

CISCO Router Load-Balancing Discussion

When a router learns multiple routes to a specific network via multiple routing processes (or routing protocols, such as RIP, RIPv2, IGRP, EIGRP, and OSPF), it installs the route with the lowest administrative distance in the routing table.

Sometimes the router must select a route from among many learned via the same routing process with the same administrative distance. In this case, the router chooses the path with the lowest cost (or metric) to the destination. Each routing process calculates its cost differently and the costs may need to be manipulated in order to achieve load-balancing.

If the router receives and installs multiple paths with the same administrative distance and cost to a destination, load-balancing can occur. The number of paths used is limited by the number of entries the routing protocol puts in the routing table. Four entries is the default in IOS for most IP routing protocols with the exception of Border Gateway Protocol (BGP), where one entry is the default. Six different paths configured is the maximum number.

EIGRP puts up to four routes of equal cost in the routing table, which the router then load-balances. The type of load balancing (per packet or per destination) depends on the type of switching being done in the router. EIGRP, however, can also load-balance over unequal cost links.

Note: Using **max-paths**, you can configure EIGRP to use up to six routes of equal cost.

Let us say there are four paths to a given destination, and the metrics for these paths are:

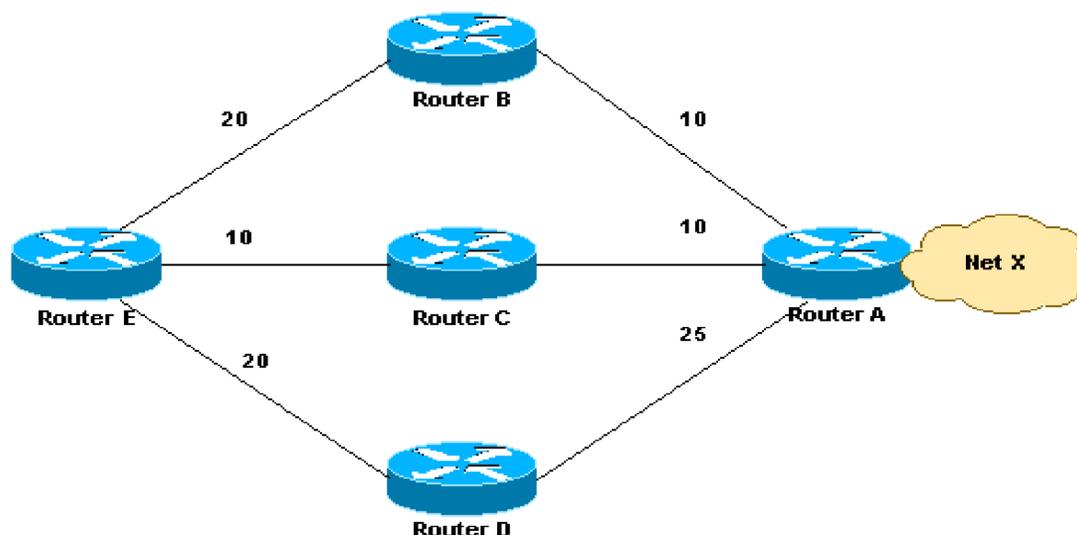
- path 1: 1100
- path 2: 1100
- path 3: 2000
- path 4: 4000

The router, by default, places traffic on both path 1 and 2. Using EIGRP, you can use the **variance** command to instruct the router to also place traffic on paths 3 and 4. The variance is a multiplier: traffic will be placed on any link that has a metric less than the best path multiplied by the variance. To load balance over paths 1, 2, and 3, use variance 2, because $1100 \times 2 = 2200$, which is greater than the metric through path 3. Similarly, to also add path 4, issue variance 4 under the **router eigrp** command.

Every routing protocol supports equal cost path load balancing. ***In addition to that, IGRP and EIGRP also support unequal cost path load balancing.***

The variable **n** can take a value between 1 and 128, with the default being 1, which means equal cost load balancing. Traffic is also distributed among the links with unequal costs, proportionately, with respect to the metric. **Note:** If a path is not a feasible successor, then the path is not used in load balancing.

Network Diagram



In the network diagram above, there are three ways to get to Network X:

- E-B-A with a metric of 30
- E-C-A with a metric of 20
- E-D-A with a metric of 45

Router E chooses the second path above, E-C-A with a metric of 20, because 20 is better than 30 and 45. The path, E-B-A, is also installed as a feasible path in the topology table for router E. To instruct EIGRP to select the path E-B-A as well, configure variance with a multiplier of 2:

```
router eigrp 1
 network x.x.x.x
 variance 2
```

The variance number, 2 in the case, simply doubles the metric value of its best route, 20, so that the new standard for moving a route into the routing table is 40, instead of 20. Now EIGRP will put any routes whose metrics are 40 or less in its routing table because these routes now satisfy the new variance enabled feasibility condition.

In the above configuration, EIGRP now uses two paths to get to Network X, E-C-A and E-B-A, because both paths have a metric under 40. EIGRP does not use path E-D-A because that path has a metric of 45, which is not less than the value of minimum metric of 40, due to the variance configured. Also, the reported distance of neighbor D is 25, which is greater than the feasible distance (FD) of 20 through C. This means that even if variance is set to 3, the E-D-A path will not be selected for load balancing since Router D is not a feasible successor.

Traffic Sharing with EIGRP

EIGRP provides both unequal cost path load balancing and intelligent load balancing, such as traffic sharing. The traffic-share command can control how traffic is distributed among routes when there are multiple routes for the same destination network that have different costs. With the keyword **balanced**, the router distributes traffic proportionately to the ratios of the metrics associated with different routes. This is the default setting:

```
router eigrp 1
  network x.x.x.x
  variance 2
  traffic-share balanced
```

Let us look at the traffic share count for this example:

- For path E-C-A: $30/20 = 3/2 = 1$
- For path E-B-A: $30/30 = 1$

Since the ratio is not an integer, we round down to the nearest integer. In this example EIGRP sends one packet to E-C-A and one packet to E-B-A.

Now, let us assume that the metric between E-B is 25 and B-A is 15. The E-B-A metric would be 40, but this path will not be selected for load balancing because the cost of this path, 40, is not less than $(20 * 2)$, where 20 is the FD and 2 is the variance. To include this path also in load sharing, the variance should be changed to 3. And the traffic share count ratio would be:

- For path E-C-A: $40/20 = 2$
- For path E-B-A: $40/40 = 1$

In this situation EIGRP sends two packets to E-C-A and one packet to E-B-A. In this way, EIGRP not only provides unequal cost path load balancing, but also intelligent load balancing.

Similarly, when the keyword **min** is used, the traffic is sent only across the minimum-cost path, even when there are multiple paths in the routing table.

```
router eigrp 1
  network x.x.x.x
  variance 3
  traffic-share min across-interfaces
```

In this situation, EIGRP sends packets only through E-C-A, which is the best path to the destination network. This is identical to the forwarding behavior if the **variance** command is not used. However, if you use the **traffic-share min** command and the **variance** command, even though traffic is sent over the minimum-cost path only, all feasible routes get installed into the routing table, which decreases convergence times. We have seen how to configure unequal cost path load balancing in EIGRP. Similarly, the same can be done in IGRP, except for the feasibility condition, which is not applicable to IGRP.

Per-Destination and Per-Packet Load Balancing

You can set load-balancing to work per-destination or per-packet. Per-destination load balancing means the router distributes the packets based on the destination address. Given two paths to the same network, all packets for destination1 on that network go over the first path, all packets for destination2 on that network go over the second path, and so on. This preserves packet order, with potential unequal usage of the links. If one host receives the majority of the traffic all packets use one link, which leaves bandwidth on other links unused. A larger number of destination addresses leads to more equally used links. To achieve more equally used links use IOS software to build a route-cache entry for every destination address, instead of every destination network, as is the case when only a single path exists. Therefore traffic for different hosts on the same destination network can use different paths. The downside of this approach is that for core backbone routers carrying traffic for thousands of destination hosts, memory and processing requirements for maintaining the cache become very demanding.

Per-packet load-balancing means that the router sends one packet for destination1 over the first path, the second packet for (the same) destination1 over the second path, and so on. Per-packet load balancing guarantees equal load across all links. However, there is potential that the packets may arrive out of order at the destination because differential delay may exist within the network. In Cisco IOS software, except the release 11.1CC, per packet load balancing does disable the forwarding acceleration by a route cache, because the route cache information includes the outgoing interface. For per-packet load balancing, the forwarding process determines the outgoing interface for each packet by looking up the route table and picking the least used interface. This ensures equal utilization of the links, but is a processor intensive task and impacts the overall forwarding performance. This form of per-packet load balancing is not well suited for higher speed interfaces.

Per-destination or per-packet load-balancing depends on the type of switching scheme used for IP packets. By default, on most Cisco routers, fast switching is enabled under interfaces. This is a demand caching scheme that does per-destination load-balancing. To set per-packet load-balancing, enable process switching (or disable fast switching), use these commands:

```
Router# config t
Router(config)# interface Ethernet 0
Router(config-if)# no ip route-cache
Router(config-if)# ^Z
```

Now the router CPU looks at every single packet and load balances on the number of routes in the routing table for the destination. This can crash a low-end router because the CPU must do all the processing. To re-enable fast switching, use these commands:

```
Router# config t
Router(config)# interface Ethernet 0
Router(config-if)# ip route-cache
Router(config-if)# ^Z
```