Title:
Cisco CCNA Certification Exam Tutorial: Variance And Unequal Cost Load Balancing

Word Count:
493

Summary:
Using the variance command with IGRP can be tricky! Learn how to use this command and a vital debug from Chris Bryant, CCIE #12933.

Keywords:
Ccna, exam, pass, free, igrp, eigrp, load, balancing, equal, unequal, variance, metric, debug, ip, transactions, routing, switching, Bryant, advantage, 12933

Article Body:
To pass the CCNA exam, you've got to know how to work with IGRP and EIGRP unequal-cost load balancing. You may not see much IGRP in production networks anymore, but you'll see a lot of EIGRP, and part of fine-tuning your EIGRP network is making sure that all paths are in use while allowing for varying bandwidth rates.

Using the variance command is the easy part - it's getting the metric that's the hard part with IGRP. With EIGRP, you just look in the topology table and that's it. With IGRP, you've got to run a debug to get the right metric.

The variance command is a multiplier when the value supplied with the variance command is multiplied by the lowest-cost metric, it must exceed the higher-cost metric in order for the higher-cost route to be added.

If that sounds complicated, it's not. It's one of those things that sounds difficult, but isn't. Trust me!

In this example, R1 has two paths to 172.23.0.0, but is currently using only one. By looking in the IP routing table, we've seen that the lowest-cost metric for network 172.23.0.0 on R1 is 8576. This path goes through the 172.12.123.0 network. There is another valid path that uses the 172.12.13.0 network, but is not currently in use.

I 172.23.0.0/16 [100/8576] via 172.12.123.2, 00:00:53, Serial0

IGRP does not have a “show" command that displays all valid routes to a

destination, as does EIGRP. The command debug ip igrp transactions will show the current metric of the routes using the 512 KBPS route.

```
R1#debug ip igrp transactions

IGRP protocol debugging is on

19:17:51: IGRP: broadcasting request on Loopback0

19:17:51: IGRP: broadcasting request on Serial0

19:17:51: IGRP: broadcasting request on Serial1

19:17:51: IGRP: received update from 172.12.13.3 on Serial1

19:17:51: subnet 172.12.13.0, metric 23531 (neighbor 21531)

19:17:51: subnet 172.12.123.0, metric 23531 (neighbor 8476)

19:17:51: network 1.0.0.0, metric 24031 (neighbor 8976)

19:17:51: network 2.0.0.0, metric 22131 (neighbor 1600)

19:17:51: network 3.0.0.0, metric 22031 (neighbor 501)

19:17:51: network 172.23.0.0, metric 21631 (neighbor 1100)

R1(config)#router igrp 1
R1(config-router)#variance 3
R1#show ip route 172.23.0.0
Routing entry for 172.23.0.0/16
Known via "igrp 1", distance 100, metric 8576
Redistributing via igrp 1
Advertised by igrp 1 (self originated)
Last update from 172.12.123.2 on Serial0, 00:00:01 ago
Routing Descriptor Blocks:
* 172.12.13.3, from 172.12.13.3, 00:00:20 ago, via Serial1
Route metric is 21631, traffic share count is 1
Total delay is 21000 microseconds, minimum bandwidth is 512 Kbit
Reliability 255/255, minimum MTU 1500 bytes
Loading 1/255, Hops 0
172.12.123.3, from 172.12.123.3, 00:00:20 ago, via Serial0
Route metric is 8576, traffic share count is 3
```

```
Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
Reliability 255/255, minimum MTU 1500 bytes
Loading 1/255, Hops 0
172.12.123.2, from 172.12.123.2, 00:00:01 ago, via Serial0
Route metric is 8576, traffic share count is 3
Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
Reliability 255/255, minimum MTU 1500 bytes
Loading 1/255, Hops 0
```

The metric for 172.23.0.0 through the direct connection is 21631. A variance of 3 means that any route with a metric less than the best metric multiplied by the variance (in this case, 8576 x 3 = 25728) will be entered into the routing table. R1 now has three unequal-cost paths to 172.23.0.0 in its routing table, and load balancing will take place.

IGRP unequal-cost load balancing takes some practice, but as you can see, once you get the metric it's easy to work with. Just make sure you know how to get that metric!