CONFIGURATION

R1 — Gi 0/0 — .1 — 10.12.0.0 /30 — Gi 0/0 — .2 — R2 — Lo 0 — 10.2.0.1 /24

R1	
1	R1(config)# **router eigrp 42**
2	R1(config-router)# **eigrp router-id 0.0.0.1**
3	*Optional. Notice that it requires the "eigrp" keyword even though it's in the eigrp block.*
4	*The OSPF command was just "router-id…" IOS may fix this for you.*
5	R1(config-router)# **network 10.12.0.1 0.0.0.0**
6	*This wildcard allows me to pick just the one interface with that exact address*
7	*Notice there's no "area," like in OSPF. That concept is handled by the Autonomous System*
8	*number, 42, at the top.*

R2	
9	R2(config)# **router eigrp 42**
10	*The AS number, 42, needs to match R1 and any other routers that are going to cooperate.*
11	R2(config-router)# **eigrp router-id 0.0.0.2**
12	R2(config-router)# **network 10.0.0.0 0.255.255.255**
13	*This wildcard will enable EIGRP on all interfaces in the 10.0.0.0 /8 network*
14	*Actually, if you want the classful network, the wildcard is optional.*
15	Jan 2 12:21:02.639: %DUAL-5-NBRCHANGE: EIGRP-IPv4 42: Neighbor 10.12.0.1
16	(GigabitEthernet0/0) is up: new adjacency

VERIFICATION

Show IP EIGRP Interfaces [detail] [Gi0/0]

This tells you which interfaces were ensnared by the network commands and whether each has any neighbors ("peers"). In EIGRP, timer settings (hello interval, etc.) are part of the hello message, so they can vary from neighbor to neighbor. Those timers are shown when you add "detail" to the command line. Both the normal and detailed versions of the command can be limited to just one interface by appending it. Both versions skip interfaces where EIGRP is passive or not enabled.

```
 1 R2# show ip eigrp interfaces
 2 EIGRP-IPv4 Interfaces for AS(42)
 3                        Xmit Queue    Mean   Pacing Time   Multicast    Pending
 4 Interface       Peers  Un/Reliable   SRTT   Un/Reliable   Flow Timer   Routes
 5 Gi0/0             1       0/0          1        0/1           50          0
 6 Lo0               0       0/0          0        0/1            0          0
 7
 8 R2# show ip eigrp interfaces detail Gi0/0
 9 EIGRP-IPv4 Interfaces for AS(42)
10                        Xmit Queue    Mean   Pacing Time   Multicast    Pending
11 Interface       Peers  Un/Reliable   SRTT   Un/Reliable   Flow Timer   Routes
12 Gi0/0             1       0/0          1        0/1           50          0
13   Hello-interval is 5, Hold-time is 15
14   Split-horizon is enabled
15                 Rest omitted
```

Show IP Protocols

This repeats the configuration, perfect if "show running-config" is unavailable on a test. On line 23, if we had used a non-classful wildcard for the 10.0.0.0 network, you'd see that in CIDR notation, e.g. 10.0.0.0 /30 for the wildcard 0.0.0.3 (netmask 255.255.255.252). "Routing Information Sources" only lists neighbors who give us routes, so in our example, R1 lists R2, but R2 doesn't list R1 (look at the topology diagram on page 1; R1 has no routes that R2 doesn't already know).

```
 1 R1# show ip protocols
 2 *** IP Routing is NSF aware ***
 3
 4 Routing Protocol is "eigrp 42"
 5   Outgoing update filter list for all interfaces is not set
 6   Incoming update filter list for all interfaces is not set
 7   Default networks flagged in outgoing updates
 8   Default networks accepted from incoming updates
 9   EIGRP-IPv4 Protocol for AS(42)
10     Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
11     NSF-aware route hold timer is 240
12     Router-ID: 0.0.0.1
13     Topology : 0 (base)
14       Active Timer: 3 min
15       Distance: internal 90 external 170
16       Maximum path: 4
17       Maximum hopcount 100
18       Maximum metric variance 1
19
20   Automatic Summarization: disabled
21   Maximum path: 4
22   Routing for Networks:
23     10.12.0.1/32
24   Routing Information Sources:
25     Gateway         Distance      Last Update
26     10.12.0.2            90        01:52:54
     Distance: internal 90 external 170
```

Show IP EIGRP Neighbors

The address column is literally the IP address of the neighbor's interface (OSPF showed the router ID). The interface is ours. The "Hold" column is the countdown until we consider a hello overdue from that neighbor. Since we're using default timings (5 second hellos on Ethernet with a 3x hold timer), this shouldn't go lower than 10 before a hello is received and it jumps back to 15.

```
 1 R1# show ip eigrp neighbors
 2 EIGRP-IPv4 Neighbors for AS(42)
 3 H   Address                 Interface       Hold Uptime    SRTT   RTO   Q   Seq
 4                                             (sec)          (ms)         Cnt Num
 5 0   10.12.0.2               Gi0/0             10 01:44:54 1596   5000  0   2
```

Show IP Route [EIGRP]

The "D" in the left column (DUAL) and the administrative distance of 90 identify EIGRP routes.

```
 1 R1# show ip route eigrp
 2 Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
 3        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
 4                Rest of legend and gateway of last resort omitted
 5        10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
 6 D        10.2.0.0/24 [90/130816] via 10.12.0.2, 07:57:13, GigabitEthernet0/0
```

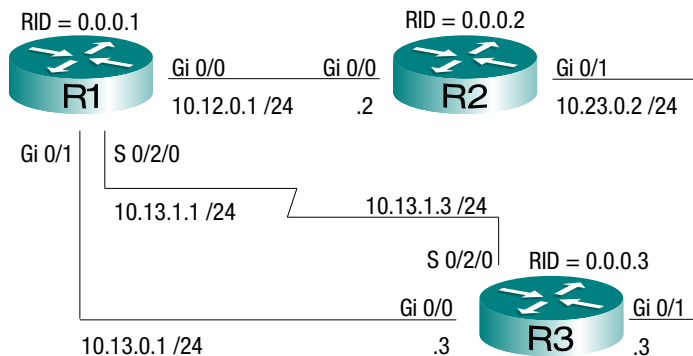| Show CMD	AS	Router ID	Networks	Interfaces	Passives	Neighbors
show ip protocols	•	•	•			some
show ip eigrp interface	•				•	count
show ip eigrp neighbors	•					•

THE TOPOLOGY TABLE

OSPF basically stores the topology of an entire area on each router. An EIGRP router stores only its own routes to each possible next hop, each route's metric, and the metric that route received on its next-hop router.

Terminology Recap

Successor Route	The chosen best route to a destination, offered to the routing table
Feasible Distance (FD)	The successor route's metric. Equals the distance to the next hop neighbor + the neighbor's reported distance
Reported Distance (RD)	A neighbor's metric for a route as reported to us. AKA the Advertised Distance (AD)
Feasible Successor	A non-best route that is immediately usable to replace a lost successor. It has an RD < the FD, ensuring that it won't create a routing loop.

RID = 0.0.0.1

R1

Gi 0/0   Gi 0/0   R2   Gi 0/1
10.12.0.1 /24   .2   10.23.0.2 /24

Gi 0/1   S 0/2/0

10.13.1.1 /24   10.13.1.3 /24

S 0/2/0   RID = 0.0.0.3

Gi 0/0   R3   Gi 0/1
10.13.0.1 /24   .3   .3

RID = 0.0.0.2

Show IP EIGRP Topology

Organized by subnet, like the routing table, this output shows EIGRP routes to each known subnet, even if they weren't chosen for inclusion in the routing table. For example, 10.12.0.0, 10.13.0.0, and 10.13.1.0 were offered to R1's routing table with EIGRP's administrative distance of 90 and ignored because they were also offered to the routing table as directly connected (AD = 0, see chapter 7a).

```
 1 R1# show ip eigrp topology
 2 EIGRP-IPv4 Topology Table for AS(42)/ID(0.0.0.1)
 3 Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
 4        r - reply Status, s - sia Status
 5
 6 P 10.13.0.0/24, 1 successors, FD is 2816
 7         via Connected, GigabitEthernet0/1
 8 P 10.23.0.0/24, 2 successors, FD is 3072
 9         via 10.12.0.2 (3072/2816), GigabitEthernet0/0
10         via 10.13.0.3 (3072/2816), GigabitEthernet0/1
11         via 10.13.1.3 (40512256/2816), Serial0/2/0
12 P 10.13.1.0/24, 1 successors, FD is 40512000
13         via Connected, Serial0/2/0
14 P 10.12.0.0/24, 1 successors, FD is 2816
15         via Connected, GigabitEthernet0/0
```

10.23.0.0 is more interesting. Both 10.12.0.2 and 10.13.0.3 [Lines 9-10] were chosen as successor routes and offered to the routing table because their cost (feasible distance) of 3072 was lowest. They were placed in the routing table because there were no other offers with a lower AD. Like OSPF, EIGRP will offer up to 4 equal-cost routes to the routing table by default and the router will load-balance between them. Their metric became the feasible distance for that destination and is shown at the end of line 8. Successor routes always have a metric equal to the feasible distance. The second number in parentheses on lines 9 and 10 is the reported distance for those routes—n this case, the costs reported by R2 and R3 to reach the directly connected gigabit network between them. You can recognize that metric (2816) from R1's own directly connected gigabit links on lines 6 and 14.

10.13.1.3 [line 11] wasn't offered because it was hideously expensive (40 512 256). It also is not a feasible successor because its reported distance of 2816 is not less than the feasible distance of the successor route (also 2816). The metric was calculated based on the "bandwidth" command on the local interface, *not* the actual clock rate of the serial line and not the bandwidth setting at the other end.

To tie up some loose ends,

- The P in the left column tells you that the routes are passive. Active (A) routes are "actively" being recalculated. You want them to be passive
- For each "via" line, the address belongs to the next hop router and the interface is local

Loops and Feasible Successor Convergence

A feasible successor is essentially preapproved, meaning that during a failure, the router can immediately switch to it knowing that it is mathematically impossible for that route to create a routing loop. In the example above, we know that switching to the slower serial line wouldn't cause a loop, but the algorithm can't rule it out completely, so before it is used, EIGRP will need to query its neighbors to be sure, delaying recovery.

Mr. Odom glosses over what a routing loop is and doesn't even say as much as the paragraph above,

If the topology above is fully converged, R2 knows how to get to R3 with a cost of 1, which it advertises, and R1 has a cost of 2, thanks to knowledge it gained from R2. Now if the link between R2 and R3 goes down, R2 will be looking for a new path to R3. R1's cost of 2 will be the best available and suddenly you have a routing loop. Packets for R3 are sent from R2 to R1 and from R1 to R2.

Avoiding loops accounts for many of the complexities in routing protocols, from route poisoning to hold-down timers, feasible successors, and OSPF's obsessive topology table that stores every possible link in the entire area on every router.

Show IP EIGRP Topology <network>

By default, the EIGRP *composite* metric is based solely on the minimum bandwidth along a path (slowest link) and the cumulative delay (sum of all incoming interface delays during advertisement = all interfaces actual data would pass *out* through on its way to a destination). The topology table stores the individual metric components, including some that are ignored by default, like reliability and traffic load. Cisco recommends that you don't bother with those.

Each of the descriptor blocks describes one next hop possibility in the direction of the network you named. As each advertisement is passed from router to router, the recipient adds the delay of its own interface to the delay metric and replaces the bandwidth if its interface is slower than what's shown. So, in this command, you're literally seeing the minimum bandwidth and total delay.

The other possible metric components are listed also, but their weightings (or "K" values) are 0 by default. Curiously, this is the first time we've seen routers referred to by their router-id [Lines 14, 24, 34].

```
 1  R1# show ip eigrp topology 10.23.0.0/24
 2  EIGRP-IPv4 Topology Entry for AS(42)/ID(0.0.0.1) for 10.23.0.0/24
 3    State is Passive, Query origin flag is 1, 2 Successor(s), FD is 3072
 4    Descriptor Blocks:
 5    10.12.0.2 (GigabitEthernet0/0), from 10.12.0.2, Send flag is 0x0
 6        Composite metric is (3072/2816), route is Internal
 7        Vector metric:
 8          Minimum bandwidth is 1000000 Kbit
 9          Total delay is 20 microseconds
10          Reliability is 255/255
11          Load is 1/255
12          Minimum MTU is 1500
13          Hop count is 1
14          Originating router is 0.0.0.2
15    10.13.0.3 (GigabitEthernet0/1), from 10.13.0.3, Send flag is 0x0
16        Composite metric is (3072/2816), route is Internal
17        Vector metric:
18          Minimum bandwidth is 1000000 Kbit
19          Total delay is 20 microseconds
20          Reliability is 255/255
21          Load is 1/255
22          Minimum MTU is 1500
23          Hop count is 1
24          Originating router is 0.0.0.3
25    10.13.1.3 (Serial0/2/0), from 10.13.1.3, Send flag is 0x0
26        Composite metric is (832256/2816), route is Internal
27        Vector metric:
28          Minimum bandwidth is 8000 Kbit
29          Total delay is 20010 microseconds
30          Reliability is 255/255
31          Load is 1/255
32          Minimum MTU is 1500
33          Hop count is 1
34          Originating router is 0.0.0.3
```

LOAD BALANCING

As mentioned, EIGRP will offer the routing table up to 4 equal-cost routes to the same destination by default. Just like with OSPF, this number can be changed using "maximum-paths." Unlike OSPF, EIGRP has the ability to load-balance across unequal *feasible successors* using the "variance" command. By default, variance is set to 1 to only use equal-cost routes, but can be set up to 128.

```
 1  R1(config)# router eigrp 42
 2  R1(config-router)# maximum-paths 6
 3  R1(config-router)# variance 2
 4               Paths with a metric up to 2x the feasible distance will be included in routing table
```

The RD of the varying route will still need to be less than the FD of the successor, to prevent loops.

The bandwidth and delay values used by EIGRP are based on defaults for the type of interfaces used.

- On Ethernet interfaces, these values will adjust to match the speed of the interface

- If you intentionally slow down an Ethernet interface using the speed command, the bandwidth and delay will automatically adjust

- You can override these defaults using the "bandwidth" and "delay" interface configurations

- Serial lines always default to the bandwidth of T1 speed (1.544 Mbps) no matter the clock rate; you'll need to manually set the bandwidth if you're using something else. The default delay of 0.02 seconds (20 000 μsec) can probably be left alone
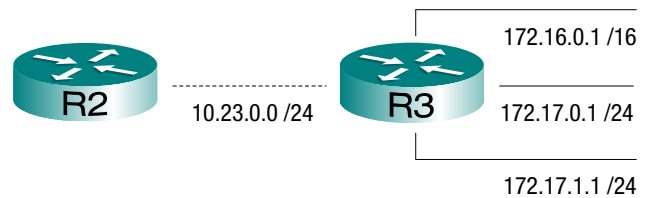
```
R3(config)# interface s0/2/0
R3(config-if)# clock rate 64000
R3(config-if)# bandwidth 64
```
     Value is in Kbits / second
```
R3(config-if)# delay 2000
```
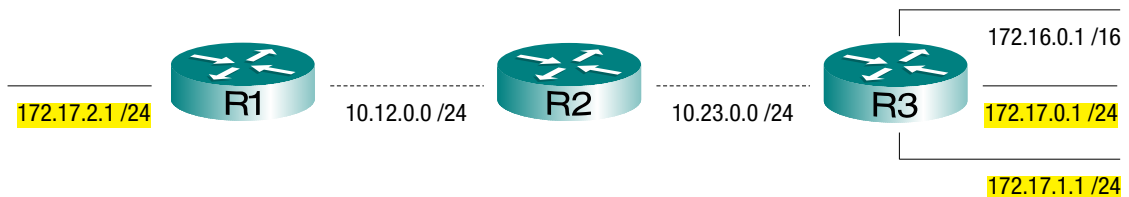   *This matches the default, in tens of μsecs. (= 0.02 seconds), so has no effect*

- Cisco recommends that you use accurate bandwidth settings. If you really must mess with EIGRP's decisions, change the delay; it has fewer side effects

AUTOSUMMARY

Autosummarization simplifies routing tables and routing protocol updates by announcing a route to an entire classful network whenever the update is crossing a different classful network. For example, here R3 will advertise to R2 that it has the entire classful nets 172.16.0.0 / 16 and 172.17.0.0 /16 because the update messages are crossing a different classful net, 10.0.0.0 /8. This can be fine and leave room for growth on R3, as long as it's intentional and you don't later do this:

```
                                                    172.16.0.1 /16
  [R2]  ................  [R3]
       10.23.0.0 /24           172.17.0.1 /24

                                                    172.17.1.1 /24
```

Discontiguous Network

```
172.17.2.1 /24 [R1]  ...........  [R2]  ...........  [R3]   172.16.0.1 /16
                  10.12.0.0 /24       10.23.0.0 /24        172.17.0.1 /24

                                                            172.17.1.1 /24
```

Now you have both R1 and R3 telling R2 that they have the entire classful network 172.17.0.0 /16. This is a situation where you would want to summarize manually, if at all. For example, R3 could safely advertise 172.16.0.0 /16 and 172.17.0.0 /23.

EIGRP can do all of these things, even the bad ones. If you do get it wrong, you could see a 50% ping loss from R2 as it automatically load-balances two equal-cost routes to 172.17.0.0 /16, sending half of the pings east, and half west. By default, EIGRP doesn't auto-summarize.

```
R3(config)# router eigrp 42
R3(config-router)# no auto-summary
```

Discontiguous Network—Subnets of the same classful net are separated by a different classful net.