Description

This feature adds support for BGP UCMP in the multi-agent routing protocol model. The TOI for BGP UCMP in the ribd routing protocol model can be found here.

Unequal Cost Multi Path (UCMP) for BGP is a mechanism for forwarding traffic from a device for an ECMP route in the ratio of the weights with which the next hops of that route are programmed in the FIB. This is done for BGP by disseminating BGP link bandwidth extended community attribute information with BGP paths such that the receiver device of all such paths for a route programs next hops for that route in the FIB in the ratio of the received link bandwidth values.

Platform Compatibility

This feature is platform independent.

Configuration

The following configuration commands have been added as part of this feature support.

As of EOS-4.22.1F:

- ucmp mode <mode#> <M> <+deviation%>
- neighbor <peer|peer-group> link-bandwidth auto
- neighbor <peer|peer-group> link-bandwidth default 0.0-4294967295.0[K|M|G]
- ucmp link-bandwidth encoding-weighted
- ip extcommunity-list <list> permit|deny lbw any
- ip extcommunity-list <list> permit|deny lbw <ASN:0.0-4294967295.0[K|M|G]>
- set extcommunity lbw <0.0-4294967295.0 or nn.nn(K|M|G)<
- set extcommunity lbw <ASN:0.0-4294967295.0 or ASN:nn.nn(K|M|G)> delete

As of EOS-4.24.1F:

- neighbor <> send-community link-bandwidth (aggregate [reference] | divide (equal | ratio))
- set extcommunity lbw (aggregate [reference] | divide (equal | ratio))
- neighbor <peer|peer-group> link-bandwidth update-delay <0-600>
- ucmp link-bandwidth update-delay <0-600>

Enabling UCMP locally

Following command under ‘router bgp’ mode will be used to enable UCMP on the local device to perform link bandwidth based traffic load sharing for BGP next-hops.

```
[no|default] ucmp mode <mode#> <M> <+deviation%>
```

UCMP can be configured in global BGP configuration mode with ‘ucmp mode 1’ configuration. UCMP mode 1 takes the following optional parameters:

- (Optional) A maximum set size for the UCMP next-hop set (M)
- (Optional) A % deviation constraint from the optimal traffic load sharing ratio for every next-hop in the UCMP set (deviation%)

The optimal or desired load sharing ratio for the i’th next-hop in the UCMP set is (lbwi / LBW). lbwi is the link bandwidth corresponding to the i’th next-hop while the denominator is the sum of link bandwidths received for all next-hops in the UCMP set.

For mode 1, UCMP will be expressed in terms of an integer repetition count for every next-hop in the ECMP set, where the sum of next-hop repetition counts for all next-hops adds up to say N (N <= M). The purpose of the deviation constraint is to bound the computed traffic sharing ratio for any next-hop in the set (n/N) to be within the % range specified by the optimal load sharing ratio (lbwi / LBW).

If these optional parameters aren’t configured, default values will be chosen for the UCMP next-hop set size which equals the configured max-ecmp limit or platform maximum ECMP limit and a default % deviation constraint of 5% will be used. ‘maximum-paths <max-paths> [ecmp <max-ecmp>]’ needs to be configured before configuring UCMP globally since UCMP relies on ECMP groups being formed first. The configurable UCMP set-size will be bounded by the configured values of max-paths and max-ecmp.

Example:

```
Arista(config-router-bgp)#ucmp mode 1 32 0.01
```

‘ucmp mode 1 [<M> [<deviation%>]]’ is a mandatory command along with ‘maximum-paths <max-paths> [ecmp <max-ecmp>]’ command to enable UCMP for received routes at the DUT locally.

Enabling BGP Link Bandwidth attribute setting inbound

There is no configuration necessary to receive link bandwidth attributes from a BGP peer.

If a BGP route is received without a link bandwidth attribute, and a link bandwidth attribute is not added to the route by an inbound route-map, there are two ways to introduce a link bandwidth attribute to the route:

For a given BGP neighbor, the following command can be used for auto generation of link bandwidth attribute from the interface speed over which the peering for this neighbor is established, assuming routes advertised by this neighbor do not already carry the link bandwidth attribute. The link speed can be derived only for BGP peering sessions established over LAGs and Routed Port interfaces, not for SVIs.

```
[no|default] neighbor <peer|peer-group> link-bandwidth auto
```

An alternative to the ‘auto’ inbound link bandwidth generation is ‘default’ inbound link bandwidth generation. default’ link bandwidth generation can be used to provide a default link bandwidth attribute for routes received from a neighbor, assuming routes advertised by this neighbor do not already carry the link bandwidth attribute.

```
[no|default] neighbor <peer|peer-group> link-
bandwidth default 0.0-4294967295.0[K|M|G]
```

The value for the default link bandwidth configuration can be accepted in Kbps, Mbps or Gbps suffixed by K|M|G as seen above.

Example:

```
Arista(config-router-bgp)#neighbor 192.0.2.1 link-
bandwidth default 10G
```

The command above will set a default link bandwidth of 10Gbps for routes received without any link bandwidth attribute from peer 192.0.2.1.

Enabling BGP Link Bandwidth attribute regeneration outbound

The following commands extend the existing ‘neighbor <> send-community’ CLI available on a per neighbor/peer-group basis to additionally accept the ‘link-bandwidth’ keyword along with options for regenerating the link bandwidth attribute value for an advertised route.

The accumulated link bandwidth attribute value received for an ECMP route can be propagated to eligible peers in more than one way. Configuring which model/mode to use for regeneration is at the discretion of the administrator/operator and can be propagated either in an aggregate manner or with fractionalized bandwidth values to each of the eligible BGP peers.

Aggregate Regeneration Mode

The following configuration is used to advertise the aggregate link bandwidth of a route. The aggregate link bandwidth of a route is the cumulative link bandwidth received for all paths comprising the ECMP set for that route. If the route is not ECMP, the aggregate link bandwidth is the link bandwidth of the active path for that route.

```
[no|default] neighbor <peer|peer-group> send-community link-bandwidth aggregate
```

An alternative while advertising the aggregate link bandwidth of a route to a peer is to compute a multiplier or ratio from the current interface speed of the peering interface with respect to a reference link bandwidth (interface speed) configurable for that peer. This can be done with an optional <ref-bandwidth> configuration suffix after the aggregate keyword. Again, the interface speed can be derived only for BGP peering sessions established over LAGs and Routed Port interfaces, not for SVIs.

```
[no|default] neighbor <peer|peer-group> send-community link-bandwidth aggregate 0.0-4294967295.0[K|M|G]
```

Example:

```
Arista(config-router-bgp)#neighbor 192.0.2.1 send-community link-bandwidth aggregate 40G
```

The example above configures aggregate link bandwidth regeneration with reference link bandwidth of 40G. If the peer link-speed for peer 192.0.2.1 is 10G, a multiplier of 1/4 will be in effect while advertising the aggregate link bandwidth value of every route sent to this peer. Note that this multiplier can never exceed 1; should the reference link bandwidth value be lesser than the peer link-speed, we simply use a multiplier of 1. Ergo, we do not allow the advertised value to be inflated in this manner.

Divide Regeneration Mode

Instead of advertising the aggregate link bandwidth for a route to every peer, the following configuration aids in dividing the aggregate value equally across all peers for whom this configuration is in effect. One constraint is that all such peers should share common outbound policies.

```
[no|default] neighbor <peer|peer-group> send-community link-bandwidth divide equal
```

Instead of dividing the aggregate link bandwidth of a route equally across all peers to whom it’s advertised, the following configuration command helps in link-speed ratio based division of the aggregate link bandwidth value. Essentially, a multiplier is computed on a per peer basis using the peer link-speed with respect to the aggregate link-speed of all peers for whom this configuration is in effect. The aggregate link bandwidth value is subject to this multiplier to compute the link bandwidth value to advertise. As before, all peers configured with this command are subject to the constraint of sharing common outbound policies for division across all such peers to be in effect.

```
[no|default] neighbor <peer|peer-group> send-community link-bandwidth divide ratio
```

Example:

```
Arista(config-router-bgp)#neighbor 192.0.2.10 send-community link-bandwidth divide ratio
Arista(config-router-bgp)#neighbor 192.0.2.20 send-community link-bandwidth divide ratio
Arista(config-router-bgp)#neighbor 192.0.2.30 send-community link-bandwidth divide ratio
```

Consider in the example above, 3 peers are configured with ‘link-bandwidth divide ratio’ regeneration policy. If the 3 peers have link-speeds of 10G, 40G and 40G respectively, the aggregate link bandwidth of every route is subject to a multiplier of 1/9, 4/9 and 4/9 before advertisement to each peer respectively.

With every link bandwidth regeneration scheme, the advertised link bandwidth is capped by the peer’s link-speed in every scenario, unless the device is configured in link-bandwidth encoding-weighted mode.

Enabling BGP Link Bandwidth attribute regeneration outbound using route-maps

The route-map configuration mode will also support all the link bandwidth regeneration schemes described earlier that were configurable on a per neighbor basis. The advantage of supporting these regenerations schemes within a route-map is to be able to provide per-prefix level granularity using match statements in conjunction with set statements.

As such, the following link bandwidth regeneration schemes will be configurable within route-map configuration mode.

The following 2 commands are for advertising the aggregate link bandwidth of a route and aggregate with reference link bandwidth configuration respectively for all peers to which the route-map is in effect in the outbound direction.

```
[no|default] set extcommunity lbw aggregate
[no|default] set extcommunity lbw aggregate 0.0-4294967295.0[K|M|G]
```

The following 2 commands will be used for division of the aggregate link bandwidth equally or ratio based division respectively across all peers for whom the route-map containing this set statement is in effect in the outbound direction. The constraint for having all peers with this route-map application sharing all common outbound policies still applies though.

```
[no|default] set extcommunity lbw divide equal
[no|default] set extcommunity lbw divide ratio
```

If the link bandwidth regeneration schemes are configured for a route-map that is applied in the inbound direction, they do not have any effect.

Enabling BGP Link Bandwidth attribute setting using route-maps

A new set clause has been added in Route-map Config Mode for the link-bandwidth attribute. As default behavior, the link bandwidth extended community set action on an outbound/inbound route-map will overwrite an existing link bandwidth attribute (if present) on the route.

The following command is used within route-map configuration mode to configure setting of link bandwidth attribute value. The route-map can then be applied to peers or peer-groups for whom this set policy for advertised routes will be in effect. This command helps provide granularity on a per route basis to set different values of link bandwidth attribute using match and set clauses with the route-map. This set clause takes effect for both inbound and outbound applied route-maps.

```
[no|default] set extcommunity lbw 0.0-4294967295.0[K|M|G]
```

The following command is used for deletion of a certain link bandwidth attribute from a route either on the inbound during reception of the route or on outbound while advertising the route depending on where the route-map containing this set statement is in effect. The command requires providing an ASN (BGP Autonomous System) value along with the link bandwidth value for deletion from the route. Only those routes matching both the ASN and the link bandwidth value will be subject to deletion.

```
[no|default] set extcommunity lbw ASN:0.0-4294967295.0[K|M|G] delete
```

Enabling BGP Link Bandwidth attribute matching using extended community lists

The following command is used to match against a specified link bandwidth extended community attribute value or any link bandwidth extended community attribute value. If the match does not succeed, the next route-map clause can have a set action to set a ‘default’ extended community link bandwidth attribute value on the route. This is useful for application in inbound route-maps when routes are not received with any link bandwidth attribute value at all and a default value can be used to keep UCMP in effect for that route.

```
[no|default] ip extcommunity-
list <name> permit|deny lbw <any|ASN:0.0-4294967295.0[K|M|G]>
```

Enabling update-delay for Link Bandwidth attribute

The following command enables delay in generation of route advertisement updates when the link bandwidth extended community value keeps increasing for paths received from a neighbor.

```
[no|default] neighbor <peer|peer-group> link-bandwidth update-
delay <0-600>
```

The delay is specified in a 600s range. This out delay applies to update churn resulting in link bandwidth increase for an advertised route. Update churn resulting in a decrease in advertised link bandwidth for a route will not be subject to any delay. The reasoning behind this is to coalesce the incremental increases due to accumulation of received link bandwidth when routes are initially being advertised so as to prevent unnecessary churn at the next layer.

A corresponding global BGP configuration mode command is also available to enforce a link bandwidth related churn delay in advertisement of routes for that BGP instance.

```
[no|default] ucmp link-bandwidth update-delay <0-600>
```

The update-delay command is an optional configuration to alleviate unnecessary churn that might be caused by link bandwidth increase.

Enabling weighted mode for Link Bandwidth attribute

The link bandwidth attribute prescribed by specification is advertised in units of Bytes/s. In the context of the UCMP solution locally, the received link bandwidth attribute from paths of an ECMP route have meaning relative to each other only. There could be a need to express the link bandwidth attribute in easily interpretable weight values rather than large link speeds in Gbps.

The following optional globally configurable CLI command treats the link bandwidth attribute value as a raw weight across received and advertised routes.

```
[no|default] ucmp link-bandwidth encoding-weighted
```

In weighted mode, any link bandwidth values input to the CLI will be interpreted as raw values without units instead of in bits per second.

When the link bandwidth attribute is expressed/interpreted as weights, it does not make sense to equate received/advertised weight values with actual peer link capacities. Therefore, the following do not take effect in weighted mode:

- The advertised value of the link bandwidth attribute will not be capped by the peer link-speed
- ‘neighbor <peer|peer-group> link-bandwidth auto’ will not take effect

Show Commands

The following control plane show commands have been enhanced to display the Link Bandwidth attribute and UCMP related state information:

- show ip[v6] bgp (<prefix> | <address> | detail)
- show ip[v6] bgp neighbors <address> (received-routes | routes | advertised-routes) (<prefix> | <address> | detail)
- show rib route ip[v6]
- show ip[v6] route
- show bgp instance

The BGP prefix entry show command “show ip bgp [<prefix> | <address> | detail]” shows all the paths for a route in the BGP RIB. If the route is tagged with the link bandwidth attribute, it displays the attribute value in Bytes per second along with the ASN of the originating peer. If UCMP is enabled locally and the prefix is UCMP-eligible, the output also prints the route as UCMP.

```
Arista(config-router-bgp)#show ip bgp 192.0.2.0
BGP routing table information for VRF default
Router identifier 0.0.4.1, local AS number 64503
BGP routing table entry for 192.0.2.0/24
 Paths: 2 available
```

```
  64501
    198.51.100.1 from 198.51.100.3 (0.0.1.1)
      Origin IGP, metric 0, localpref 100, IGP metric 0, weight 0, tag
 0
      R
eceived 01:21:
03 ago, valid, internal, ECMP head, ECMP, UCMP, best, ECMP contributor
      Extended Community: Link-Bandwidth-AS:64501:1.2 GBps
      Rx SAFI: Unicast
  64502
    198.51.100.2 from 198.51.100.4 (0.0.3.1)
      Origin IGP, metric 0, localpref 100, IGP metric 0, weight 0, tag
 0
      Received 01:
21:00 ago, valid, internal, ECMP, UCMP, ECMP contributor
      Extended Community: Link-Bandwidth-AS:64502:2.5 GBps
      Rx SAFI: Unicast
```

The “show rib route ip” command for a specific route displays RIB state information for that route. If UCMP is in effect for the route, this command also prints the corresponding link bandwidth values for each next hop.

```
Arista(config-router-bgp)#show rib route ip 192.0.2.0
VRF: default
Codes: C - Connected, S - Static, P - Route Input
       B - BGP, O - Ospf, O3 - Ospf3, I - Isis, R - Rip, VL - VRF Leak
       > - Best Route, * - Unresolved Next hop
       EM - Exact match of the SR-TE Policy
       NM - Null endpoint match of the SR-TE Policy
       AM - Any endpoint match of the SR-TE Policy
       L - Part of a recursive route resolution loop
       A - Next hop not resolved in ARP/ND
       NF - Not in FEC
>B    192.0.2.0/24 [200 pref/0 MED] updated 01:21:54 ago
         via 198.51.100.1
link bandwidth 1.2 GBps [1 pref/0 metric] type ipv4
            via 198.51.100.3, Ethernet2
         via 198.51.100.2 link bandwidth 2.5 GBps
 [1 pref/0 metric] type ipv4
            via 198.51.100.4, Ethernet3
```

The “show ip route” command for a specific route displays FIB state information for that route. If UCMP is in effect for the route and the ECMP next hops have unequal link bandwidth values, this command also prints the corresponding weight or integer repetition counts of those next hops in the FIB for the ratio in which UCMP will be done for that route.

```
Arista(config-router-bgp)#show ip route 192.0.2.0

VRF: default
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B - BGP, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked,
       RC - Route Cache Route

 B I      192.0.2.0/24 [200/0] via 198.51.100.3, Ethernet2, weight 1/3
                               via 198.51.100.4, Ethernet3, weight 2/3
```

The “show bgp instance” has been enhanced to display UCMP state information that is configurable for the BGP instance. The UCMP state being displayed consists of the mode of configuration, maximum UCMP set size, configured deviation value, and any UCMP link-bandwidth update-delay configuration.

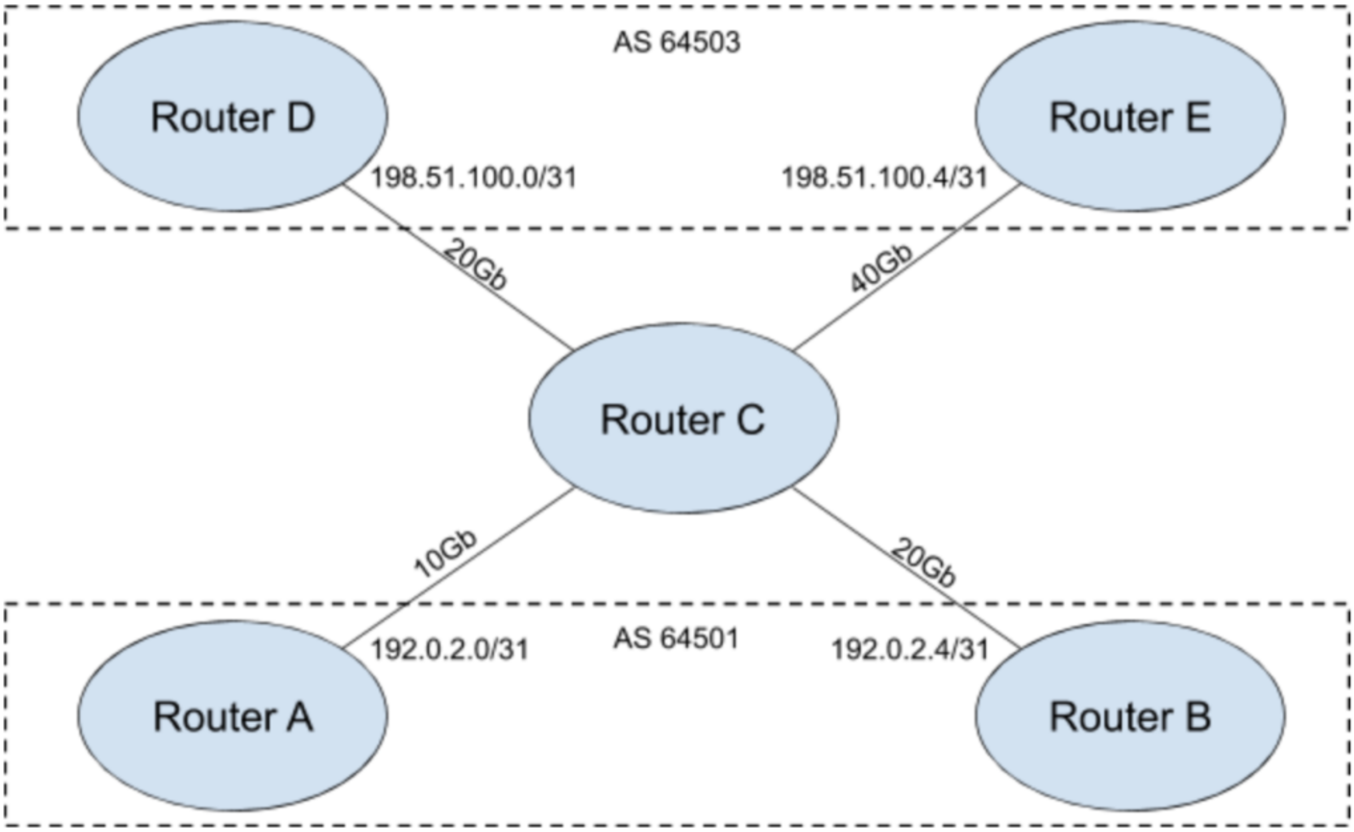
```
Arista#show bgp instance
BGP instance information for VRF default
BGP Local AS: 64503, Router ID: 0.0.4.1
...
UCMP mode: 1
   UCMP maximum paths: 64
   UCMP deviation: 0.010000
   UCMP link bandwidth delay: 00:00:00

...
```

Additionally, the following show commands have been enhanced to display the UCMP per-via weights at the platform level.

```
Arista#show ip|ipv6 hardware fib fec



Weight    Intf
----------------------------------------------------------------------
----------------



15        Po9



15        Po8



15        Et36



14        Et50/1



2         Et49/1



1         Po3



1         Po2



1         Et43
Arista#show ip|ipv6 hardware ale adj
      FecId            FecType       L2Adj    Prefixes   Next hop
 Weight   Intf
----------------------------------------------------------------------
----------------
    4294968748      'forwardRoute'   1509      3800      203.0.113.4
 1        Po3
```

```
                                                         203.0.113.0
 1        Et43
                                                         203.0.113.20
 15       Po8
                                                         203.0.113.6
 2        Et49/1
                                                         203.0.113.2
 1        Po2
                                                         203.0.113.16
 14       Et50/1
                                                         203.0.113.18
 15       Et36
                                                         203.0.113.22
 5        Po9

Arista#show ip|ipv6 hardware ale l2-adj
 L2Adj   Adjs   Weight
    Type                   MacAddr             Vlan   Intf
----------------------------------------------------------------------
----------------
 1511    1      4
         ‘nextHopArpResolved’   00:1c:73:bd:c5:25   1013   Et49/1
                10
        ‘nextHopArpResolved’   00:1c:73:bd:c7:af   1022   Po8
                3
         ‘nextHopArpResolved’   00:1c:73:bd:c5:25   1018   Po3
                2
         ‘nextHopArpResolved’   00:1c:73:bd:c5:25   1009   Po2
                10
        ‘nextHopArpResolved’   00:1c:73:bd:c7:af   1016   Po9
                10
        ‘nextHopArpResolved’   00:1c:73:bd:c7:af   1008   Et36
                1
         ‘nextHopArpResolved’   00:1c:73:bd:c5:25   1014   Et43
                10
        ‘nextHopArpResolved’   00:1c:73:bd:c7:af   1017   Et50/1

Arista#show platform trident l3 software next-hops unicast
ECMP Size Start Arp AdjIndex RefCount nexthops (Weights)
   1    8    13   N     4304     3000
      1096(  1) 1108(  1) 1098(  1) 1094(  1) 1110(  1
) 1097(  1) 1109(  1)  1107(  1)
   2    8     5   N     4287    22000
      1096(  1) 1108(  1) 1098(  1) 1094(  1  1110(  1
) 1097(  1) 1109(  1)  1107(  1)
```

Sample Topology



In the topology above, Router A and Router B will be advertising the same network 203.0.113.0/24 to Router C.

For the 203.0.113.0/24 route sent from Router A and Router B to Router C, link bandwidth will be added inbound on Router C via a route-map or in either of the following ways.

```
RouterC(config-router-bgp)#neighbor 192.0.2.0 link-bandwidth auto
RouterC(config-router-bgp)#neighbor 192.0.2.0 link-
bandwidth default 10G
RouterC(config-router-bgp)#neighbor 192.0.2.4 link-bandwidth auto
RouterC(config-router-bgp)#neighbor 192.0.2.4 link-
bandwidth default 20G
```

UCMP is enabled locally on Router C as shown.

```
RouterC(config-router-bgp)#ucmp mode 1 32 0.01
```

The weight assignment for the prefix 203.0.113.1 is shown in “show ip route” output.

```
RouterC#show ip route 203.0.113.1

VRF name: default
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - ISIS level 1, I L2 - ISIS level 2,
       A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service

 B E    203.0.113.0/24 [200/0] via 192.0.2.0, Ethernet1, weight 1/3
                               via 192.0.2.4, Ethernet2, weight 2/3
```

Router C can advertise the aggregate link bandwidth for prefix 203.0.113.0/24 to Router D and E.

```
RouterC(config-router-bgp)#neighbor 198.51.100.0 send-community link-
bandwidth aggregate
RouterC(config-router-bgp)#neighbor 198.51.100.4 send-community link-
bandwidth aggregate
```

Router D (198.51.100.0)’s link-capacity of 20Gb is less than the aggregate link capacity of 30Gb so Router C only advertises 20Gb to Router D. Note that the link-bandwidth value in “show ip bgp” is displayed in Bytes per second instead of bits per second.

```
RouterD#show ip bgp 203.0.113.1
BGP routing table information for VRF default
Router identifier 0.0.0.4, local AS number 64503
BGP routing table entry for 203.0.113.0/24
 Paths: 1 available
  64502 64501
    198.51.100.1 from 198.51.100.1 (0.0.0.3)
      Origin IGP, metric 0, localpref 100, weight 0, valid, external,
best
      Extended Community: Link-Bandwidth-AS:64501:2.5 GBps
```

Unlike Router D, Router E receives the total link bandwidth of 30Gb. This is because Router E’s peering link capacity is 40Gb, so the advertised link-bandwidth of 30Gb is not subject to capping.

```
RouterE#show ip bgp 203.0.113.1
BGP routing table information for VRF default
Router identifier 0.0.0.5, local AS number 64503
BGP routing table entry for 203.0.113.0/24
 Paths: 1 available
  64502 64501
    198.51.100.5 from 198.51.100.5 (0.0.0.3)
      Origin IGP, metric 0, localpref 100, weight 0, valid, external,
best
      Extended Community: Link-Bandwidth-AS:64501:3.8 GBps
```

The aggregate link bandwidth can also be divided equally between Router D and E if they are in the same Rib-out.

```
RouterC(config-router-bgp)#neighbor 198.51.100.0 send-community link-
bandwidth divide equal
RouterC(config-router-bgp)#neighbor 198.51.100.4 send-community link-
bandwidth divide equal
```

With the aggregate link bandwidth of 30Gb divided equally between two peers, each peer receives a link bandwidth of 15Gb.

```
RouterD#show ip bgp 203.0.113.1
BGP routing table information for VRF default
Router identifier 0.0.0.4, local AS number 64503
BGP routing table entry for 203.0.113.0/24
 Paths: 1 available
  64502 64501
    198.51.100.1 from 198.51.100.1 (0.0.0.3)
      Origin IGP, metric 0, localpref 100, weight 0, valid, external,
best
      Extended Community: Link-Bandwidth-AS:64501:1.9 GBps

RouterE#show ip bgp 203.0.113.1
```

```
BGP routing table information for VRF default
Router identifier 0.0.0.5, local AS number 64503
BGP routing table entry for 203.0.113.0/24
 Paths: 1 available
  64502 64501
    198.51.100.5 from 198.51.100.5 (0.0.0.3)
      Origin IGP, metric 0, localpref 100, weight 0, valid, external,
best
      Extended Community: Link-Bandwidth-AS:64501:1.9 GBps
```

The aggregate link bandwidth can also be shared based on the ratio of the peers’ link capacities if they are in the same Rib-out.

```
RouterC(config-router-bgp)#neighbor 198.51.100.0 send-community link-
bandwidth divide ratio
RouterC(config-router-bgp)#neighbor 198.51.100.4 send-community link-
bandwidth divide ratio
```

Router D with the 20Gb peering link capacity receives a 10Gb link bandwidth

```
RouterD#show ip bgp 203.0.113.1
BGP routing table information for VRF default
Router identifier 0.0.0.4, local AS number 64503
BGP routing table entry for 203.0.113.0/24
 Paths: 1 available
  64502 64501
    198.51.100.1 from 198.51.100.1 (0.0.0.1)
      Origin IGP, metric 0, localpref 100, weight 0, valid, external,
best
      Extended Community: Link-Bandwidth-AS:64501:1.2 GBps
```

while Router E with the 40Gb peering link capacity receives a 20Gb link bandwidth

```
RouterE#show ip bgp 203.0.113.1
BGP routing table information for VRF default
Router identifier 0.0.0.5, local AS number 64503
BGP routing table entry for 203.0.113.0/24
 Paths: 1 available
```

```
  64502 64501
    198.51.100.5 from 198.51.100.5 (0.0.0.2)
      Origin IGP, metric 0, localpref 100, weight 0, valid, external,
best
      Extended Community: Link-Bandwidth-AS:64501:2.5 GBps
```

Syslog Messages

No additional messages were added for this feature.

Limitations

Release Availability

This feature is available with the ribd routing protocol model since EOS-4.15.4F.

This feature is available with the multi-agent routing protocol model since EOS-4.24.1F.

Certain parts of this feature (listed below) are available with the multi-agent routing protocol model since EOS-4.22.1F.

- Enabling UCMP locally
- Enabling BGP Link Bandwidth attribute injecting inbound
- Enabling weighted mode for Link Bandwidth attribute
- Enabling BGP Link Bandwidth attribute matching using extended community list
    - Inbound direction only
- Enabling BGP Link Bandwidth attribute injection using route-ma
    - Inbound direction only

Notable exclusions and omissions

- Receiving, regenerating, and using the Link Bandwidth attribute is only supported for the IPv4 and IPv6 unicast address families
- UCMP is not supported for 6PE routes
- For a BGP route whose next hop resolves over another route or a tunnel, the Link Bandwidth attribute is not inherited from the resolving route or tunnel
- If the next hop of a received BGP route is modified by inbound policy, the Link Bandwidth attribute for the received BGP route cannot be used locally for UCMP
- ip extcommunity-list <NAME> permit lbw range <value> <value> is not supported

Resources

- BGP UCMP TOI for ribd routing protocol model
- Support for UCMP “adjust auto” (multi-agent) TOI
- IETF link bandwidth draft