BGP Load Sharing & Load Balancing

Load-Sharing vs Load-Balancing

Load balancing is an attempt to process traffic evenly across a network with multiple links. The reason this term is less preferred than load sharing is because it is difficult to achieve perfect load balancing. I can achieve load balancing (eg: with cef I could use per-packet, per-destination, per source-destination etc) it can be difficult to achieve true load balancing across each of the paths.

Load sharing is a term used when attempting to share some of the traffic across multiple links. A good example of load sharing is when having two devices connect using two links of different speed. Let's say link one is 9Mbit/s, and the other is 3Mbit/s. For every three packets we send through the 9Mbit link, we would want to send one packet down the 3Mbit/s link. The result is that the 9Mbit/s link would send a higher proportion of traffic than the 3Mbit/s link.

When talking about either term, load balancing or load sharing, we are talking about a unidirectional method of sharing data through paths to the destination. This is because network layer traffic (layer 3 IP traffic) is connectionless; meaning that each packet that is sent, is a completely independent piece of data. So return traffic can take a totally different path to which is was sent. If we want to load-balance or load-share back to the source, we would have to configure this separately

Normally BGP will have the best path in routing table but we can add more paths but the best path will still , this can be configured for iBGP or eBGP

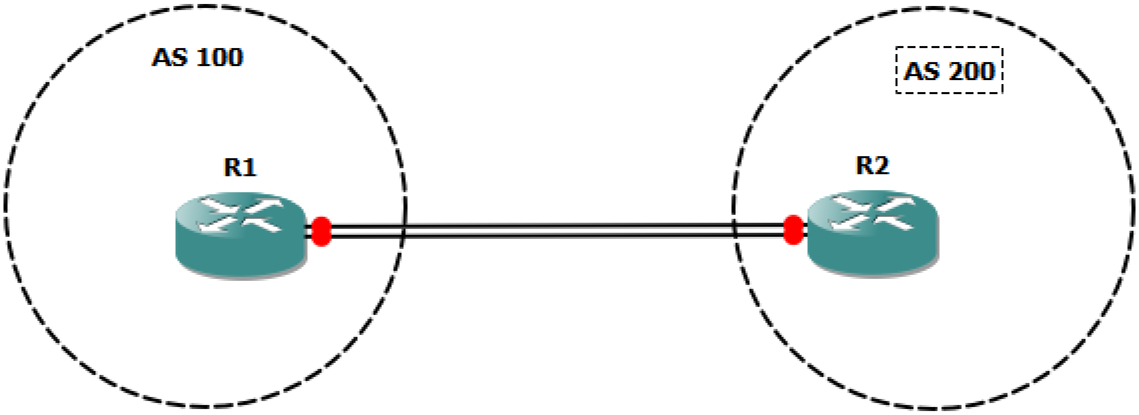
eBGP multipath :

maximum-paths 2
(both paths must have same attributes and belong to same As)
if both paths had equal cost , bgp will load balancing

iBGP multipath :

(both paths must have same attributes but next-hop must be different)
maximum-paths ibgp 2

Load Sharing & Load Balancing Lab 1



R1

```
int f0/0
ip add 10.12.12.1 255.255.255.0
no sh
int f0/1
ip add 10.122.122.1 255.255.255.0
no sh
int loop0
ip add 1.1.1.1 255.255.255.255
int loop1
ip add 111.111.111.111 255.255.255.255
router bgp 100
net 111.111.111.111 mask 255.255.255.255
nei 2.2.2.2 remote-as 200
nei 2.2.2.2 ebgp-multihop 2
nei 2.2.2.2 update-source Loopback0

ip route 2.2.2.2 255.255.255.255 FastEthernet0/1 10.122.122.2
ip route 2.2.2.2 255.255.255.255 FastEthernet0/0 10.12.12.2
```

R2

```
int f0/0
ip add 10.12.12.2 255.255.255.0
no sh
int f0/1
ip add 10.122.122.2 255.255.255.0
no sh
int loop0
ip add 2.2.2.2 255.255.255.255

router bgp 200
nei 1.1.1.1 remote-as 100
nei 1.1.1.1 ebgp-multihop 2
nei 1.1.1.1 update-source Loopback0

ip route 1.1.1.1 255.255.255.255 FastEthernet0/1 10.122.122.1
ip route 1.1.1.1 255.255.255.255 FastEthernet0/0 10.12.12.1
```

```
R2#sh ip route
     1.0.0.0/32 is subnetted, 1 subnets
S       1.1.1.1 [1/0] via 10.122.122.1, FastEthernet0/1
                [1/0] via 10.12.12.1, FastEthernet0/0
     2.0.0.0/32 is subnetted, 1 subnets
C       2.2.2.2 is directly connected, Loopback0
     111.0.0.0/32 is subnetted, 1 subnets
B       111.111.111.111 [20/0] via 1.1.1.1, 00:00:36
     10.0.0.0/24 is subnetted, 2 subnets
C       10.122.122.0 is directly connected, FastEthernet0/1
C       10.12.12.0 is directly connected, FastEthernet0/0
```

```
R2#traceroute 111.111.111.111
Type escape sequence to abort.
Tracing the route to 111.111.111.111

  1 10.122.122.1 36 msec 36 msec *
```

R2

```
router bgp 200
maximum-paths 2    ( do not forget to type clear ip bgp *)
```

```
R2#traceroute 111.111.111.111
Type escape sequence to abort.
Tracing the route to 111.111.111.111

  1 10.12.12.1 60 msec
    10.122.122.1 36 msec
    10.12.12.1 32 msec
```

To verify :

```
R2(config)#int f0/0
R2(config-if)#ip load-sharing per-packet
R2(config-if)#int f0/1
R2(config-if)#ip load-sharing per-packet
```

```
R2#sh ip cef exact-route 10.12.12.2 111.111.111.111
10.12.12.2      -> 111.111.111.111: FastEthernet0/1 (next hop 10.122.122.1)
R2#sh ip cef exact-route 10.12.12.2 111.111.111.111
10.12.12.2      -> 111.111.111.111: FastEthernet0/0 (next hop 10.12.12.1)
R2#sh ip cef exact-route 10.12.12.2 111.111.111.111
10.12.12.2      -> 111.111.111.111: FastEthernet0/1 (next hop 10.122.122.1)
R2#sh ip cef exact-route 10.12.12.2 111.111.111.111
10.12.12.2      -> 111.111.111.111: FastEthernet0/0 (next hop 10.12.12.1)
```

What makes BGP so special for unequal-cost load balancing?
Nothing else but the routing-loop detection feature implemented for eBGP session. When a BGP speakers receives a route from the external AS, it looks for its own AS# in the AS_PATH attribute, and discards matching routes. This prevents routing loops on AS-scale. Additionally, this allows BGP to use alternative eBGP paths for unequal-cost load balancing. The proportions for the alternative paths are chosen based on the special BGP extended community attribute called DMZ Link bandwidth. By default, this attribute value is copied from the bandwidth of the interface connecting to the eBGP peer

```
router bgp 100
 bgp maximum-path 3
 bgp dmzlink-bw
 neighbor 1.1.1.1 remote-as 200
 neighbor 1.1.1.1 dmzlink-bw
 neighbor 2.2.2.2 remote-as 200
 neighbor 2.2.2.2 dmzlink-bw
```

In order for paths to be eligible for unequal-cost load balancing, they must have the same weight, local-preference, AS_PATH length, Origin and MED. Then, the local speaker may utilize the paths inversely proportional to the value of DMZ Link bandwidth attribute. Keep in mind that BGP multipathing is disabled by default, until you enable it with the command bgp maximum path. IBGP speakers may use DMZ Link bandwidth feature as well, for the paths injected into the local AS via eBGP. In order for this to work, DMZ Link Bandwidth attribute must be propagated across the local AS (send-community extended command) and the exit points for every path must have equal IGP costs in the iBGP speaker's RIB. The data-plane implementation remains the same as for EIGRP multipathing, as CEF is the same underlying switching method.

What is BGP Link Bandwidth ?
it is feature used to advertise the bandwidth of an autonomous system exit link as an extended community . This feature is configured for links between directly connected external BGP (eBGP) neighbors. The link bandwidth extended community attribute is propagated to iBGP peers when extended community exchange is enabled. This feature is used with BGP multipath features to configure load balancing over links with unequal bandwidth.

NOTE : The **bgp dmzlink-bw** command is used to configure BGP to distribute traffic proportionally to the bandwidth of external links. This command is configured for multipath load balancing between directly connected external BGP (eBGP) neighbors. This command is used with BGP multipath features to configure load balancing over links with unequal bandwidth. The neighbor dmzlink-bw command must also be configured for each external link through which multipath load balancing is configured to advertise the link bandwidth as an extended community. The neighbor send-community command must be configured to exchange the link bandwidth extended community with internal BGP (iBGP) peers.

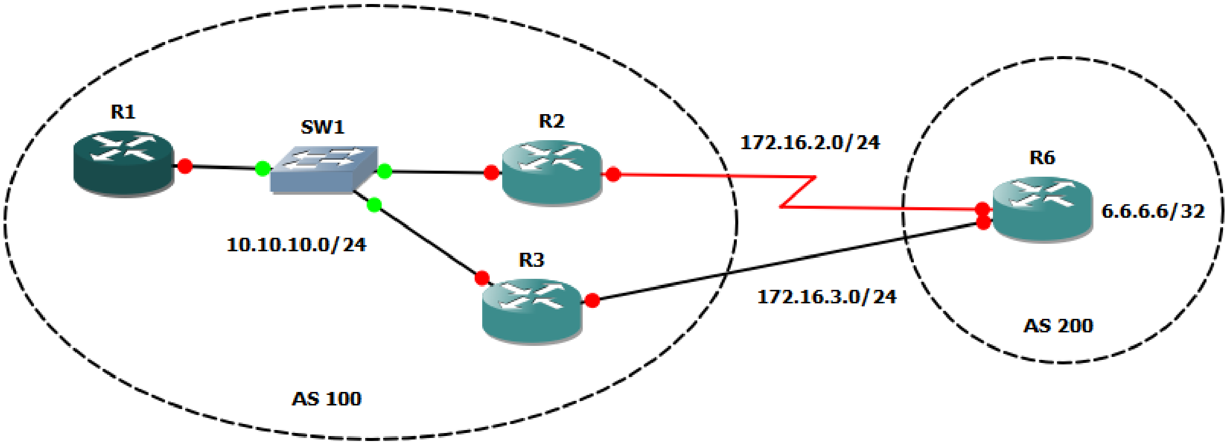
<u>Prerequisites for BGP Link Bandwidth</u>

- BGP load balancing or multipath load balancing must be configured before BGP Link Bandwidth feature is enabled
- BGP extended community exchange must be enabled between iBGP neighbors to which the link bandwidth attribute is to be advertised.
- CEF or distributed CEF must be enabled on all participating routers.

<u>Restrictions for BGP Link Bandwidth</u>

- The BGP Link Bandwidth feature can be configured only under IPv4 and VPNv4 address family sessions.
- BGP can originate the link bandwidth community only for directly connected links to eBGP neighbors

Load Sharing & Load Balancing Lab 2



R1
int f0/0
ip add 10.10.10.1 255.255.255.0
int loop 0
ip add 1.1.1.1 255.255.255.255

R2
int f0/0
ip add 10.10.10.2 255.255.255.0
int s0/0
ip add 172.16.2.1 255.255.255.0
int loop 0
ip add 2.2.2.2 255.255.255.0

R3
int f0/0
ip add 10.10.10.3 255.255.255.0
int f0/1
ip add 172.16.3.1 255.255.255.0
int loop 0
ip add 3.3.3.3 255.255.255.0

R4
int s0/0
ip add 172.16.2.2 255.255.255.0
int f0/0
ip add 172.16.3.2 255.255.255.0
int loop 0
ip add 6.6.6.6 255.255.255.255

R1

```
 router bgp 100
nei 10.10.10.2 remote-as 100
nei 10.10.10.3 remote-as 100
bgp dmzlink-bw
nei 10.10.10.2 send-community both
nei 10.10.10.3 send-community both
maximum-paths ibgp 4
```

R2

```
router bgp 100
nei 172.16.2.2 remote-as 200
nei 10.10.10.1 remote-as 100
nei 10.10.10.3 remote-as 100
bgp dmzlink-bw
nei 10.10.10.2 send-community both
nei 10.10.10.3 send-community both
nei 10.10.10.1 next-hop-self
nei 10.10.10.3 next-hop-self
maximum-paths ibgp 4
maximum-paths 4
nei 172.16.2.2 dmzlink-bw
```

Notice :

```
R2(config-router)#nei 10.10.10.3 dmzlink-bw
%BGP: Propagation of DMZ-Link-Bandwidth is supported only for single-hop EBGP peers
```

R3

```
router bgp 100
nei 172.16.3.2 remote-as 200
nei 10.10.10.2 remote-as 100
nei 10.10.10.1 remote-as 100
bgp dmzlink-bw
nei 10.10.10.2 send-community both
nei 10.10.10.1 send-community both
nei 10.10.10.1 next-hop-self
nei 10.10.10.2 next-hop-self
nei 172.16.3.2 dmzlink-bw
maximum-paths ibgp 4
maximum-paths 4
```

R6

```
router bgp 200
 network 6.6.6.6 mask 255.255.255.255
 neighbor 172.16.2.1 remote-as 100
 neighbor 172.16.3.1 remote-as 100
```

```
R2#sh ip bgp 6.6.6.6
BGP routing table entry for 6.6.6.6/32, version 2
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Multipath: eBGP iBGP
Flag: 0x820
  Advertised to update-groups:
        1    2
  200
    10.10.10.3 from 10.10.10.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal
      DMZ-Link Bw 1250 kbytes
  200
    172.16.2.2 from 172.16.2.2 (6.6.6.6)
      Origin IGP, metric 0, localpref 100, valid, external, best
      DMZ-Link Bw 193 kbytes
```

Good Luck
CCIE , CCSI: Yasser Auda
https://www.facebook.com/YasserRamzyAuda
https://learningnetwork.cisco.com/people/yasserramzy/content
https://www.youtube.com/user/yasserramzyauda