Day 25: RIP & EIGRP

Introduction

- This Lecture covers two dynamic routing protocols: **RIP (Routing Information Protocol)** and **EIGRP (Enhanced Interior Gateway Routing Protocol)**
- Neither is explicitly listed on the CCNA exam topics list

Why Study RIP & EIGRP?

- You might still get a few questions about RIP or EIGRP on the exam, so it's best to be prepared
- Cisco's exam topics list states: *"The following topics are general guidelines for the content likely to be included on the exam. However, other related topics may also appear on any specific delivery of the exam."*
- Don't assume that if something isn't on the exam topics list, you don't need to know anything about it
- Won't study RIP and EIGRP to the same depth as OSPF (which is explicitly listed on the exam topics)
- This Lecture provides an overview of both with basic configurations — just enough to understand how they work, not to memorize for the exam
- Many concepts learned here also apply to OSPF, making it easier when OSPF is introduced in Day 26

RIP (Routing Information Protocol)

RIP Basics

- **Industry standard** protocol (not Cisco proprietary)

- **Distance vector** interior gateway protocol — uses "routing-by-rumour" logic to learn and share routes
    - *(Refer to Day 24 for a review of distance vector logic)*

Metric

- Uses **hop count** as its metric
    - Each router in the path to the destination = one hop
    - Bandwidth is irrelevant (a 10 Gbps link = 1 hop, a 10 Mbps link = 1 hop)
- **Maximum hop count is 15**
    - Anything more than 15 is considered unreachable
    - RIP will not insert the route into the routing table
- Because of the 15 hop limit, RIP **cannot be used for very large networks**

Usage

- Almost never used in real networks
- Can possibly be used in small networks or lab environments as a simple IGP that is quick and easy to set up

Versions

- **RIPv1 and RIPv2** — for IPv4
- **RIPng (Next Generation)** — for IPv6 *(not covered in this course)*

Message Types

1. **Request message** — asks RIP-enabled neighbor routers to send their routing table
2. **Response message** — sends the local router's routing table to neighboring routers

- By default, RIP-enabled routers share their routing table **every 30 seconds**
    - Can cause problems in networks with lots of routers — these regular updates can clog up the network

---

RIPv1 vs RIPv2

RIPv1

- Very old protocol — **don't use version 1**
- Only advertises **classful addresses** (Class A, B, and C)
- Does **not** support VLSM and CIDR
- Does **not** include subnet mask information in advertisements
  - Class A range → assumed /8
  - Class B range → assumed /16
  - Class C range → assumed /24
- Forces subnets to become classful networks:

Original Subnet	Becomes (Classful)
10.1.1.0/24	10.0.0.0 (Class A)
172.16.192.0/18	172.16.0.0 (Class B)
192.168.1.4/30	192.168.1.0 (Class C)

- Not acceptable in modern networks — we need the ability to use subnets, not just classful networks
- Messages are **broadcast** to **255.255.255.255** — all routers on the local segment will receive them

RIPv2

- **Supports VLSM and CIDR** — not restricted to classful
- **Includes subnet mask information** in route advertisements (e.g., a /30 network is advertised as /30)

- Messages are **multicast** to **224.0.0.9** (in the Class D range, reserved for multicast)

Broadcast vs Multicast

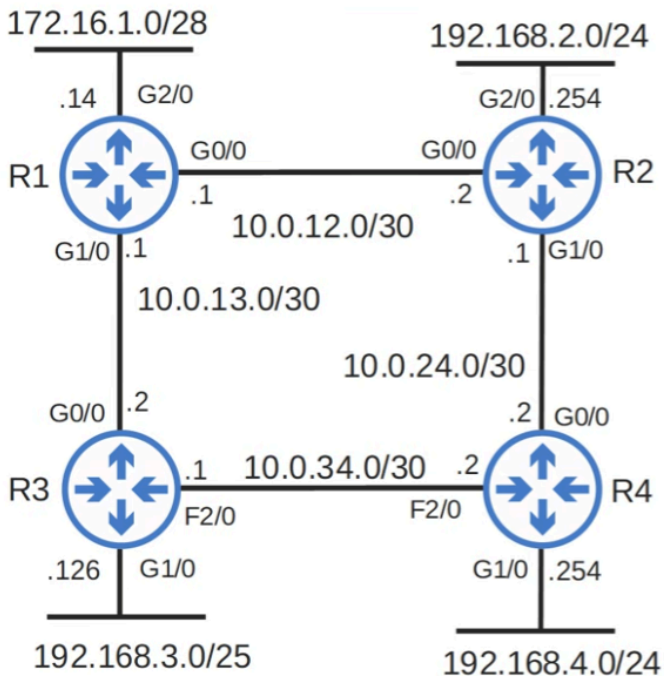
- **Broadcast:** delivered to all devices on the local network
- **Multicast:** only received by devices that have joined that specific multicast group
- Only need to know this basic difference at the CCNA level; deeper multicast knowledge is at the CCIE level

---

RIP Basic Configuration

> ***Note: RIP configuration is not actually on the CCNA, but is introduced for two reasons: 1) RIP configuration is very simple and acts as a good introduction to dynamic routing configuration, and 2) some of the mechanics are similar to how OSPF configuration works, making it easier when we go in depth on OSPF later.***

Network Topology




```
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#no auto-summary
R1(config-router)#network 10.0.0.0
R1(config-router)#network 172.16.0.0
```

Configuration Commands

```
R1(config)# router rip
R1(config-router)# version 2
R1(config-router)# no auto-summary
R1(config-router)# network 10.0.0.0
R1(config-router)# network 172.16.0.0
```

`router rip`

- Enters RIP configuration mode
- Prompt changes from `config` to `config-router`

`version 2`

- Configures the router to use RIP version 2
- Not technically necessary, but you should always use version 2
- Classful IPv4 addressing is a thing of the past; modern networks need VLSM and CIDR

`no auto-summary`

- Auto-summary is **on by default**
- Automatically converts advertised networks to classful networks
- Example: using classful logic, 172.16.1.0/28 (a Class B address) would be advertised as 172.16.0.0/16
- **Always use `version 2` followed by `no auto-summary`**

`network` Command — Key Behavior

- The command itself is **classful** — automatically converts to classful networks
  - Example: `network 10.0.12.0` → converted to `10.0.0.0` (Class A → /8 assumed, all bits after the first 8 become 0)

- No need to enter a network mask

Walkthrough — `network 10.0.0.0` on R1:

- R1's G0/0 interface = 10.0.12.0/30
- R1's G1/0 interface = 10.0.13.0/30
- The `network 10.0.0.0` command does **not** simply advertise the 10.0.0.0 network — its actual function is explained in the next section

---

How the `network` Command Functions

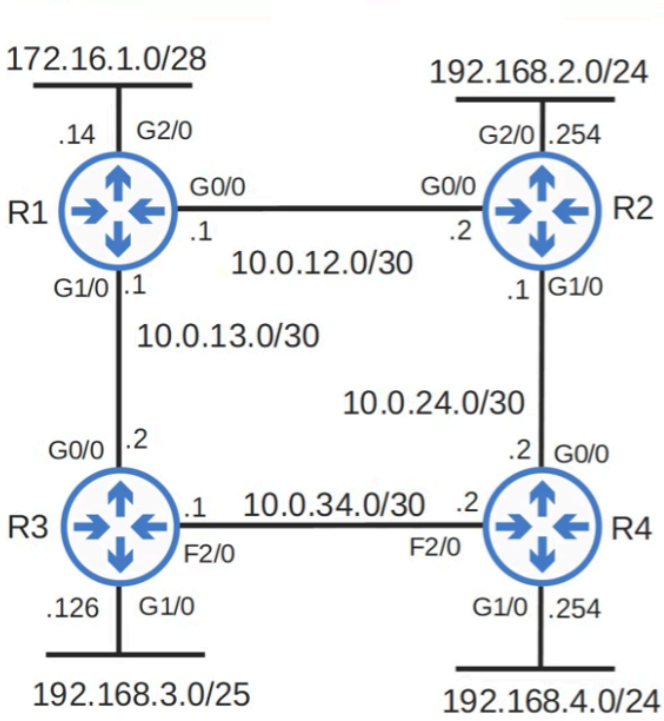
Core Behavior

- The `network` command tells the router to look for interfaces with an IP address that is in the specified range
- Then it will **activate RIP** on the interface or interfaces that fall in the range
- It will form adjacencies with other connected RIP-enabled neighbors
- It will advertise the **network prefix of the interface** — this is not necessarily the prefix you specified in the network command

> ***Note: This is also how the EIGRP and OSPF `network` commands operate, although there are a few differences.***

`network` Command Walkthrough

`network 10.0.0.0` on R1



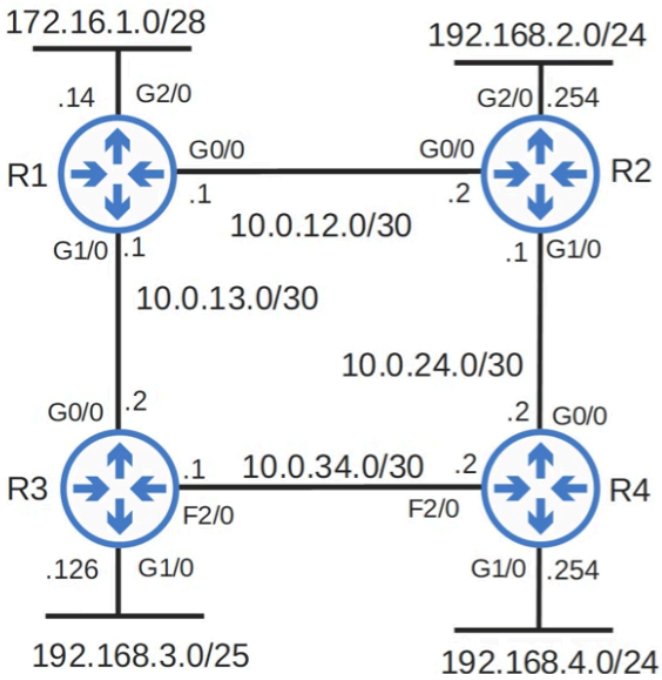

```
R1(config-router)#network 10.0.0.0
```

- Because the **network** command is classful, 10.0.0.0 is assumed to be 10.0.0.0/8
- R1 will look for any interfaces with an IP address that matches 10.0.0.0/8 (because it is /8 it only needs to match the first 8 bits)
- 10.0.12.1 and 10.0.13.1 both match, so RIP is activated on G0/0 and G1/0.
- R1 forms adjacencies with its neighbors R2 and R3.
- R1 advertises 10.0.12.0/30 and 10.0.13.0/30 (NOT 10.0.0.0/8) to its RIP neighbors.

1. Because the `network` command is classful, 10.0.0.0 is assumed to be 10.0.0.0/8
2. R1 looks for any interfaces with an IP address matching 10.0.0.0/8
    - /8 means only the first 8 bits (first octet) need to match
    - 10.0.12.1 and 10.0.13.1 both match (same first octet of 10)
3. RIP is activated on G0/0 and G1/0
4. R1 forms adjacencies with neighbors R2 and R3
5. R1 advertises **10.0.12.0/30** and **10.0.13.0/30** (the actual network prefixes of its interfaces) to R2 and R3
    - R1 does **not** advertise 10.0.0.0/8

> ***Important: The `network` command doesn't tell the router which networks to advertise. It tells the router which interfaces to activate RIP on, and then the router advertises the network prefix of those interfaces.***

`network 172.16.0.0` on R1




```
R1(config-router)#network 172.16.0.0
```

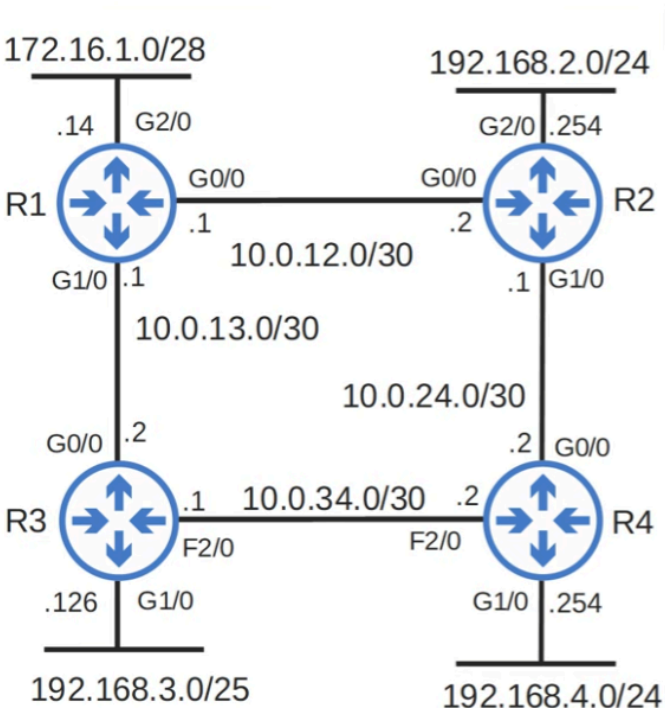
- Because the **network** command is classful, 172.16.0.0 is assumed to be 172.16.0.0/16
- R1 will look for any interfaces with an IP address that matches 172.16.0.0/16
- 172.16.1.14 matches, so R1 will activate RIP on G2/0.
- There are no RIP neighbors connected to G2/0, so no new adjacencies are formed.
- R1 advertises 172.16.1.0/28 (NOT 172.16.0.0/16) to its RIP neighbors.

1. Because the command is classful, 172.16.0.0 is assumed to be 172.16.0.0/16
2. R1 looks for interfaces matching 172.16.0.0/16
    - 172.16.1.14 matches → RIP activated on G2/0
3. No RIP neighbors connected to G2/0, so no new adjacencies formed
4. R1 advertises **172.16.1.0/28** (NOT 172.16.0.0/16) to its RIP neighbors

> ***Important: Although there are no RIP neighbors connected to G2/0, R1 will continuously send RIP advertisements out of G2/0 — this is unnecessary traffic, so G2/0 should be configured as a passive interface.***

Passive Interface

Configuration Command



R1(config-router)#passive-interface g2/0

- The **passive-interface** command tells the router to stop sending RIP advertisements out of the specified interface (G2/0)
- However, the router will continue to advertise the network prefix of the interface (172.16.1.0/28) to its RIP neighbors (R2, R3).
- You should always use this command on interfaces which don't have any RIP neighbors.
- EIGRP and OSPF both have the same passive interface functionality, using the same command.

```
R1(config-router)# passive-interface G2/0
```

- Configures G2/0 as a passive interface

> ***Note: The command is done from RIP configuration mode, not directly on the interface itself. That's why you need to specify the interface in the command.***

What It Does

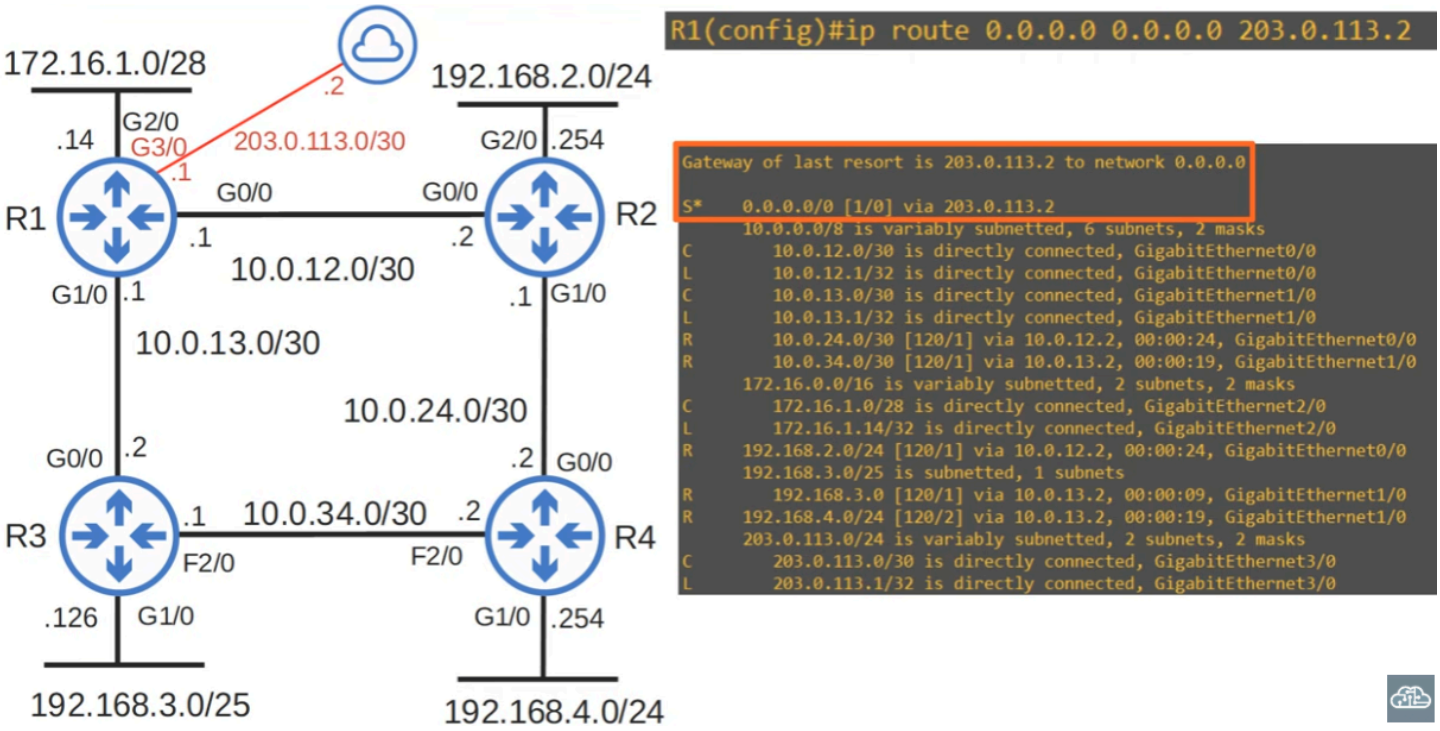
- Tells the router to **stop sending RIP advertisements** out of the specified interface (G2/0)
- The router **will continue to advertise** the network prefix of the interface (172.16.1.0/28) to its RIP neighbors (R2 and R3)

> ***Recommendation: Always use this command on interfaces which don't have any RIP neighbors.***

> ***Note: EIGRP and OSPF both have the same passive interface functionality, using the same command.***

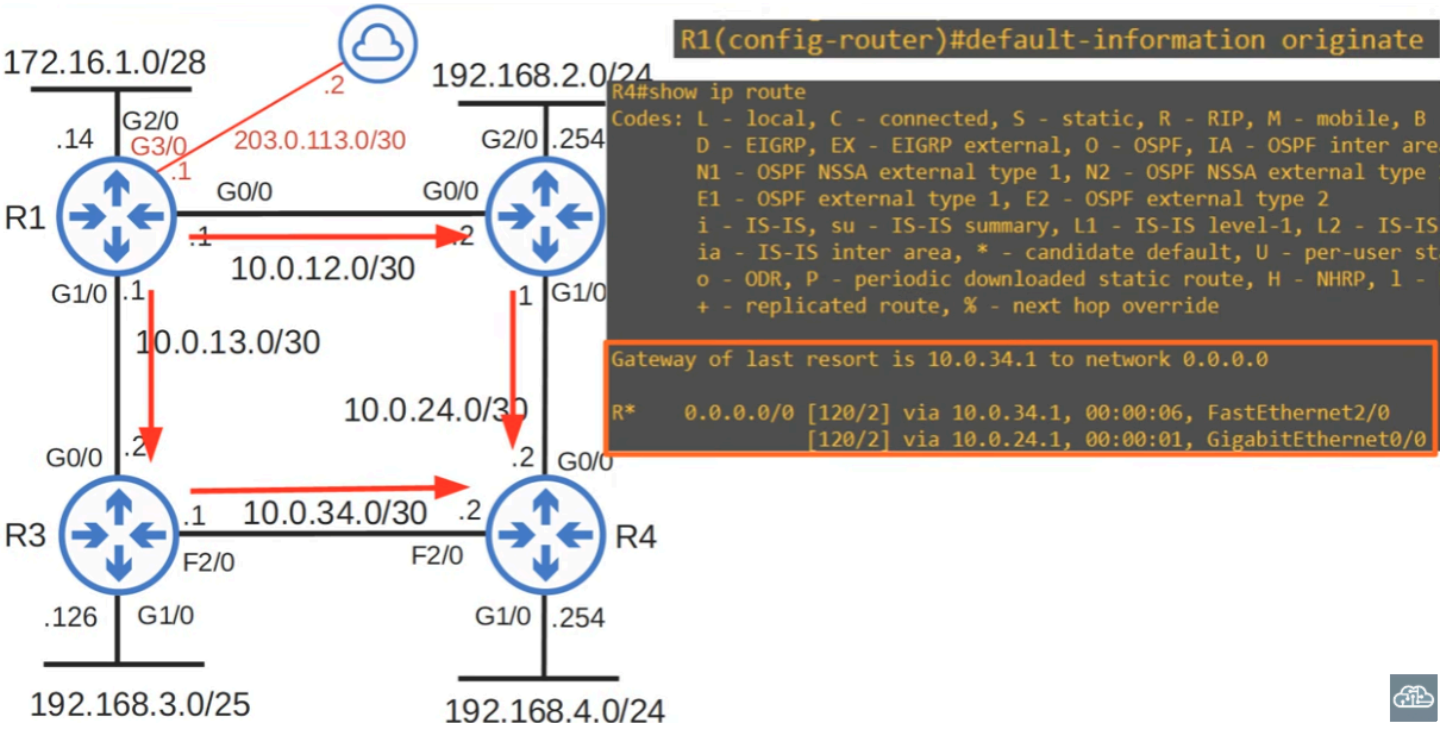
Default Route Advertisement in RIP

Setup



- Added an Internet connection to R1 via G3/0
- Configured a static default route pointing to the Internet
- Any packets that don't match other entries in R1's routing table will be sent to the Internet
- Routing table shows: `Gateway of last resort is 203.0.113.2 to network 0.0.0.0`

Configuration Command




```
R1(config-router)# default-information originate
```

- Done from RIP configuration mode
- Tells R1 to advertise its default route to RIP neighbors
- R1 advertises the route to R2 and R3, and they then advertise it to R4

Result on R4

```
R4#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter are
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
       ia - IS-IS inter area, * - candidate default, U - per-user st
       o - ODR, P - periodic downloaded static route, H - NHRP, l -
       + - replicated route, % - next hop override

Gateway of last resort is 10.0.34.1 to network 0.0.0.0

R*    0.0.0.0/0 [120/2] via 10.0.34.1, 00:00:06, FastEthernet2/0
                [120/2] via 10.0.24.1, 00:00:01, GigabitEthernet0/0
```

- R4's routing table shows: `Gateway of last resort is 10.0.34.1 to network 0.0.0.0`
- Two routes visible: one via F2/0 to R3, and one via G0/0 to R2
- Both routes have the **same hop count** → R4 **load-balances** traffic over the two routes

> ***Note: RIP treats all connections equally as one hop, so even though the connection via R3 is a slower FastEthernet connection, RIP considers it equal to the faster Gigabit Ethernet connection via R2.***

> ***Note: OSPF also has the same `default-information originate` command to share a default route with neighbors.***

`show ip protocols` Command (RIP)

Verification Command

```
R1# show ip protocols
```

- Can be used for RIP, EIGRP, and OSPF to check various stats

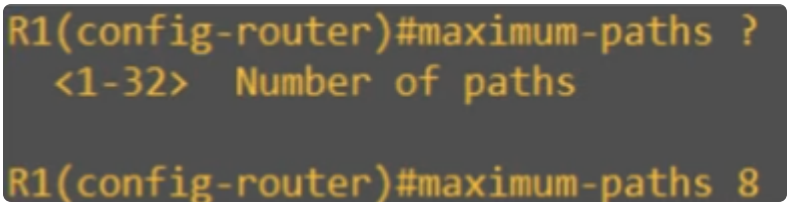
```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 28 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface              Send  Recv  Triggered RIP  Key-chain
    GigabitEthernet0/0     2     2
    GigabitEthernet1/0     2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    172.16.0.0
  Passive Interface(s):
    GigabitEthernet2/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.0.12.2            120      00:00:21
    10.0.13.2            120      00:00:06
  Distance: (default is 120)
```

Key Output Fields

Field	Description
Protocol	RIP
Version	Version 2 (as configured)
Automatic network summarization	Not in effect (due to `no auto-summary`)
Maximum paths	4 (default ECMP load-balancing, can be changed)
Networks	Shows networks entered via `network` command (not actual advertised networks)
Passive interfaces	G2/0
Routing information sources	R1's RIP neighbors (R2: 10.0.12.2, R3: 10.0.13.2)
Distance	Administrative distance = 120 (default)

> ***Note: Timers are shown but not covered for RIP or EIGRP; will be covered in depth with OSPF.***

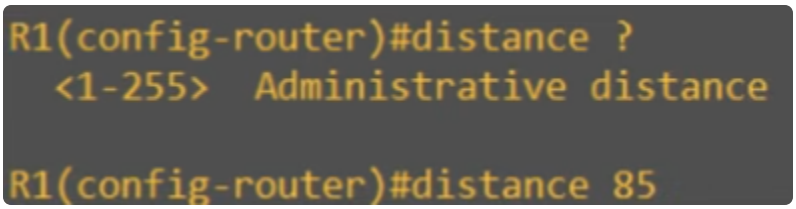
`maximum-paths` Command



```
R1(config-router)# maximum-paths 8
```

- Default: 4 paths (ECMP load-balancing)
- Range: 1 to 32
- Done from RIP configuration mode
- Same command for EIGRP and OSPF

`distance` Command




```
R1(config-router)# distance 85
```

- Default AD for RIP: 120
- Range: 1 to 255
- Done from RIP configuration mode
- Example: setting to 85 makes RIP routes preferred over EIGRP routes (EIGRP AD = 90)
- Same command for EIGRP and OSPF

EIGRP (Enhanced Interior Gateway Routing Protocol)

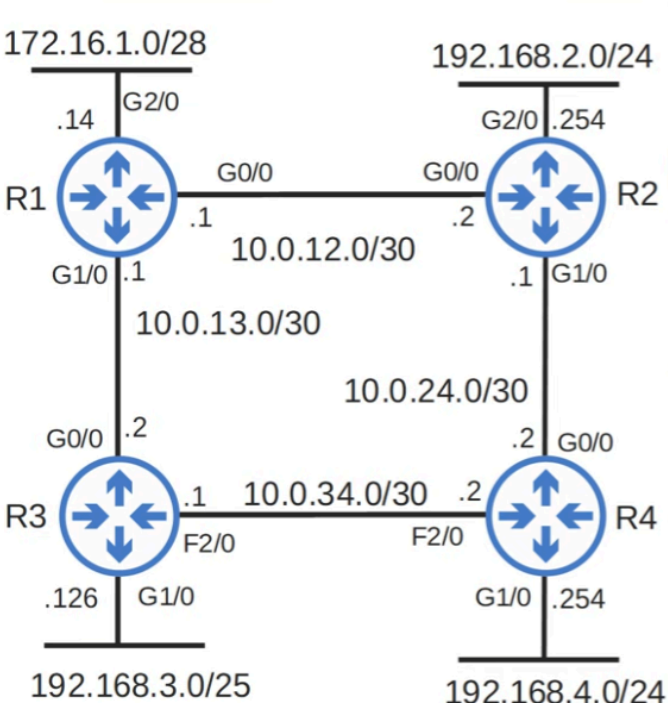
EIGRP Basics

- **Enhanced Interior Gateway Routing Protocol**
- Improved version of the older IGRP (Interior Gateway Routing Protocol)
- Was **Cisco proprietary**, Cisco has published it openly but parts remain proprietary
    - Practically speaking, still considered a **Cisco-only protocol**
- Considered an **"advanced" or "hybrid" distance vector** routing protocol
- Improves on basic operations of RIP
- **Much faster** than RIP in reacting to network changes
- **No 15 hop-count limit** — supports very large networks
- Messages sent via **multicast address 224.0.0.10**
    - RIPv1: broadcast (255.255.255.255)
    - RIPv2: multicast (224.0.0.9)
    - EIGRP: multicast (224.0.0.10)
    - Try to remember these addresses — they could come up as exam questions
- **Only IGP that can perform unequal-cost load-balancing**
    - By default: ECMP load-balancing over 4 paths (like RIP)
    - Can configure to load-balance over multiple unequal-cost paths
    - Load-balances **in proportion to bandwidth** (more traffic on lower-metric/faster paths, less traffic on higher-metric/slower paths)
- Not used as much as OSPF because of Cisco limitation → OSPF is the main focus of CCNA for dynamic routing protocols

---

EIGRP Basic Configuration

Configuration Commands




```
R1(config)#router eigrp 1
R1(config-router)#no auto-summary
R1(config-router)#passive-interface g2/0
R1(config-router)#network 10.0.0.0
R1(config-router)#network 172.16.1.0 0.0.0.15
```

- The AS (Autonomous System) number must match between routers, or they will not form an adjacency and share route information.
- Auto-summary might be enabled or disabled by default, depending on the router/IOS version. If it's enabled, disable it.
- The **network** command will assume a classful address if you don't specify the mask.
- EIGRP uses a *wildcard mask* instead of a regular subnet mask.

```
R1(config)# router eigrp 1
R1(config-router)# no auto-summary
R1(config-router)# passive-interface G2/0
R1(config-router)# network 10.0.0.0
R1(config-router)# network 172.16.1.0 0.0.0.15
```

> ***Note: You can have both RIP and EIGRP running at the same time, but that would just be a waste of resources on the routers, so usually there will just be one IGP running on a router.***

`router eigrp 1`

- Enters EIGRP configuration mode
- **1** = Autonomous System (AS) number

- **AS number must match** between routers or they will not form an adjacency and share route information

`no auto-summary`

- Same function as in RIP — advertises classful networks instead of the actual network prefix configured on its interfaces
- May be enabled or disabled by default depending on router/IOS version
- If enabled, disable it

`network 10.0.0.0` (without mask)

- Same as RIP — assumes classful address if you don't specify the mask
- `network 10.0.0.0` → assumed 10.0.0.0/8
- Activates EIGRP on any interface whose IP address falls in the 10.0.0.0/8 range (any IP starting with 10)
- That includes G0/0 and G1/0, so EIGRP is activated on both interfaces

`network 172.16.1.0 0.0.0.15` (with mask)

- Activates EIGRP on the G2/0 interface
- **Uses a wildcard mask (0.0.0.15) instead of a regular subnet mask (255.255.255.240)**

Wildcard Masks

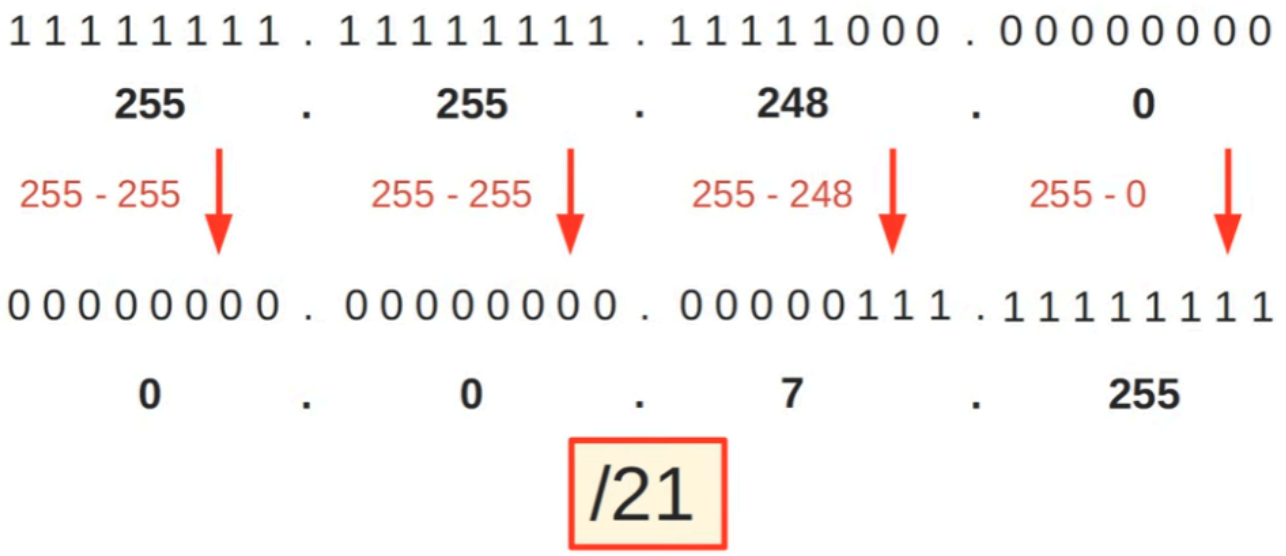
Definition

- A wildcard mask is an **"inverted" subnet mask**
- All 1s in the subnet mask → 0 in the wildcard mask
- All 0s in the subnet mask → 1 in the wildcard mask

Conversion Examples

Prefix	Subnet Mask (Decimal)	Subnet Mask (Binary)	Wildcard Mask (Binary)	Wildcard Mask (Decimal)
/8	255.0.0.0	11111111.00000000.00000000.00000000	00000000.11111111.11111111.11111111	0.255.255.255
/14	255.252.0.0	11111111.11111100.00000000.00000000	00000000.00000011.11111111.11111111	0.3.255.255
/16	255.255.0.0	11111111.11111111.00000000.00000000	00000000.00000000.11111111.11111111	0.0.255.255
/19	255.255.224.0	11111111.11111111.11100000.00000000	00000000.00000000.00011111.11111111	0.0.31.255
/21	255.255.248.0	11111111.11111111.11111000.00000000	00000000.00000000.00000111.11111111	0.0.7.255
/24	255.255.255.0	11111111.11111111.11111111.00000000	00000000.00000000.00000000.11111111	0.0.0.255
/25	255.255.255.128	11111111.11111111.11111111.10000000	00000000.00000000.00000000.01111111	0.0.0.127
/28	255.255.255.240	11111111.11111111.11111111.11110000	00000000.00000000.00000000.00001111	0

Shortcut Method



- Subtract each octet of the subnet mask from 255
- Example: 255.255.248.0 → (255-255).(255-255).(255-248).(255-0) = 0.0.7.255

How Wildcard Masks Function

So to find out how many bits must match, simply **count the number of 0s** in the wildcard mask (in binary).

- **0** in wildcard mask = bits **must match** between interface IP and network command
- **1** in wildcard mask = bits **don't have to match**

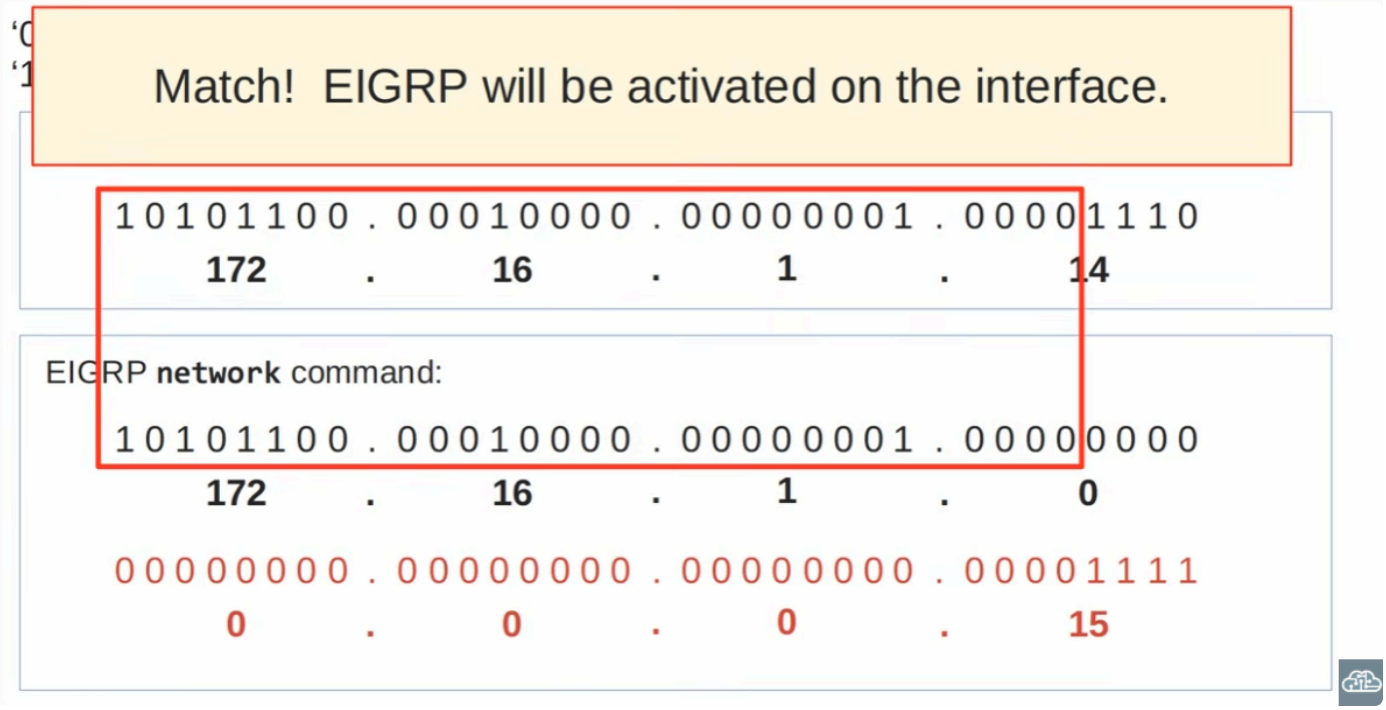
Practice Examples

(Using R1's G2/0 interface IP: 172.16.1.14)

Example 1: `network 172.16.1.0 0.0.0.15`

- **Wildcard Mask:** 0.0.0.15 (`00000000.00000000.00000000.00001111`)
- **Interface IP:** 172.16.1.14 = `10101100.00010000.00000001.00001110`

- **Network Command:** 172.16.1.0 = `10101100.00010000.00000001.00000000`



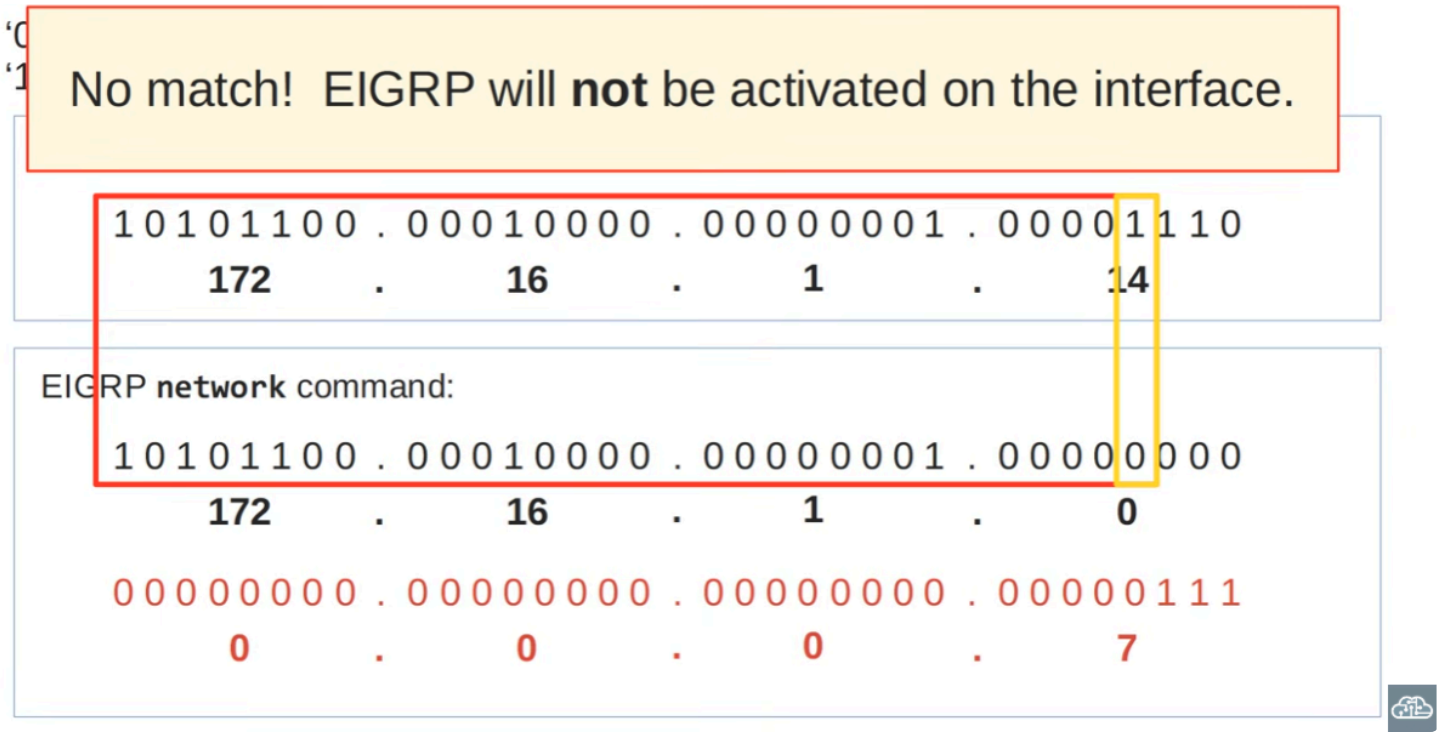
Octet	Interface IP	Network Command	Match?
1	10101100	10101100	Yes
2	00010000	00010000	Yes
3	00000001	00000001	Yes
4	0000**1110**	0000**0000**	Yes (first 4 bits match)

- First 28 bits must match → **Match** → EIGRP activated

Example 2: `network 172.16.1.0 0.0.0.7`

- **Wildcard Mask:** 0.0.0.7 (`00000000.00000000.00000000.00000111`)
- **Interface IP:** 172.16.1.14 = `10101100.00010000.00000001.00001110`

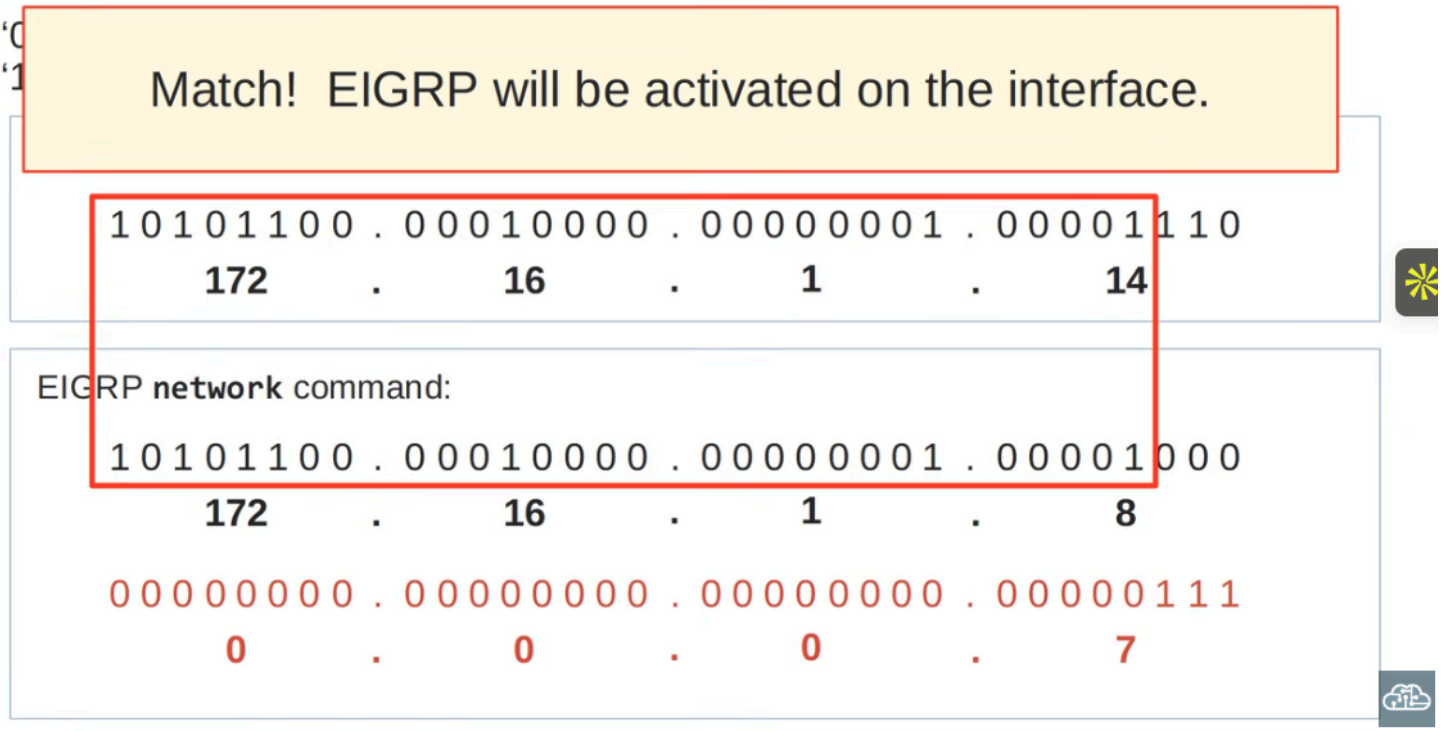
- **Network Command:** 172.16.1.0 = `10101100.00010000.00000001.00000000`



Octet	Interface IP	Network Command	Match?
1	10101100	10101100	Yes
2	00010000	00010000	Yes
3	00000001	00000001	Yes
4	00001110	00000000	**No** (5th bit differs)

- First 29 bits must match → **No match** (bit 29 differs) → EIGRP NOT activated

Example 3: `network 172.16.1.8 0.0.0.7`

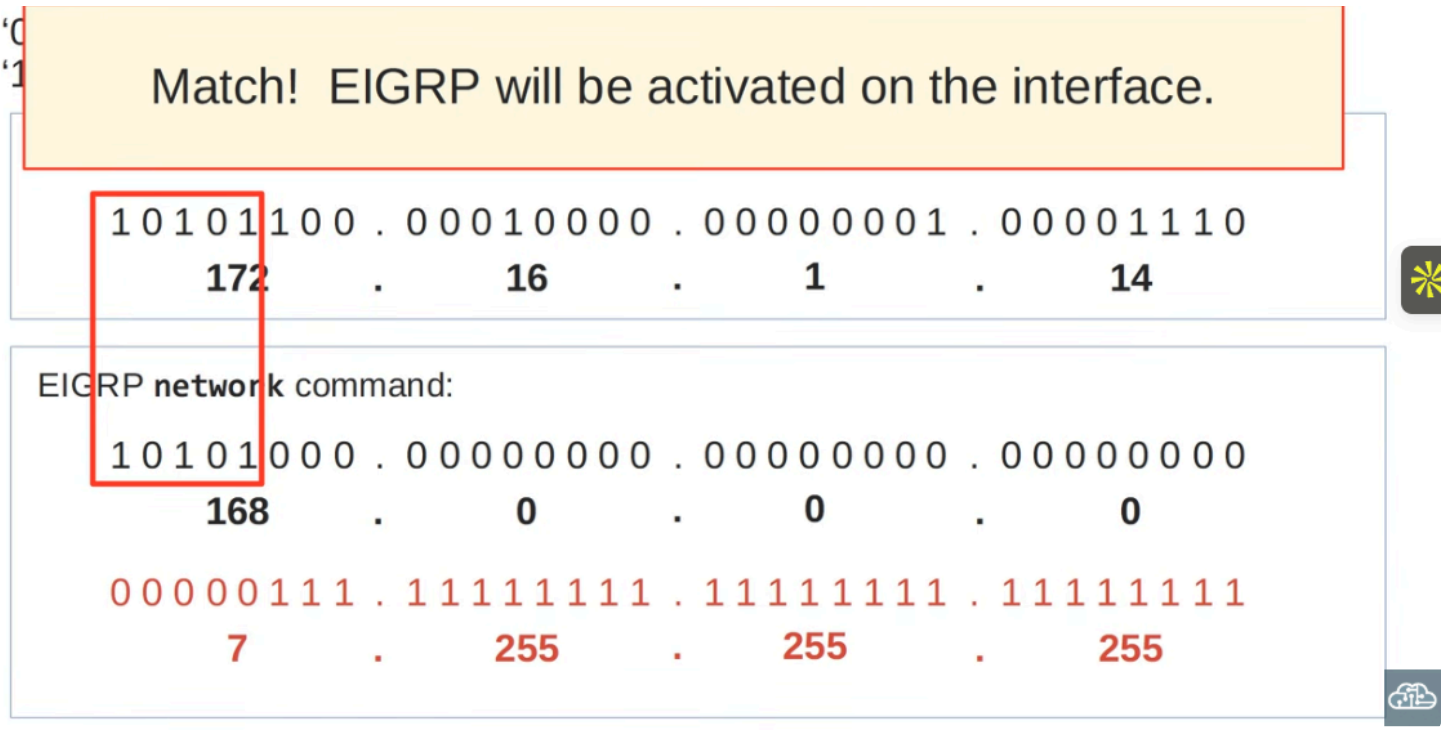


- **Wildcard Mask:** 0.0.0.7 (`00000000.00000000.00000000.00000111`)
- **Interface IP:** 172.16.1.14 = `10101100.00010000.00000001.00001110`
- **Network Command:** 172.16.1.8 = `10101100.00010000.00000001.00001000`

Octet	Interface IP	Network Command	Match?
1	10101100	10101100	Yes
2	00010000	00010000	Yes
3	00000001	00000001	Yes
4	00001110	00001000	Yes (first 5 bits match)

- First 29 bits must match → **Match** → EIGRP activated

Example 4: `network 168.0.0.0 7.255.255.255`

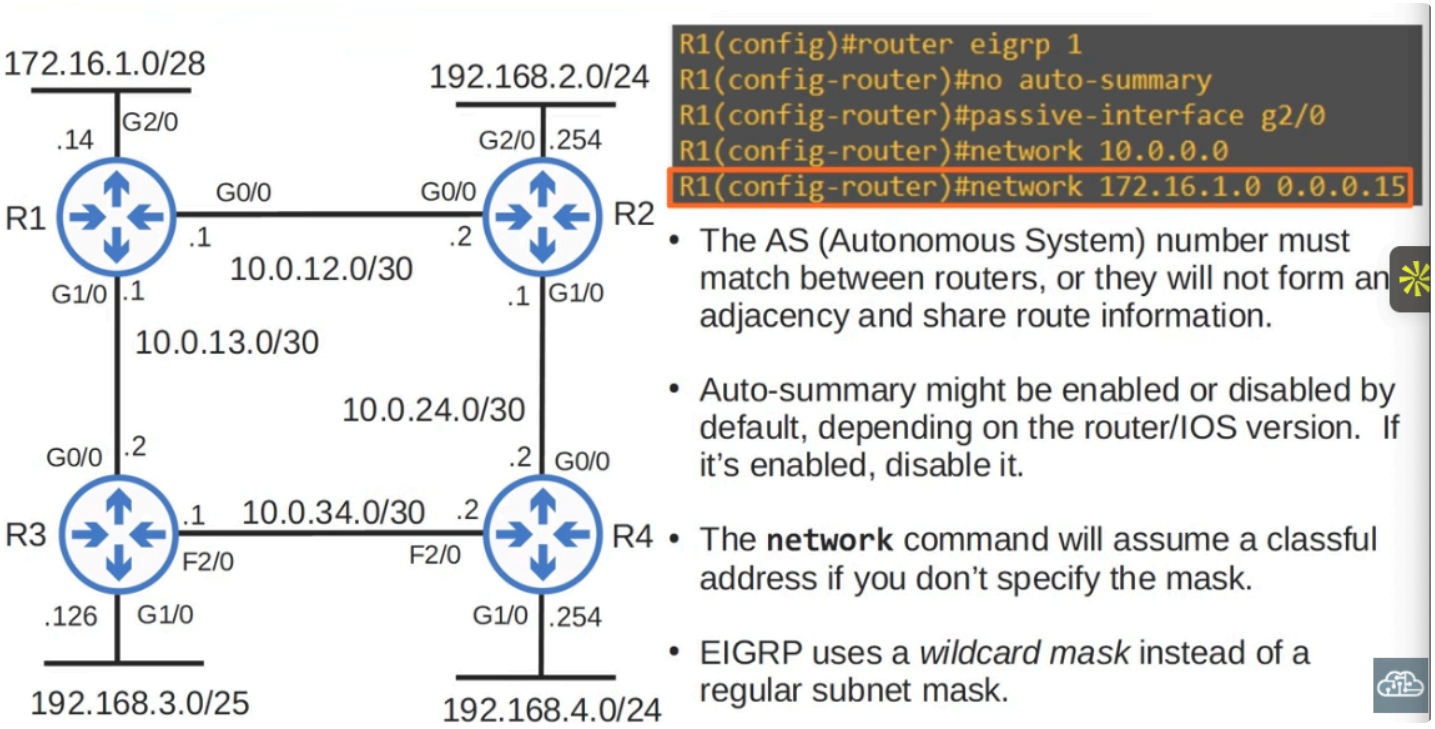


- **Wildcard Mask:** 7.255.255.255 (`00000111.11111111.11111111.11111111`)
- **Interface IP:** 172.16.1.14 = `10101100.00010000.00000001.00001110`
- **Network Command:** 168.0.0.0 = `10101000.00000000.00000000.00000000`

Octet	Interface IP	Network Command	Match?
1	10101**100**	10101**000**	Yes (first 5 bits match)
2	00010000	00000000	Yes (Wildcard allows any value)
3	00000001	00000000	Yes (Wildcard allows any value)
4	00001110	00000000	Yes (Wildcard allows any value)

- First 5 bits must match → **Match** → EIGRP activated

Additional Notes



- Usually you'll just use the same prefix length as the interface itself (e.g., /28 wildcard mask for a /28 network)
- Or use a /32 wildcard mask
- How would you write a /32 wildcard mask?
  - Subnet mask is 255.255.255.255
  - So the wildcard mask would be all (0.0.0.0) to specify the exact IP address
- OSPF also uses wildcard masks (reviewed in a future video)

> ***Important: The `network` command only specifies which interfaces to activate EIGRP on. R1 will then advertise the network prefix of that interface (172.16.1.0/28 in this case).***

`show ip protocols` Command (EIGRP)

Verification Command

```
R1# show ip protocols
```

Key Output Fields

Field	Description
Routing protocol	EIGRP 1 (1 = AS number)
K-values	K1=1, K3=1 (used), K2=0, K4=0, K5=0 (not used)
Metric calculation	Based on **bandwidth of slowest link** + **sum of delay values** of all links
Router ID	172.16.1.14 (default: highest IP on physical interface)
Automatic summarization	Disabled
Maximum paths	4 (ECMP load-balancing default)
Networks	10.0.0.0 and 172.16.1.0/28
Passive interface	G2/0
Neighbors	R2 and R3
AD values	**90** for internal routes, **170** for external routes

> ***Note: K2, K4, and K5 are set to 0 by default and are not used, but that can be changed with configuration.***

> ***Note: Internal routes are normal EIGRP routes. External routes are routes from outside EIGRP which are then inserted into EIGRP — that's a more advanced topic for the CCNP level.***

`show ip protocols` Command (EIGRP)

Verification Command

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(1)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 172.16.1.14
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1

  Automatic Summarization: disabled
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    172.16.1.0/28
  Passive Interface(s):
    GigabitEthernet2/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.0.12.2              90      00:00:23
    10.0.13.2              90      00:00:23
  Distance: internal 90 external 170
```

```
R1# show ip protocols
```

Key Output Fields

Field	Description
Routing protocol	EIGRP 1 (1 = AS number)
K-values	K1=1, K3=1 (used), K2=0, K4=0, K5=0 (not used)
Metric calculation	Based on **bandwidth of slowest link** + **sum of delay values** of all links
Router ID	172.16.1.14 (default: highest IP on physical interface)
Automatic summarization	Disabled
Maximum paths	4 (ECMP load-balancing default)
Networks	10.0.0.0 and 172.16.1.0/28
Passive interface	G2/0
Neighbors	R2 and R3
AD values	**90** for internal routes, **170** for external routes

> ***Note: K2, K4, and K5 are set to 0 by default and are not used, but that can be changed with configuration.***

> ***Note: Internal routes are normal EIGRP routes. External routes are routes from outside EIGRP which are then inserted into EIGRP — that's a more advanced topic for the CCNP level.***

EIGRP Router ID

How the Default Router ID is Determined (R1's Example)

- The professor's `show ip protocols` output shows R1's Router ID as **172.16.1.14**
- The professor walks through why:
  1. First check: Is there a manually configured router ID? → **No** (none configured yet)
  2. Second check: Is there a loopback interface? → **No** (no loopback interfaces configured)
  3. Third check: What is the highest IP address on any physical interface? → **172.16.1.14** on G2/0
     - So, G2/0's 172.16.1.14 became the router ID

Determination Priority

1. **Manual configuration** (highest priority)
2. **Highest IP address on any loopback interface**
3. **Highest IP address on any physical interface** (lowest priority)

> ***Note: Loopback interfaces are virtual interfaces inside the router (covered more in OSPF videos).***

> ***Note: The router ID isn't actually an IP address — it's a 32-bit number formatted like a dotted-decimal IP address, and you can change it to any 32-bit number.***

Configuring the Router ID

```
R1(config-router)#eigrp router-id ?
  A.B.C.D  EIGRP Router-ID in IP address format

R1(config-router)#eigrp router-id 1.1.1.1
```

```
Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
NSF-aware route hold timer is 240
Router-ID: 1.1.1.1
Topology : 0 (base)
```

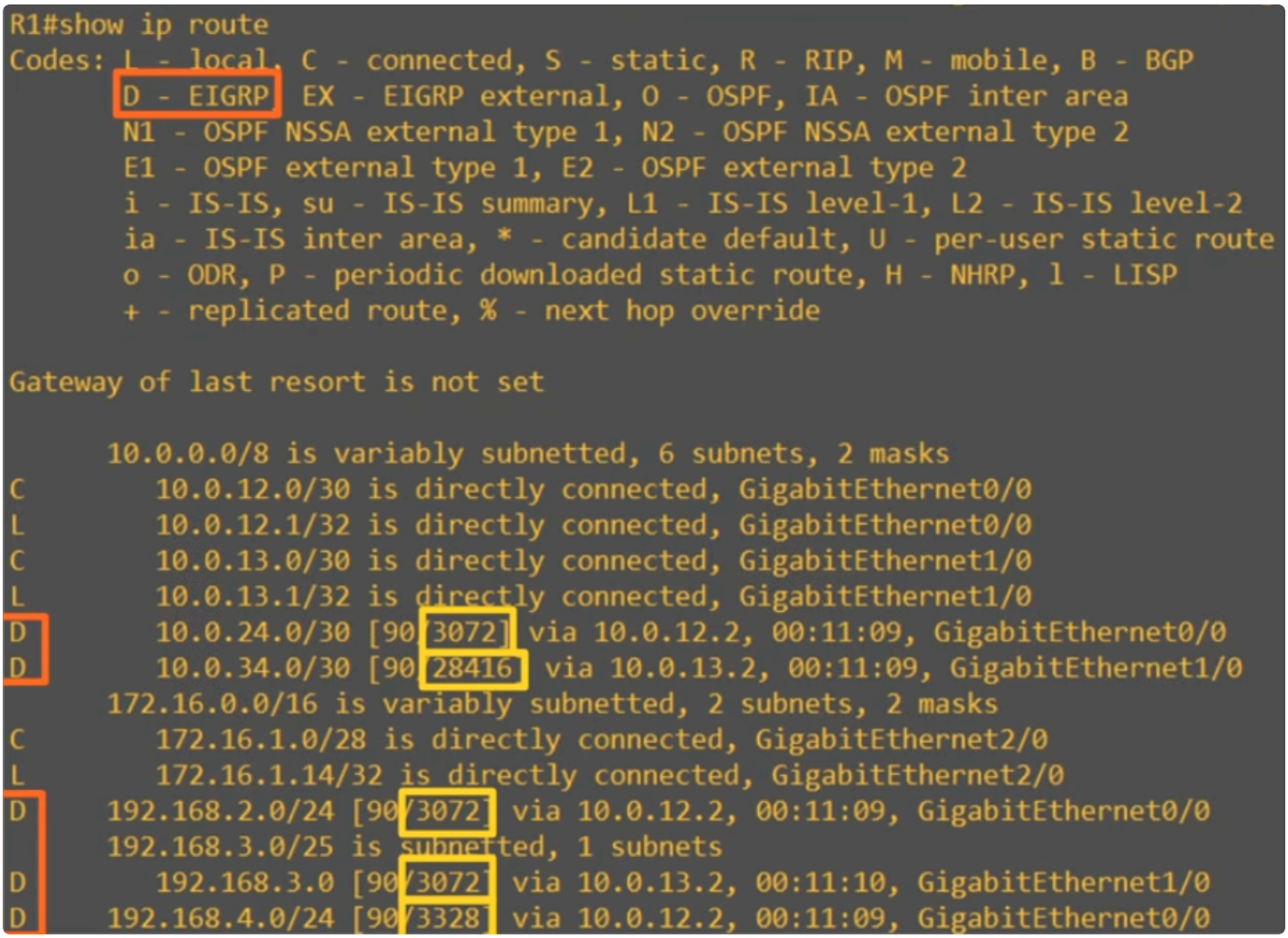
```
R1(config-router)# eigrp router-id 1.1.1.1
```

- Done from EIGRP configuration mode
- Now you can see that the router ID changes to 1.1.1.1, since manual configuration takes highest priority

---

EIGRP in the Routing Table

Key Observations



- EIGRP routes are indicated with the letter **D** (not E)
- Metric costs are much higher than OSPF and RIP (e.g., 3072, 3328, 28416)
- This is a very small network — in large networks, these numbers can get much larger
- That's perhaps a downside of EIGRP, the metrics are harder to understand