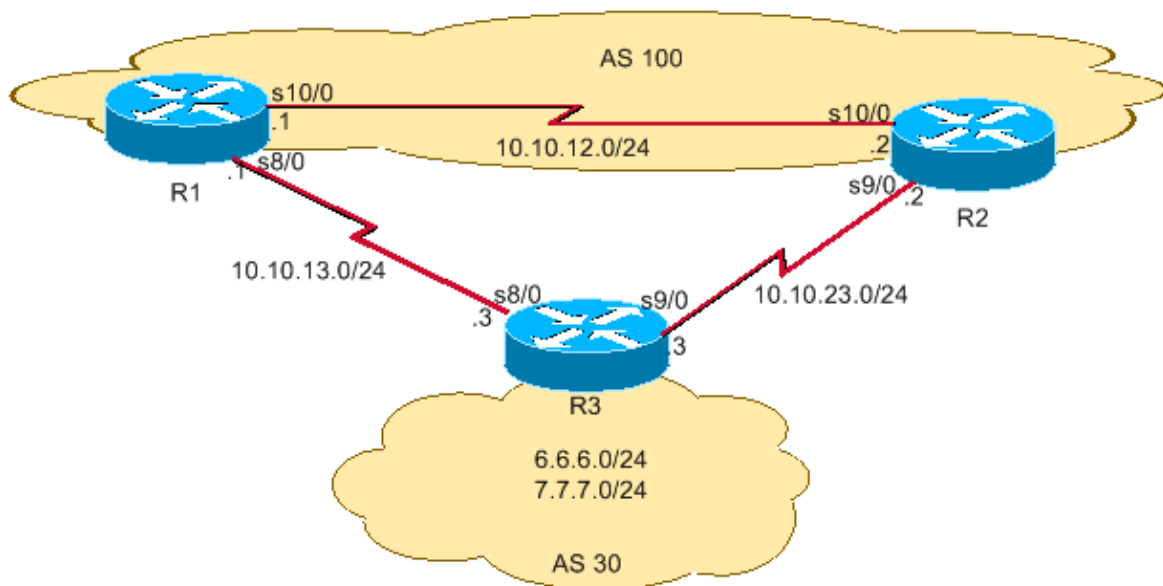
R2 installs the prefix in its BGP table:

- 6.6.6.0/24 with a local preference of 125
- 7.7.7.0/24 with a local preference of 130

Since a higher local preference is preferred in the BGP path selection criteria, the path with a local preference of 130 (130 is greater than 125) is selected as the best path within AS 100, and is installed in the IP routing table of R1 and R2. For more information on BGP path selection criteria, see BGP Best Path Selection Algorithm for more information.

Network Diagram

This document uses the network setup shown in the diagram below.



Configurations

This document uses the configurations shown below.

- R3
- R1
- R2

R3

```

Current configuration : 2037 bytes
!
version 12.2
!
hostname R3
!
interface Loopback0
 ip address 6.6.6.1 255.255.255.0
!
interface Ethernet0/0
 ip address 7.7.7.1 255.255.255.0
!
interface Serial8/0
 ip address 10.10.13.3 255.255.255.0

!--- Interface connected to R1.

!
interface Serial9/0
 ip address 10.10.23.3 255.255.255.0

!--- Interface connected to R2.

!
router bgp 30
 network 6.6.6.0 mask 255.255.255.0
 network 7.7.7.0 mask 255.255.255.0

!--- Network commands announce prefix 6.6.6.0/24

!--- and 7.7.7.0/24.

 neighbor 10.10.13.1 remote-as 100

!--- Establishes peering with R1.

 neighbor 10.10.13.1 send-community

!--- Configures to send community attribute to R1.

 neighbor 10.10.13.1 route-map Peer-R1 out

!--- Configures outbound policy as defined by

!--- route-map "Peer-R1" when peering with R1.

 neighbor 10.10.23.2 remote-as 100

!--- Establishes peering with R2.

 neighbor 10.10.23.2 send-community

!--- Configures to send community attribute to R2.

 neighbor 10.10.23.2 route-map Peer-R2 out

!--- Configures outbound policy as defined by

!--- route-map "Peer-R2" when peering with R2.

no auto-summary

```

```

!
ip classless
ip bgp-community new-format

!--- Allows you to configure the BGP community

!--- attribute in AA:NN format.

!
access-list 101 permit ip host 6.6.6.0 host 255.255.255.0
access-list 102 permit ip host 7.7.7.0 host 255.255.255.0
!
!
route-map Peer-R1 permit 10
match ip address 101
set community 100:300

!--- Sets community 100:300 for routes matching access-list 101.

!
route-map Peer-R1 permit 20
match ip address 102
set community 100:250

!--- Sets community 100:250 for routes matching access-list 102.

!
route-map Peer-R2 permit 10
match ip address 101
set community 100:250

!--- Sets community 100:250 for routes matching access-list 101.

!
route-map Peer-R2 permit 20
match ip address 102
set community 100:300

!--- Sets community 100:300 for routes matching access-list 102.

!
end

```

R1

```

Version 12.2
!
hostname R1
!
interface Loopback0
ip address 200.200.200.1 255.255.255.0
!
interface Serial8/0
ip address 10.10.13.1 255.255.255.0

!--- Connected to R3.

!
interface Serial10/0
ip address 10.10.12.1 255.255.255.0

!--- Connected to R2.

```

```

!
router bgp 100
  no synchronization
  bgp log-neighbor-changes
  neighbor 10.10.12.2 remote-as 100

  !--- Establishes peering with R2.

  neighbor 10.10.12.2 next-hop-self
  neighbor 10.10.13.3 remote-as 30

  !--- Establishes peering with R3.

  neighbor 10.10.13.3 route-map Peer-R3 in

  !--- Configures the inbound policy as defined by

  !--- route-map "Peer-R3" when peering with R3.

  no auto-summary
!
ip bgp-community new-format

!--- Allows you to configure the BGP community

!--- attribute in AA:NN format.

ip community-list 1 permit 100:300
ip community-list 2 permit 100:250

!--- Defines community list 1 and 2.

!
route-map Peer-R3 permit 10
  match community 1
  set local-preference 130

!--- Sets local preference 130 for all routes

!--- matching community list 1.

!
route-map Peer-R3 permit 20
  match community 2
  set local-preference 125

!--- Sets local preference 125 for all routes

!--- matching community list 2.

!
route-map Peer-R3 permit 30

!--- This line allows all the BGP routes

!--- learned from R3, which did not match the

```

```
!--- route-map Peer-R3 permit 10 and

!--- 20 statements, to be installed into its BGP table.

!
end
```

R2

```
Version 12.2
!
hostname R2
!
interface Loopback0
ip address 192.168.50.1 255.255.255.0
!
interface Serial9/0
ip address 10.10.23.2 255.255.255.0

!--- Connected to R3.

!
interface Serial10/0
ip address 10.10.12.2 255.255.255.0

!--- Connected to R1.

!
router bgp 100
no synchronization
bgp log-neighbor-changes
neighbor 10.10.12.1 remote-as 100

!--- Establishes iBGP peering with R1.

neighbor 10.10.12.1 next-hop-self
neighbor 10.10.23.3 remote-as 30

!--- Establishes peering with R3.

neighbor 10.10.23.3 route-map Peer-R3 in

!--- Configures inbound policy as defined by

!--- route-map "Peer-R3" when peering with R3.

no auto-summary
!
ip bgp-community new-format

!--- Allows you to configure the BGP community

!--- attribute in AA:NN format.

!
ip community-list 1 permit 100:300
ip community-list 2 permit 100:250

!--- Defines community list 1 and 2.
```



```

!
route-map Peer-R3 permit 10
  match community 1
  set local-preference 130

!--- Sets local preference 130 for all routes

!--- matching community list 1.

!
route-map Peer-R3 permit 20
  match community 2
  set local-preference 125

!--- Sets local preference 125 for all routes

!--- matching community list 2.

!
route-map Peer-R3 permit 30

!--- This line allows all the BGP routes learned

!--- from R3 which did not match the route-map

!--- Peer-R3 permit 10 and 20 statements, to be

!--- installed into its BGP table.

!
end

```

Verify

R1 receives prefixes 6.6.6.0/24 and 7.7.7.0/24 with communities 100:300 and 100:250, as shown in bold below in the **show ip bgp** output.

Note: Once these routes are installed into the BGP table based on the configured policy, prefixes with community 100:300 are assigned local preference 130 and prefixes with community 100:250 are assigned local preference 125.

```

R1# show ip bgp 6.6.6.0
BGP routing table entry for 6.6.6.0/24, version 2
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    10.10.12.2
    30
    10.10.13.3 from 10.10.13.3 (6.6.6.1)
      Origin IGP, metric 0, localpref 130, valid, external, best
      Community: 100:300

!--- Prefix 6.6.6.0/24 with community 100:300 received from

!--- 10.10.13.3 (R3) is assigned local preference 130.

```

```

R1# show ip bgp 7.7.7.0
BGP routing table entry for 7.7.7.0/24, version 4
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    10.10.13.3
    30
      10.10.12.2 from 10.10.12.2 (192.168.50.1)
        Origin IGP, metric 0, localpref 130, valid, internal, best

!--- Received prefix 7.7.7.0/24 over iBGP from 10.10.12.2

!--- (R2) with local preference 130.

30
  10.10.13.3 from 10.10.13.3 (6.6.6.1)
    Origin IGP, metric 0, localpref 125, valid, external
    Community: 100:250

!--- Prefix 7.7.7.0/24 with community 100:250 received from

!--- 10.10.13.3 (R3) is assigned local preference 125.

```

```

R1# show ip bgp
BGP table version is 4, local router ID is 200.200.200.1
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
*> 6.6.6.0/24       10.10.13.3         0      130     0 30 i
*>i7.7.7.0/24       10.10.12.2         0      130     0 30 i
*                   10.10.13.3         0      125     0 30 i

```

The show ip bgp on R1 confirms that the best path selected on R1 are with local preference(LocPrf) = 130.

Similarly, R2 receives prefixes 6.6.6.0/24 and 7.7.7.0/24 with communities 100:250 and 100:300, as shown in bold below in **show ip bgp** output.

Note: Once these routes are installed into the BGP table, based on the configured policy, prefixes with community 100:300 are assigned local preference 130 and prefixes with community 100:250 are assigned local preference 125.

```

R2# show ip bgp 6.6.6.0
BGP routing table entry for 6.6.6.0/24, version 2
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    10.10.23.3
    30
      10.10.23.3 from 10.10.23.3 (6.6.6.1)
        Origin IGP, metric 0, localpref 125, valid, external
        Community: 100:250

!--- Prefix 6.6.6.0/24 with community 100:250 received from

!--- 10.10.23.3 (R3) is assigned local preference 125.

```

```

30
 10.10.12.1 from 10.10.12.1 (200.200.200.1)
   Origin IGP, metric 0, localpref 130, valid, internal, best

!--- Received prefix 6.6.6.0/24 over iBGP from 10.10.12.1

!--- (R1) with local preference 130.

R2# show ip bgp 7.7.7.0
BGP routing table entry for 7.7.7.0/24, version 3
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    10.10.12.1
    30
    10.10.23.3 from 10.10.23.3 (6.6.6.1)
      Origin IGP, metric 0, localpref 130, valid, external, best
      Community: 100:300

!--- Prefix 7.7.7.0/24 with community 100:300 received from

!--- 10.10.23.3 (R3) is assigned local preference 130.

```

```

R2# show ip bgp
BGP table version is 3, local router ID is 192.168.50.1
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
*  6.6.6.0/24       10.10.23.3        0     125    0 30 i
*>i                 10.10.12.1        0     130    0 30 i
*> 7.7.7.0/24       10.10.23.3        0     130    0 30 i

```

The show ip bgp output above on R2 confirms the best path selected on R2 are with local preference(localPrf) = 130.

The IP route to prefix 6.6.6.0/24 should prefer the R1–R3 link exiting out of AS 100 towards AS 30. The **show ip route** on R1 and R2 confirms that.

```

R1# show ip route 6.6.6.0
Routing entry for 6.6.6.0/24
  Known via "bgp 100", distance 20, metric 0
  Tag 30, type external
  Last update from 10.10.13.3 3d21h ago
  Routing Descriptor Blocks:
    * 10.10.13.3, from 10.10.13.3, 3d21h ago
      Route metric is 0, traffic share count is 1
      AS Hops 1

!--- On R1, the IP route to prefix 6.6.6.0/24 points

!--- to next hop 10.10.13.3 which is R3 serial 8/0

!--- interface on the R1-R3 link.

```

```
R2# show ip route 6.6.6.0
Routing entry for 6.6.6.0/24
  Known via "bgp 100", distance 200, metric 0
  Tag 30, type internal
  Last update from 10.10.12.1 3d21h ago
  Routing Descriptor Blocks:
  * 10.10.12.1, from 10.10.12.1, 3d21h ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
```

!--- On R2, IP route to prefix 6.6.6.0/24 points

!--- to next hop R1 (10.10.12.1) on its iBGP link.

!--- Thus traffic to network 6.6.6.0/24 from R2 will

!--- exit through R2-R1 and then R1-R3 link from

!--- AS 100 towards AS 30.

The IP route to prefix 7.7.7.0/24 should prefer R2-R3 link exiting out of AS 100 towards AS 30. The **show ip route** command on R1 and R2 confirms that.

```
R2# show ip route 7.7.7.0
Routing entry for 7.7.7.0/24
  Known via "bgp 100", distance 20, metric 0
  Tag 30, type external
  Last update from 10.10.23.3 3d22h ago
  Routing Descriptor Blocks:
  * 10.10.23.3, from 10.10.23.3, 3d22h ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
```

!--- On R2, IP route to prefix 7.7.7.0/24 points

!--- to next hop 10.10.23.3 which is R3 serial 9/0

!--- interface on R2-R3 link.

```
R1# show ip route 7.7.7.0
Routing entry for 7.7.7.0/24
  Known via "bgp 100", distance 200, metric 0
  Tag 30, type internal
  Last update from 10.10.12.2 3d22h ago
  Routing Descriptor Blocks:
  * 10.10.12.2, from 10.10.12.2, 3d22h ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
```

!--- On R1, IP route to prefix 7.7.7.0/24 points

!--- to next hop R1 (10.10.12.2) on its iBGP link.

!--- Thus traffic to network 7.7.7.0/24 from R1

!--- will exit through R1-R2 and then R2-R3 link

!--- from AS 100 towards AS 30.

In the event of failure of one link, for example the R1–R3 link, all traffic should follow the R2–R3 link. Simulate that by shutting down the link between R1–R3.

```
R1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int s8/0
R1(config-if)#shut
R1(config-if)#
3d22h: %BGP-5-ADJCHANGE: neighbor 10.10.13.3 Down Interface flap
3d22h: %LINK-5-CHANGED: Interface Serial8/0, changed state to
administratively down
3d22h: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial8/0,
changed state to down
```

Notice the IP routing table for prefix 6.6.6.0/24 and 7.7.7.0/24 on R1 and R2. Use R2–R3 link to exit out of AS 100.

```
R1# show ip route 6.6.6.0
Routing entry for 6.6.6.0/24
  Known via "bgp 100", distance 200, metric 0
  Tag 30, type internal
  Last update from 10.10.12.2 00:01:47 ago
  Routing Descriptor Blocks:
  * 10.10.12.2, from 10.10.12.2, 00:01:47 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
```

```
R1# show ip route 7.7.7.0
Routing entry for 7.7.7.0/24
  Known via "bgp 100", distance 200, metric 0
  Tag 30, type internal
  Last update from 10.10.12.2 3d22h ago
  Routing Descriptor Blocks:
  * 10.10.12.2, from 10.10.12.2, 3d22h ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
```

The **show** output above shows that the route to prefixes 6.6.6.0/24 and 7.7.7.0/24 points to the next hop 10.10.12.2, (R2), which is expected. Now take a look at the IP routing table on R2 to check next-hop of prefix 6.6.6.0/24 and 7.7.7.0/24. The next hop should be R3 for the configured policy to work successfully.

```
R2# show ip route 6.6.6.0
Routing entry for 6.6.6.0/24
  Known via "bgp 100", distance 20, metric 0
  Tag 30, type external
  Last update from 10.10.23.3 00:04:10 ago
  Routing Descriptor Blocks:
  * 10.10.23.3, from 10.10.23.3, 00:04:10 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
```

```
R2# show ip route 7.7.7.0
Routing entry for 7.7.7.0/24
  Known via "bgp 100", distance 20, metric 0
  Tag 30, type external
  Last update from 10.10.23.3 3d22h ago
  Routing Descriptor Blocks:
    * 10.10.23.3, from 10.10.23.3, 3d22h ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
```

The next hop 10.10.23.3 is R3 Serial 9/0 interface on the R2–R3 link. This confirms the configured policy is working as expected.

Related Information

- [RFC 1998](#)
 - [Troubleshooting BGP](#)
 - [BGP: Frequently Asked Questions](#)
 - [Sample Configurations for Load Sharing with BGP in Single and Multihomed Environments](#)
 - [Achieve Optimal Routing and Reduce BGP Memory Consumption](#)
 - [BGP Technical Support](#)
 - [Technical Support – Cisco Systems](#)
-

All contents are Copyright © 1992–2003 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.