Title:
Cisco CCNA Certification Exam Tutorial: Distance Vector Command Review

Word Count:
602

Summary:
To earn your CCNA, you've got to keep a lot of commands straight! In this command review, see many of these commands defined and in action from Chris Bryant, CCIE #12933.

Keywords:
ccna, certification, exam, distance, vector, command, meaning, review, ip, route, static, routing

Article Body:
Part of studying for CCNA exam success is keeping all these new commands straight in your head! And let's face it, there are a lot of commands you need to know in order to pass the CCNA exam and earn that certification. Here's a review of some very important distance vector and static routing commands you need to know, along with their proper usage and console output.

Bandwidth
IGRP makes a default assumption that any Serial interface running IGRP is connected to a T1 line, which runs at 1544 KBPS. With equal-cost load-balancing enabled by default, this may be an undesirable assumption.

To alter IGRP’s assumption, use the bandwidth command on the serial interface in question. Note that this command does NOT actually affect the bandwidth available to the interface; it merely changes IGRP’s assumption of the bandwidth.

```
R2#conf t

R2(config)#int s0

R2(config-if)#bandwidth 512
```

Clear ip route *

This command clears your routing table of all non-static and non-connected routes. In a lab environment, it’s very handy because it forces your routers running routing protocols to send and request updates, rather than waiting for the regularly scheduled updates.

```
R2#clear ip route *
```

Debug ip igrp events

Debug ip igrp events allows you to see IGRP updates being sent and requested. Here, the debug is run and then the routing table is cleared. The router immediately broadcasts update requests via the IGRP-enabled interfaces.

```
R2#debug ip igrp event

IGRP event debugging is on

R2#clear ip route *

06:02:51: IGRP: broadcasting request on BRI0

06:02:51: IGRP: broadcasting request on Serial0.123
```

Debug ip igrp transactions

To configure IGRP unequal-cost load-sharing with the variance command, you’ve got to know the metric of the less-desirable routes. EIGRP keeps these in its topology table; IGRP has no such table.

To get the metrics of routes not in the routing table, run debug ip igrp transactions. To force IGRP updates, the routing table below was cleared with clear ip route *.

```
R2#debug ip igrp transactions


IGRP protocol debugging is on

R2#clear ip route *
```

```
06:05:33: IGRP: received update from 172.12.123.1 on Serial0.123

06:05:33: subnet 172.12.123.0, metric 10476 (neighbor 8476)

06:05:33: network 1.0.0.0, metric 8976 (neighbor 501)

06:05:33: IGRP: edition is now 3

06:05:33: IGRP: sending update to 255.255.255.255 via BRI0 (172.12.12.2)

06:05:33: network 1.0.0.0, metric=8976

06:05:33: IGRP: sending update to 255.255.255.255 via Serial0.123 (172.12.123.2)
- suppressing null update

06:05:34: IGRP: received update from 172.12.12.1 on BRI0

06:05:34: subnet 172.12.13.0, metric 160250 (neighbor 8476)

06:05:34: network 1.0.0.0, metric 158750 (neighbor 501)
```

Debug ip rip

```
R2#debug ip rip

IP protocol debugging is on

R2#clear ip route *

6:14:53: RIP: received v2 update from 172.23.23.3 on Ethernet0

6:14:53: 1.0.0.0/8 via 0.0.0.0 in 16 hops (inaccessible)

6:14:53: 1.1.1.1/32 via 0.0.0.0 in 2 hops

6:14:53: 172.12.0.0/16 via 0.0.0.0 in 16 hops (inaccessible)

6:14:53: 172.12.12.2/32 via 0.0.0.0 in 2 hops

6:14:53: 172.12.13.0/30 via 0.0.0.0 in 1 hops
```

```
6:14:53: 172.12.123.0/24 via 0.0.0.0 in 1 hops

6:14:53: 172.23.0.0/16 via 0.0.0.0 in 16 hops (inaccessible)
```

Run debug ip rip to troubleshoot routing update problems, RIP authentication problems, and to view the routing update contents. Clear ip route * was run to clear the routing table and to force a RIP update.

Ip route

```
R2#conf t

R2(config)#ip route 1.1.1.1 255.255.255.255 172.12.123.1
```

OR

```
R2(config)#ip route 1.1.1.1 255.255.255.255 serial0
```

To configure a static route to a given destination IP address, use the ip route command. The destination is followed by a subnet mask, and that can be followed by either the next-hop IP address or the exit interface on the local router.

Ip route 0.0.0.0 0.0.0.0

```
R2#conf t

R2(config)#ip route 0.0.0.0 0.0.0.0 172.12.123.1
```

OR

```
R2(config)#ip route 0.0.0.0 0.0.0.0 ethernet0
```

To configure a default static route, use either of these two commands.

You could have any number for the first “0.0.0.0", since the second set of zeroes is the subnet mask. This means that any destination will match this route statement.

That's a good review to get started with! I'll be back tomorrow with Part II of this CCNA exam command review!