



BGP DMZ Bandwidth Management

This chapter describes the various aspects of BGP Demilitarized Zone (DMZ) bandwidth management, including aggregate bandwidth, link bandwidth for unequal cost recursive load balancing, and transitive-bandwidth extended community support.

- [BGP DMZ aggregate bandwidth, on page 1](#)
- [Removal of link-bandwidth extended community to iBGP peers, on page 3](#)
- [BGP DMZ link bandwidth for unequal cost recursive load balancing, on page 5](#)
- [BGP DMZ link bandwidth enhancement, on page 11](#)
- [BGP DMZ transitive-bandwidth extended community support, on page 14](#)

BGP DMZ aggregate bandwidth

BGP DMZ aggregate bandwidth is a feature that aggregates the link-bandwidth values of DMZ eBGP multipaths when advertising routes to iBGP peers, and enables accurate internal bandwidth representation for better routing decisions.

BGP DMZ aggregate bandwidth operation

BGP aggregates bandwidth without an explicit command if these conditions are met:

- The network has multipaths and all multipaths have link-bandwidth values.
- You set the *next-hop* attribute to *next-hop-self*. The *next-hop* attribute for all routes advertised to the specified neighbor is the address of the local router.
- You do not configure an outbound policy that might change the DMZ link-bandwidth value.

DMZ link bandwidth aggregation rules

DMZ link bandwidth aggregation follows these rules:

- If BGP does not know the DMZ link-bandwidth value (*dmz-link-bandwidth*) for any one of the multipaths (eBGP or iBGP), BGP does not download the DMZ link-bandwidth value for all multipaths, including the best path, to the routing information base (RIB).
- BGP does not consider the DMZ link-bandwidth value of iBGP multipath during aggregation.
- BGP can advertise the route with an aggregate value as a best path or an add-path.

- Add-path does not qualify for DMZ link bandwidth aggregation as the next hop is preserved. BGP does not support configuring next-hop-self for add-path.
- For VPNv4 and VPNv6 address family identifiers (AFIs), if you configure the DMZ link-bandwidth value using an outbound route-policy, specify the route table or use the *additive* keyword. Otherwise, the system does not import routes on the receiving end of the peer.

Configure BGP DMZ aggregate bandwidth

Configure BGP DMZ aggregate bandwidth in a sample topology.

This example uses a topology of R1---(iBGP)---R2---(iBGP)---R3 to demonstrate how aggregated DMZ link-bandwidth values are sent between routers. The routers in the topology advertise and receive aggregated DMZ link-bandwidth values.

- On R1, BGP prefix has:
 - path 1 (bestpath) with link-bandwidth value 100
 - path 2 (eBGP multipath) with link-bandwidth value 30, and
 - path 3 (eBGP multipath) with link-bandwidth value 50.

When the best path is advertised to R2, R1 sends an aggregated DMZ link-bandwidth value of 180; this is the aggregated value of paths 1, 2, and 3.

- On R2, BGP prefix has:
 - path 1 (bestpath) with link-bandwidth value 60
 - path 2 (eBGP multipath) with link-bandwidth value 200, and
 - path 3 (eBGP multipath) with link-bandwidth value 50.

When the best path is advertised to R3, R2 sends an aggregated DMZ link-bandwidth value of 310; this is the aggregated value of paths 1, 2, and 3.

- On R3, BGP prefix has:
 - path 1 (bestpath) with LB 180 (learned from R1)
 - path 2 (iBGP multipath) with LB 310 (learned from R2)

This sample configuration demonstrates how to set the link-bandwidth extended community on a per-path basis at either the neighbor-in or neighbor-out policy attach points. The **dmz-link-bandwidth** command is configured under eBGP neighbor configuration mode. All paths received from that particular neighbor are marked with the link-bandwidth extended community when sent to iBGP peers.

Procedure

Step 1 Configure an inbound or outbound route-policy.

Example:

```

Router(config)# extcommunity-set bandwidth dmz_ext
Router(config-ext)# 1:1290400000
Router(config-ext)# end-set
Router(config)#route-policy dmz_rp
Router(config-rpl)#set extcommunity bandwidth dmz_ext
Router(config-rpl)#pass
Router(config-rpl)#end-policy
Router(config)#router bgp 65000
Router(config-bgp)#neighbor 10.0.101.1
Router(config-bgp-nbr)#remote-as 1001
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#route-policy dmz_rp in
Router(config-bgp-nbr-af)#route-policy pass out
Router(config-bgp-nbr-af)#commit

```

Step 2 Configure the **dmz-link-bandwidth** command for the BGP neighbor.

Example:

```

Router(config)#router bgp 65000
Router(config-bgp)#neighbor 10.0.101.2
Router(config-bgp-nbr)#remote-as 1001
Router(config-bgp-nbr)#dmz-link-bandwidth
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#route-policy pass in
Router(config-bgp-nbr-af)#route-policy pass out
Router(config-bgp-nbr-af)#commit

```

The system applies policy-based link bandwidth settings to BGP neighbors.

Removal of link-bandwidth extended community to iBGP peers

The removal of link-bandwidth extended community to iBGP peers is a feature that:

- minimizes the risk of exposing DMZ link-bandwidth community parameters to network zones where they are not recognized or necessary
- allows BGP to send traffic over multiple internal BGP (iBGP) learned paths, and
- sends traffic proportional to the bandwidth of the links used to exit the autonomous system.

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Removal of Link-Bandwidth Extended Community to iBGP Peers	Release 7.3.2	The demilitarized zone (DMZ) link-bandwidth extended community allows BGP to send traffic over multiple internal BGP (iBGP) learned paths. The traffic that is sent is proportional to the bandwidth of the links that are used to exit the autonomous system. By default, iBGP propagates DMZ link-bandwidth community. This feature minimizes the risk of exposure of the community parameters, which are used to control the routing policy in the service provider network, to network zones where they are not recognized or not required.

By default, iBGP propagates DMZ link-bandwidth community. This feature minimizes the risk of exposing community parameters, which control the routing policy in the service provider network, to network zones where they are not recognized or required.

Configure route policy to remove extended communities

Configure a route policy to remove extended communities.

This task provides examples of how to configure route policies to remove extended communities, either all of them or specific ones.

Procedure

Step 1 Choose one of these steps to configure a route policy to remove extended communities

- Configure a route policy to delete all extended communities.

```
Router(config)# route-policy dmz_del_all
Router(config-rpl)# delete extcommunity bandwidth all
Router(config-rpl)# pass
Router(config-rpl)# end-policy
```

- Configure a route policy to delete only the extended communities that match an extended community mentioned in a list.

```
Router(config)# route-policy dmz_CE1_del_non_match
Router(config-rpl)# if destination in (10.9.9.9/32) then
Router(config-rpl-if)# delete extcommunity bandwidth in (10:7000)
Router(config-rpl-if)# endif
Router(config-rpl)# pass
Router(config-rpl)# end-policy
```

- Configure a route policy to delete all extended communities using parameters.

```
Router(config)# route-policy dmz_del_param2($a,$b)
Router(config-rpl)# if destination in (10.9.9.9/32) then
Router(config-rpl-if)# delete extcommunity bandwidth in ($a:$b)
Router(config-rpl-if)# endif
Router(config-rpl)# pass
Router(config-rpl)# end-policy
```

Step 2 Run **show bgp** command to confirm the BGP routing table entries and extended community attributes.

Example:

```
Router# show bgp 10.9.9.9/32
```

```
BGP routing table entry for 10.9.9.9/32
Versions:
Process bRIB/RIB SendTblVer
Speaker 15 15
Last Modified: Aug 27 13:06:45.000 for 00:08:21
Paths: (3 available, best #1)
Advertised IPv4 Unicast paths to peers (in unique update groups):
13.13.13.5
Path #1: Received by speaker 0
Advertised IPv4 Unicast paths to peers (in unique update groups):
13.13.13.5
10
```

```

10.10.10.1 from 10.10.10.1 (192.168.0.1)
Origin incomplete, metric 0, localpref 100, valid, external, best, group-best, multipath
Received Path ID 0, Local Path ID 1, version 15
Extended community: LB:10:48
Origin-AS validity: (disabled)
Path #2: Received by speaker 0
Not advertised to any peer
10
11.11.11.3 from 11.11.11.3 (192.168.0.3)
Origin incomplete, metric 0, localpref 100, valid, external, multipath
Received Path ID 0, Local Path ID 0, version 0
Extended community: LB:10:48
Origin-AS validity: (disabled)
Path #3: Received by speaker 0
Not advertised to any peer
10
12.12.12.4 from 12.12.12.4 (192.168.0.4)
Origin incomplete, metric 0, localpref 100, valid, external, multipath
Received Path ID 0, Local Path ID 0, version 0
Extended community: LB:10:48
Origin-AS validity: (disabled)

22:35 30-09-2021

```

BGP DMZ link bandwidth for unequal cost recursive load balancing

BGP DMZ link bandwidth for unequal cost recursive load balancing is a feature that provides support for unequal cost load balancing for recursive prefixes on a local node using BGP DMZ link bandwidth. The system achieves unequal load balancing by using the **dmz-link-bandwidth** command in BGP neighbor configuration mode and the **bandwidth** command in interface configuration mode.

Enable BGP unequal cost recursive load balancing

Enable BGP unequal cost recursive load balancing.

Before you begin

Ensure the network configuration supports BGP and interface bandwidth settings.

Procedure

Step 1 Enter global configuration mode to begin making configuration changes.

Example:

```
Router# config
```

Step 2 Enter BGP router configuration mode and specify the local Autonomous System Number (ASN) for your router.

Example:

```
Router(config)# router bgp 120
```

Step 3 Specify the BGP address family, IPv4 or IPv6 unicast, to configure routing for the desired protocol.

Example:

```
Router(config-bgp) # address-family ipv4 unicast
```

Step 4 Set the maximum number of parallel BGP paths, for eBGP, iBGP, or both, that can be used for load balancing, and optionally enable unequal-cost multipath.

Example:

```
Router(config-bgp-af) # maximum-paths ebgp 3
```

Step 5 Exit the address-family configuration mode and return to BGP router configuration mode.

Example:

```
Router(config-bgp-af) # exit
```

Step 6 Select the BGP neighbor you want to configure by specifying its IP address.

Example:

```
Router(config-bgp) # neighbor 10.0.0.0
```

Step 7 Enable the DMZ link-bandwidth feature for the specified BGP neighbor, allowing the router to use per-link bandwidth information for load balancing.

Example:

```
Router(config-bgp-nbr) # dmz-link-bandwidth
Router(config-bgp-nbr) # commit
```

Configure BGP unequal cost recursive load balancing: example

Provide a sample configuration for unequal cost recursive load balancing.

This example shows the configuration for various interfaces and BGP settings to achieve unequal cost recursive load balancing.

Procedure

Step 1 Execute these commands in configuration mode to set up BGP with multiple neighbors, enable DMZ link-bandwidth, and apply route policies:

Example:

```
Router(config)#interface Loopback0
Router(config-if)#ipv4 address 20.20.20.20 255.255.255.255
Router(config-if)#exit

Router(config)#interface MgmtEth0/RSP0/CPU0/0
Router(config-if)#ipv4 address 8.43.0.10 255.255.255.0
Router(config-if)#exit

Router(config)#interface TenGigE0/3/0/0
Router(config-if)#bandwidth 8000000
Router(config-if)#ipv4 address 11.11.11.11 255.255.255.0
```

```
Router(config-if)#ipv6 address 11:11:0:1::11/64
Router(config-if)#exit

Router(config)#interface TenGigE0/3/0/1
Router(config-if)#bandwidth 7000000
Router(config-if)#ipv4 address 11.11.12.11 255.255.255.0
Router(config-if)#ipv6 address 11:11:0:2::11/64
Router(config-if)#exit

Router(config)#interface TenGigE0/3/0/2
Router(config-if)#bandwidth 6000000
Router(config-if)#ipv4 address 11.11.13.11 255.255.255.0
Router(config-if)#ipv6 address 11:11:0:3::11/64
Router(config-if)#exit

Router(config)#interface TenGigE0/3/0/3
Router(config-if)#bandwidth 5000000
Router(config-if)#ipv4 address 11.11.14.11 255.255.255.0
Router(config-if)#ipv6 address 11:11:0:4::11/64
Router(config-if)#exit

Router(config)#interface TenGigE0/3/0/4
Router(config-if)#bandwidth 4000000
Router(config-if)#ipv4 address 11.11.15.11 255.255.255.0
Router(config-if)#ipv6 address 11:11:0:5::11/64
Router(config-if)#exit

Router(config)#interface TenGigE0/3/0/5
Router(config-if)#bandwidth 3000000
Router(config-if)#ipv4 address 11.11.16.11 255.255.255.0
Router(config-if)#ipv6 address 11:11:0:6::11/64
Router(config-if)#exit

Router(config)#interface TenGigE0/3/0/6
Router(config-if)#bandwidth 2000000
Router(config-if)#ipv4 address 11.11.17.11 255.255.255.0
Router(config-if)#ipv6 address 11:11:0:7::11/64
Router(config-if)#exit

Router(config)#interface TenGigE0/3/0/7
Router(config-if)#bandwidth 1000000
Router(config-if)#ipv4 address 11.11.18.11 255.255.255.0
Router(config-if)#ipv6 address 11:11:0:8::11/64
Router(config-if)#exit

Router(config)#interface TenGigE0/4/0/0
Router(config-if)#description CONNECTED TO IXIA 1/3 transceiver permit pid all
Router(config-if)#exit

Router(config)#interface TenGigE0/4/0/2
Router(config-if)#ipv4 address 9.9.9.9 255.255.0.0
Router(config-if)#ipv6 address 9:9::9/64
Router(config-if)#ipv6 enable
Router(config-if)#exit

Router(config)#route-policy pass-all
Router(config-rpl)#pass
Router(config-rpl)#end-policy

Router(config)#router static
Router(config-static)#address-family ipv4 unicast
Router(config-static-afi)#202.153.144.0/24 8.43.0.1
Router(config-static-afi)#exit
```

Configure BGP unequal cost recursive load balancing: example

```

Router(config-static)#exit

Router(config)#router bgp 100
Router(config-bgp)#bgp router-id 20.20.20.20
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#maximum-paths eibgp 8
Router(config-bgp-af)#redistribute connected
Router(config-bgp-af)#exit

Router(config-bgp)#neighbor 11.11.11.12
Router(config-bgp-nbr)#remote-as 200
Router(config-bgp-nbr)#dmz-link-bandwidth
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#route-policy pass-all in
Router(config-bgp-nbr-af)#route-policy pass-all out
Router(config-bgp-nbr-af)#exit
Router(config-bgp-nbr)#exit

Router(config-bgp)#neighbor 11.11.12.12
Router(config-bgp-nbr)#remote-as 200
Router(config-bgp-nbr)#dmz-link-bandwidth
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#route-policy pass-all in
Router(config-bgp-nbr-af)#route-policy pass-all out
Router(config-bgp-nbr-af)#exit
Router(config-bgp-nbr)#exit

Router(config-bgp)#neighbor 11.11.13.12
Router(config-bgp-nbr)#remote-as 200
Router(config-bgp-nbr)#dmz-link-bandwidth
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#route-policy pass-all in
Router(config-bgp-nbr-af)#route-policy pass-all out
Router(config-bgp-nbr-af)#exit
Router(config-bgp-nbr)#exit

Router(config-bgp)#neighbor 11.11.14.12
Router(config-bgp-nbr)#remote-as 200
Router(config-bgp-nbr)#dmz-link-bandwidth
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#route-policy pass-all in
Router(config-bgp-nbr-af)#route-policy pass-all out
Router(config-bgp-nbr-af)#exit
Router(config-bgp-nbr)#exit

Router(config-bgp)#neighbor 11.11.15.12
Router(config-bgp-nbr)#remote-as 200
Router(config-bgp-nbr)#dmz-link-bandwidth
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#route-policy pass-all in
Router(config-bgp-nbr-af)#route-policy pass-all out
Router(config-bgp-nbr-af)#exit
Router(config-bgp-nbr)#exit

Router(config-bgp)#neighbor 11.11.16.12
Router(config-bgp-nbr)#remote-as 200
Router(config-bgp-nbr)#dmz-link-bandwidth
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#route-policy pass-all in
Router(config-bgp-nbr-af)#route-policy pass-all out
Router(config-bgp-nbr-af)#exit
Router(config-bgp-nbr)#exit

```



```

Router(config-bgp)#neighbor 11.11.17.12
Router(config-bgp-nbr)#remote-as 200
Router(config-bgp-nbr)#dmz-link-bandwidth
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#route-policy pass-all in
Router(config-bgp-nbr-af)#route-policy pass-all out
Router(config-bgp-nbr-af)#exit
Router(config-bgp-nbr)#exit

Router(config-bgp)#neighbor 11.11.18.12
Router(config-bgp-nbr)#remote-as 200
Router(config-bgp-nbr)#dmz-link-bandwidth
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#route-policy pass-all in
Router(config-bgp-nbr-af)#route-policy pass-all out
Router(config-bgp-nbr-af)#exit
Router(config-bgp-nbr)#exit

Router(config-bgp)#exit
Router(config)#commit

```

Step 2 Verify the running configuration.

Example:

```

Router# show running-config
interface Loopback0
  ipv4 address 20.20.20.20 255.255.255.255
!
!
interface FourHundredGige0/1/0/0
  bandwidth 8000000
  ipv4 address 11.11.11.11 255.255.255.0
  ipv6 address 11:11:0:1::11/64
!
interface FourHundredGige0/0/0/0
  bandwidth 7000000
  ipv4 address 11.11.12.11 255.255.255.0
  ipv6 address 11:11:0:2::11/64
!
interface FourHundredGige0/3/0/0
  bandwidth 6000000
  ipv4 address 11.11.13.11 255.255.255.0
  ipv6 address 11:11:0:3::11/64
!
interface FourHundredGige0/4/0/0
  bandwidth 5000000
  ipv4 address 11.11.14.11 255.255.255.0
  ipv6 address 11:11:0:4::11/64
!
interface FourHundredGige0/0/0/0
  bandwidth 4000000
  ipv4 address 11.11.15.11 255.255.255.0
  ipv6 address 11:11:0:5::11/64
!
interface FourHundredGige0/2/0/0
  bandwidth 3000000
  ipv4 address 11.11.16.11 255.255.255.0
  ipv6 address 11:11:0:6::11/64
!
interface FourHundredGige0/3/0/0
  bandwidth 2000000
  ipv4 address 11.11.17.11 255.255.255.0

```

Configure BGP unequal cost recursive load balancing: example

```

    ipv6 address 11:11:0:7::11/64
    !
interface FourHundredGige0/3/0/0
    bandwidth 1000000
    ipv4 address 11.11.18.11 255.255.255.0
    ipv6 address 11:11:0:8::11/64
    !
interface FourHundredGige0/4/0/0
    description CONNECTED TO IXIA 1/3
    transceiver permit pid all
    !
interface FourHundredGige0/4/0/0
    ipv4 address 9.9.9.9 255.255.0.0
    ipv6 address 9:9::9/64
    ipv6 enable
    !
route-policy pass-all
    pass
end-policy
!
router static
    address-family ipv4 unicast
        202.153.144.0/24 8.43.0.1
    !
!
router bgp 100
    bgp router-id 10.20.20.20
    address-family ipv4 unicast
        maximum-paths eibgp 8
        redistribute connected
    !
neighbor 11.11.11.12
    remote-as 200
    dmz-link-bandwidth
    address-family ipv4 unicast
        route-policy pass-all in
        route-policy pass-all out
    !
!
neighbor 11.11.12.12
    remote-as 200
    dmz-link-bandwidth
    address-family ipv4 unicast
        route-policy pass-all in
        route-policy pass-all out
    !
!
neighbor 10.11.13.12
    remote-as 200
    dmz-link-bandwidth
    address-family ipv4 unicast
        route-policy pass-all in
        route-policy pass-all out
    !
!
neighbor 11.11.14.12
    remote-as 200
    dmz-link-bandwidth
    address-family ipv4 unicast
        route-policy pass-all in
        route-policy pass-all out
    !
!
neighbor 11.11.15.12

```

```

remote-as 200
dmz-link-bandwidth
address-family ipv4 unicast
  route-policy pass-all in
  route-policy pass-all out
!
!
neighbor 11.11.16.12
  remote-as 200
  dmz-link-bandwidth
  address-family ipv4 unicast
    route-policy pass-all in
    route-policy pass-all out
  !
!
neighbor 11.11.17.12
  remote-as 200
  dmz-link-bandwidth
  address-family ipv4 unicast
    route-policy pass-all in
    route-policy pass-all out
  !
!
neighbor 11.11.18.12
  remote-as 200
  dmz-link-bandwidth
  address-family ipv4 unicast
    route-policy pass-all in
    route-policy pass-all out
  !
!
!

```

BGP DMZ link bandwidth enhancement

BGP DMZ link bandwidth enhancement is a BGP feature that

- accurately advertises all valid link bandwidth values
- preserves precision in bandwidth signaling, and
- allows sending and advertising any valid BGP DMZ link bandwidth value, including values less than 1 kilobit per second (kbps).

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
BGP DMZ link bandwidth enhancement	Release 25.3.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) * This feature is now supported on Cisco 8404-SYS-D routers.

Feature Name	Release Information	Feature Description
BGP DMZ link bandwidth enhancement	Release 25.1.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q200, P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100]); Centralized Systems (8600 [ASIC: Q200]); Modular Systems (8800 [LC ASIC: Q100, Q200, P100]) You can now send and advertise any valid BGP DMZ link bandwidth value including values below 1 kilobit per second (kbps). Previously, BGP rounded smaller link bandwidth values and failed to advertise values below 1 kbps. The feature introduces these changes: CLI: The show bgp command is enhanced to display link bandwidth values in both kbps and bytes per second for improved precision and visibility.

The feature allows you to send and advertise link bandwidth values of any size, overcoming the previous format limitation that restricted values to greater than 1 kbps. With the enhancement, RPL supports 64-bit link bandwidth values.

Benefits of BGP DMZ link bandwidth enhancement

The key benefits of the feature are:

- BGP avoids rounding issues that affected smaller bandwidth values.
- Supports a wider and more granular range of bandwidth values.
- Precise bandwidth advertisement ensures that any valid link bandwidth value is advertised accurately and preserves network performance metrics.

Usage guidelines for BGP DMZ link bandwidth enhancement

- All address families support the feature. However, it is primarily used only with the IPv4 and IPv6 address families that are installed in the RIB by the default and specific VRFs.
- You cannot use DMZ link bandwidth with Multihop eBGP.

Verify DMZ link bandwidth enhancement

Verify the DMZ link bandwidth enhancement.

Before you begin

Follow these steps to verify the DMZ link bandwidth enhancement.

Procedure

Step 1 Run the **show route** command to verify the status of specific routes in the routing table.

Example:

```
Router#show route 209.165.201.1/27
...
Routing entry for 209.165.201.1/27
  Known via "bgp 1", distance 200, metric 0
  Tag 2, type internal
  Installed Mar  6 09:14:55.055 for 00:06:10
  Routing Descriptor Blocks
    1.1.1.1, from 1.1.1.1, BGP multi path
      Route metric is 0, Wt is 16000
    1.1.1.2, from 1.1.1.2, BGP multi path
      Route metric is 0, Wt is 8000
  ...
```

In the sample output, BGP utilizes link bandwidth values for multipath load balancing and adjusts and distributes weights across different paths.

Step 2 Run the **show cef** command to verify load sharing based on weights derived from link bandwidth values.

Example:

```
Router#show cef 209.165.201.1/27
...
  Weight distribution:
    slot 0, weight 16000, normalized_weight 2
    slot 1, weight 8000, normalized_weight 1

  Level 1 - Load distribution: 0 0 1
    [0] via 1.1.1.1/32, recursive
    [1] via 1.1.1.1/32, recursive
    [2] via 1.1.1.2/32, recursive
  ...
```

In the sample output, the load distribution levels indicate how packets are distributed among various paths. The weight distribution values indicate how weights are normalized across various slots.

Step 3 Run the **show bgp** command to verify the link bandwidth in bytes per second.

Example:

```
Router#show bgp 209.165.201.1/27
Thu Mar  6 09:22:09.823 UTC
BGP routing table entry for 209.165.201.1/27
Versions:
  Process          bRIB/RIB      SendTblVer
  Speaker          160385       160385
Last Modified: Mar  6 09:14:55.098 for 00:07:14
Paths: (2 available, best #1)
```

```

Not advertised to any peer
Path #1: Received by speaker 0
Not advertised to any peer
2, (received & used)
  1.1.1.1 (metric 20) from 1.1.1.1 (1.1.1.5)
    Origin IGP, localpref 100, valid, internal, best, group-best, multipath
    Received Path ID 1, Local Path ID 1, version 160385
    Extended community: LB:1000:128
                        (LB non-transitive AS:bytes/sec:1000:16000.0)
    Originator: 1.1.1.5, Cluster list: 1.1.1.1
Path #2: Received by speaker 0
Not advertised to any peer
2, (received & used)
  1.1.1.2 (metric 20) from 1.1.1.2 (1.1.1.5)
    Origin IGP, localpref 100, valid, internal, multipath
    Received Path ID 1, Local Path ID 0, version 0
    Extended community: LB:1000:64
                        (LB non-transitive AS:bytes/sec:1000:8000.0)
    Originator: 1.1.1.5, Cluster list: 1.1.1.2

```

In the example show output, BGP advertises a link bandwidth value of 16,000.0 bytes per second for the link bandwidth weight 128 and 8,000 bytes per second for the link bandwidth weight 64.

BGP DMZ transitive-bandwidth extended community support

BGP DMZ transitive-bandwidth extended community is a feature that allows BGP to process and make routing decisions based on the available transitive-bandwidth extended community using UCMP.

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
BGP DMZ transitive-bandwidth extended community support	Release 25.3.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) * This feature is now supported on Cisco 8404-SYS-D routers.

Feature Name	Release Information	Feature Description
BGP DMZ transitive-bandwidth extended community support	Release 25.1.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q200, P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100]); Centralized Systems (8600 [ASIC: Q200]); Modular Systems (8800 [LC ASIC: Q100, Q200, P100]) You can now enable BGP to process incoming DMZ transitive-bandwidth extended community, allowing bandwidth-aware routing decisions using Unequal Cost Multi-Path (UCMP). The feature allows RPL to manually set the DMZ transitive-bandwidth extended community for BGP neighbors. This extended propagation supports multivendor interoperability, optimizes traffic distribution, prevents link over utilization, and balances load across available paths. Previously, BGP supported only the non-transitive extended community. The feature introduces these changes: CLI: The transitive-bandwidth type is introduced as an extended community in RPL. YANG Data Models: • <code>Cisco-IOS-XR-um-route-policy-cfg</code> • <code>Cisco-IOS-XR-policy-repository-cfg</code> (see GitHub, YANG Data Models Navigator)

You can now configure BGP to process incoming DMZ transitive-bandwidth extended communities, enabling bandwidth-aware routing decisions through UCMP. This enhancement allows the router to interpret linked bandwidth values that are attached to incoming BGP routes and use them to distribute traffic proportionally across multiple paths based on available capacity.

In addition, the feature extends RPL support to manually set the DMZ transitive-bandwidth extended community for inbound and outbound BGP updates. This extended propagation supports multivendor interoperability, optimizes traffic distribution, and prevents link over utilization.

Benefits of BGP DMZ transitive-bandwidth extended community

The key benefits of the feature are:

- Ensures multivendor interoperability and utilizes the DMZ bandwidth value across heterogeneous networks.
- Adds a weight that enables RIB or FIB to intelligently load balance traffic across multiple paths.
- Ensures consistent network policies across domains.

How BGP DMZ transitive-bandwidth extended communities work

Summary

The key components involved in the process are:

- BGP router: Assigns and propagates DMZ transitive-bandwidth values.
- BGP peers: Receive and utilize the propagated bandwidth attributes.
- RIB/FIB: Use bandwidth values to make intelligent load balancing decisions.

The BGP DMZ transitive-bandwidth extended community workflow involves assigning bandwidth values, propagating them, and then using these values for traffic load balancing.

Workflow

These stages describe how BGP DMZ transitive-bandwidth extended community works.

1. Policy-based bandwidth assignment: A BGP router assigns a DMZ transitive-bandwidth value to BGP routes using an egress or ingress RPL.
2. Propagation of the bandwidth attribute: The router propagates the transitive-bandwidth extended community along with the route to its BGP peers.
3. Traffic is load balanced based on the bandwidth: Routers use the bandwidth values to balance traffic across multiple paths, favoring higher-capacity links.

Result

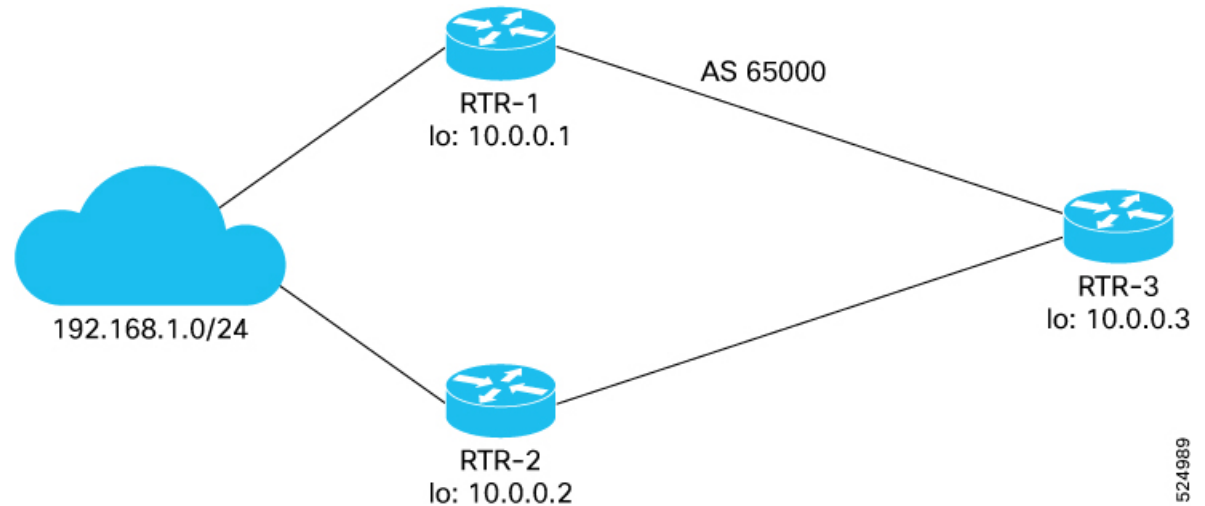
The system achieves optimized traffic distribution and load balancing across available paths based on bandwidth.

Topology for BGP path selection using UCMP

To describe a sample network configuration that demonstrates bandwidth-aware BGP path selection using UCMP.

The figure shows an example topology for bandwidth-aware BGP path selection using UCMP.

Figure 1: Example topology for bandwidth-aware BGP path selection using UCMP



The topology consists of three BGP routers: RTR-1, RTR-2, and RTR-3. All the routers are within the same AS (65000), forming an Internal Border Gateway Protocol network.

- RTR-1 and RTR-2 learn and advertise an external prefix 192.168.1.0/24. The routers apply an RPL outbound toward RTR-3, setting the link-bandwidth value through the transitive-bandwidth extended community.
- RTR-3 receives the prefix from RTR-1 and RTR-2, evaluates their advertised Link Bandwidth (LB) values, and uses UCMP to make routing decisions.

Consider a scenario where RTR-1 advertises an LB value of 65000:1250000000, and RTR-2 advertises 65000:2500000000. Based on these LB values, RTR-3 forwards approximately 33% of the traffic to RTR-1 and 67% to RTR-2, effectively sending twice as much traffic to RTR-2. This traffic distribution allows RTR-3 to balance the load according to the relative bandwidth of each link.

Configure BGP DMZ transitive-bandwidth extended community

Before you begin

Ensure that the network supports DMZ bandwidth extended communities for signaling link bandwidth.

Procedure

Step 1 Configure the transitive-bandwidth extended community for a specific prefix.

- Configure the transitive-bandwidth extended community directly within the route policy.

In the sample configuration for RTR-1, the route policy for RTR-1 advertises prefix 192.168.1.0/24, and attaches a transitive-bandwidth extended community to that route. The ASN is 65000 and the link-bandwidth is 1,250,000,000 bytes per second.

```
RTR-1#config
RTR-1 (config) #route-policy SET-LB
RTR-1 (config-rpl) #if destination in (192.168.1.0/24) then
```

```
RTR-1(config-rpl-if)#set extcommunity transitive-bandwidth (65000:1250000000)
RTR-1(config-rpl-if)#endif
RTR-1(config-rpl)#end-policy
```

- Configure the transitive-bandwidth extended community using the **extcommunity-set** command, and create a route policy that applies the transitive-bandwidth extended community to the route.

In the sample configuration for RTR-2, the extended community set is LB that matches the transitive-bandwidth extended community. The route policy SET-LB applies the transitive-bandwidth extended community that is defined by the named set LB to RTR-2.

```
RTR-2#config
RTR-2(config)#extcommunity-set transitive-bandwidth LB
RTR-2(config-ext)#65000:2500000000
RTR-2(config-ext)#end-set
RTR-2(config)#route-policy SET-LB
RTR-2(config-rpl)#if destination in (192.168.1.0/24) then
RTR-2(config-rpl-if)#set extcommunity transitive-bandwidth LB
RTR-2(config-rpl-if)#endif
RTR-2(config-rpl)#end-policy
```

Step 2 Run the **router bgp** command on RTR-1 and RTR-2 to enable BGP for the required ASN, and apply the configured route policy to outbound updates for the required neighbor.

Example:

In the sample configuration, BGP is enabled for AS 65000 on RTR-1 and RTR-2, and applies the SET-LB route policy to outbound updates to the neighbor RTR-3 (10.0.0.3).

```
Router(config)#router bgp 65000
Router(config-bgp)#neighbor 10.0.0.3
Router(config-bgp-nbr)#remote-as 65000
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#route-policy SET-LB out
```

Step 3 Run the **show running-config** command to verify the running configuration for RTR-1 and RTR-2.

Example:

This is a sample output from RTR-1.

```
route-policy SET-LB
  if destination in (192.168.1.0/24) then
    set extcommunity transitive-bandwidth (65000:1250000000)
  endif
end-policy
!
router bgp 65000
  neighbor 10.0.0.3
  remote-as 65000
  address-family ipv4 unicast
    route-policy SET-LB out
  !
!
```

Example:

This is a sample output from RTR-2.

```
extcommunity-set transitive-bandwidth LB
  65000:2500000000
end-set
!
```

```

route-policy SET-LB
  if destination in (192.168.1.0/24) then
    set extcommunity transitive-bandwidth LB
  endif
end-policy
!
router bgp 65000
  neighbor 10.0.0.3
    remote-as 65000
    address-family ipv4 unicast
      route-policy SET-LB out
    !
  !
!

```

Step 4 Run the **show bgp ipv4 unicast advertised neighbor** command to verify the routes that are advertised from RTR-1 to RTR-3, and RTR-2 to RTR-3.

Example:

The sample show output displays the routes that are advertised from RTR-1 to RTR-3. The next-hop and router ID 10.0.0.1 indicates that the route came from RTR-1. The *extended community: LB_TRANS:65000:10000000* value in kbps confirms that RTR-1 supports and advertises the link-bandwidth attribute.

```

RTR-1#show bgp ipv4 unicast advertised neighbor 10.0.0.3
Fri Apr  4 08:09:06.297 UTC
192.168.1.0/24 is advertised to 10.0.0.3
  Path info:
    neighbor: Local          neighbor router id: 10.0.0.1
    valid redistributed best
Received Path ID 0, Local Path ID 1, version 5
Attributes after inbound policy was applied:
  next hop: 0.0.0.0
  MET ORG AS
  origin: incomplete metric: 0
  aspath:
Attributes after outbound policy was applied:
  next hop: 10.0.0.1
  MET ORG AS EXTCOMM
  origin: incomplete metric: 0
  aspath:
  extended community: LB_TRANS:65000:10000000

```

Example:

The sample show output displays the routes that are advertised from RTR-2 to RTR-3. The next-hop and router ID 10.0.0.2 indicates that the route came from RTR-2. The *extended community: LB_TRANS:65000:20000000* value in kbps confirms that RTR-2 supports and advertises the link-bandwidth attribute.

```

RTR-2#show bgp ipv4 unicast advertised neighbor 10.0.0.3
Fri Apr  4 08:12:03.484 UTC
192.168.1.0/24 is advertised to 10.0.0.3
  Path info:
    neighbor: Local          neighbor router id: 10.0.0.2
    valid redistributed best
Received Path ID 0, Local Path ID 1, version 5
Attributes after inbound policy was applied:
  next hop: 0.0.0.0
  MET ORG AS
  origin: incomplete metric: 0
  aspath:
Attributes after outbound policy was applied:
  next hop: 10.0.0.2

```

Configure BGP DMZ transitive-bandwidth extended community

```

MET ORG AS EXTCOMM
origin: incomplete metric: 0
aspath:
extended community: LB_TRANS:65000:20000000

```

Step 5 Run the **show bgp ipv4 unicast** command on RTR-3, to verify the multiple paths and extended community attributes for load-balancing bandwidth advertisement.

Example:

In the sample show output from RTR-3, the highlighted values refer to the transitive-bandwidth extended communities, which are used to make routing decisions using UCMP.

```

RTR-3#show bgp ipv4 unicast 192.168.1.0/24
Fri Apr 4 08:03:49.333 UTC
BGP routing table entry for 192.168.1.0/24
Versions:
  Process          bRIB/RIB    SendTblVer
  Speaker          8           8
Last Modified: Apr 4 08:03:43.608 for 00:00:05
Paths: (2 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local
    10.0.0.1 (metric 10) from 10.0.0.1 (10.0.0.1)
    Origin incomplete, metric 0, localpref 100, valid, internal, best, group-best, multipath
    Received Path ID 0, Local Path ID 1, version 8
    Extended community: LB_TRANS:65000:10000000
                        (LB transitive AS:bytes/sec:65000:1250000000.0)
  Path #2: Received by speaker 0
  Not advertised to any peer
  Local
    10.0.0.2 (metric 10) from 10.0.0.2 (10.0.0.2)
    Origin incomplete, metric 0, localpref 100, valid, internal, multipath
    Received Path ID 0, Local Path ID 0, version 0
    Extended community: LB_TRANS:65000:20000000
                        (LB transitive AS:bytes/sec:65000:2500000000.0)

```

Step 6 Run the **show route** command on RTR-3, to verify that the routing decision reflects the intended path preferences set by the BGP policies, when using transitive-bandwidth extended communities for UCMP-based load balancing.

Example:

The sample show output from RTR-3 indicates that the router has learned route 192.168.1.0/24 through BGP (AS 65000), and is using BGP multipath to forward traffic across the two next-hops.

RTR-1 (10.0.0.1) has a weight of 10000000, and RTR-2 (10.0.0.2) has a weight of 20000000. The router uses these values to load balance traffic, sending more traffic through RTR-2 because of the higher weight.

```

RTR-3#show route 192.168.1.0/24
Fri Apr 4 08:03:53.280 UTC

Routing entry for 192.168.1.0/24
  Known via "bgp 65000", distance 200, metric 0, type internal
  Installed Apr 4 08:03:43.605 for 00:00:09
  Routing Descriptor Blocks
    10.0.0.1, from 10.0.0.1, BGP multi path
      Route metric is 0, Wt is 10000000
    10.0.0.2, from 10.0.0.2, BGP multi path
      Route metric is 0, Wt is 20000000
  No advertising protos.

```

Step 7 Run the **show cef** command on RTR-3, to view detailed information about the CEF entry for the IP prefix 192.168.1.0/24.

Example:

In the sample show output, slot 0 is assigned 1 out of 3 buckets, and slot 1 is assigned 2 out of 3 buckets. In other words, the router forwards traffic in a 1:2 ratio, sending approximately 33% through slot 0 and 67% through slot 1.

```
RTR-3#show cef 192.168.1.0/24
Fri Apr  4 08:04:03.111 UTC
192.168.1.0/24, version 20, internal 0x5000001 0x40 (ptr 0x9c449908) [1], 0x0 (0x0), 0x0 (0x0)
Updated Apr  4 08:03:43.610
Prefix Len 24, traffic index 0, precedence n/a, priority 4
gateway array (0x9c2b0798) reference count 1, flags 0x2010, source rib (7), 0 backups
    [1 type 3 flags 0x48441 (0x9c3557d8) ext 0x0 (0x0)]
LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
gateway array update type-time 1 Apr  4 08:03:43.610
LDI Update time Apr  4 08:03:43.633

Weight distribution:
slot 0, weight 10000000, normalized_weight 1
slot 1, weight 20000000, normalized_weight 2

Level 1 - Load distribution: 0 1 1
[0] via 10.0.0.1/32, recursive
[1] via 10.0.0.2/32, recursive
[2] via 10.0.0.2/32, recursive

via 10.0.0.1/32, 3 dependencies, recursive, bgp-multipath [flags 0x6080]
path-idx 0 NHID 0x0 [0x9c449818 0x0], Internal 0xb329c0a0
next hop 10.0.0.1/32 via 10.0.0.1/32

Load distribution: 0 (refcount 1)

Hash OK Interface Address
0 Y FourHundredGigE0/0/0/0 10.10.13.1

via 10.0.0.2/32, 3 dependencies, recursive, bgp-multipath [flags 0x6080]
path-idx 1 NHID 0x0 [0x9c449728 0x0], Internal 0xb329c220
next hop 10.0.0.2/32 via 10.0.0.2/32

Load distribution: 0 (refcount 1)

Hash OK Interface Address
1 Y FourHundredGigE0/0/0/1 10.10.23.2
```

