IPv4 Network layer Classfull and Classless Addressing

This talk will cover the basics of IP addressing and subnetting.

Topics covered will include:

- What is an IP Address?
- What are Classes?
- What is a Network Address?
- What are Subnet Masks and Subnet Addresses?
- How are Subnet Masks defined and used?
- How can all this be applied?
- What is CIDR?

The Network Layer (layer 3)

The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links). It ensures that each packet gets from its point of origin to its final destination. If two systems are connected to the same link, there is usually no need for a network layer.

In other word The Network layer is responsible for routing the packet based on its logical address. It also fragments and reassembles packets if necessary.

Internet Protocol (IP) Addresses (Logical Address)

What is IP Addressing?

Every host on a network needs to have a unique address, similar to you needing a unique address for your house. With this unique address, it is possible to send data from host to host. Every packet contains addressing information in the header, and the IP address in the header is used to route packets. If several people on your street had the same address, the post office would have a difficult time sorting mail. For a similar reason, IP addresses are unique on each network. IP addressing is simply configuring each host with a valid IP address. For access to the Internet, a host must have an IP address that identifies not only the host address (like a house number) but also identifies the network address (like a street number). An administrator needs to be aware of proper addressing techniques so that the hosts on the network will function correctly.

An IP address is a logical identifier for a computer or device on a network. The key feature of IP addresses is that they can be routed across networks.

The format of an IP address is a 32-bit numeric address written as four numbers separated by periods, sometimes referred to as a dotted-quad. The range of each number can be from 0 to 255.

For example, 2.165.12.230 would be a valid IP address. The four numbers in an IP address are used to identify a particular network and a host within that network.

Protocols looks at IP addresses in ***binary*** form, but as humans, we prefer to see IP addresses in ***decimal*** form. Because the protocol is seeing only binary, working with IP addresses makes more sense when you also look at the IP addresses in binary. To do so, you need to understand the two numbering systems (***binary*** *and* ***decimal***) and be able to convert from one to another.

There are many versions of IP the TCP/IP protocol: ***IP version 4*** and ***IP version 6***. *IP version 6* is much more complicated than *IP version 4* and is much newer. First, we will be working with *IP version 4* which is the address format of the four digits separated by full-stops.

IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (more than 4 billion). This means that, theoretically, if there were no restrictions, more than 4 billion devices could be connected to the Internet.

There are two types of addresses in the network layer of IPv4

1. **Classful Address**
2. **Classless Address**

1. Classful Address

IPv4 addresses Classes

IP addresses are divided into five classes:

Class A: Large networks.

Class B: Medium-sized networks.

Class C: Small networks with less than 256 devices.

Class D: Multicasting.

Class E: Reserved.

An IP address has two parts :

Network Address & Host Address (also known as local or node)

1. Class A Addresses

The designers of the IP address scheme said that the first bit of the first byte in a Class A network address must always be off, or 0. Thus a Class A address must be between 0 and127 inclusive.

Consider the following network address: 0xxxxxxx

If all the other 7 bits turned off, then turned on sequentially, the class A range of network addresses will be found:

00000000 = 0

01111111 = 127

So, a Class A network is defined in the first octet between 0 and 127, and it can't be less or more. In a Class A network address, the first byte is assigned to the network address and the three remaining bytes are used for the node addresses. The Class A format is:

NETWORK.HOST.HOST.HOST

For example, in the IP address 49.22.102.70, the 49 is the network address, and 22.102.70 is the host address. Every machine on this particular network would have the distinctive network address of 49.

Class A network addresses are 1 byte long, with the first bit of that byte reserved and the seven remaining bits available for manipulation (addressing). As a result, the maximum number of Class A networks that can be created is 128. Because each of the seven bit positions can either be a 0 or a 1, thus 27 or 128.

To complicate matters further, the network address of all 0s (00000000) is reserved to designate the default route. Additionally, the address 127, which is reserved for diagnostics, can't be used either, which means that you can really only use the numbers 1 to 126 to designate Class A network addresses. This means the actual number of usable Class A network addresses is 128 minus 2, or 126.

Each Class A address has three bytes (24-bit positions) for the host address of a machine. This means there are 2^{24} or 16,777,216 - unique combinations and, therefore, precisely that many possible unique node addresses for each Class A network. Because node addresses with the two patterns of all 0s and all 1s are reserved, the actual maximum usable number of nodes for a Class A network is 2^{24}

minus 2, which equals 16.777.214. Either way, that’s a huge number of hosts on a network segment of class A.

2. Class B Addresses

In a Class B network, the first bit of the first byte must always be turned on, but the second bit must always be turned off (**10**). If all the other 6 bits turned off and then turned on, the range of Class B network will be found:

10000000 = 128

10111111 = 191

This means that a Class B network is defined when the first byte is configured from 128 to 191.

In a Class B network address, the first 2 bytes are assigned to the 'network address', and the remaining 2 bytes are used for 'node addresses'. The format is:

NETWORK.NETWORK.HOST.HOST

For example, in the IP address 172.16.30.56, the network address portion is 172.16, and the node address portion is 30.56 .

With a network address being 2 bytes (8 bits each), there would be 216 unique combinations. But the Internet Protocol designers decided that all Class B network addresses should start with the binary digit 1, then 0. This leaves 14 bit positions to manipulate, and therefore 16.384 (that is, 214) unique Class B network addresses.

A Class B address uses two bytes for node addresses. This is 216 minus the two reserved patterns (all 0s and all 1s), for a total of 65.534 possible node addresses for each Class B network.

3. Class C Addresses

For Class C networks, the first two bits of the first octet always turned on, but the third bit can never be on (**110**). Following the same process as the previous classes, convert from binary to decimal to find the range. Here's the range for a Class C network:

11000000 = 192

11011111 = 223

So, if you see an IP address that starts at 192 and goes to 223, you'll know it is a Class C IP address.

The first 3 bytes of a Class C network address are dedicated to the network portion of the address, with only one measly byte remaining for the node address. The format is:

NETWORK.NETWORK.NETWORK.HOST

Using the example IP address 192.168.100.102, the network address is 192.168.100, and the node address is 102.

In a Class C network address, the first three bit positions are always the binary 110. The calculation is such: 3 bytes, or 24 bits, minus 3 reserved positions, leaves 21 positions. Hence, there are 221, or 2.097.152, possible Class C networks. Each unique Class C network has 1 byte to use for 'node addresses'. This leads to 28 or 256, minus the two reserved patterns of all 0s and all 1s, for a total of 254 node addresses for each Class C network.

To summarize the rules and equations for Class A, B, and C addressing:

A host ID cannot be all 0s.

A host ID cannot be all 1s.

To determine the number of networks that can be created, use the formula 2N, where *N* is the number of bits in the network portion of the address.

To determine the number of hosts that can be created, use the formula $2^N - 2$, where *N* is the number of bits in the host portion of the address.

4. Class D Addresses

The Class D address space is a radical departure from the first three classes. Unlike the previous three classes (A, B and C classes), this class does not adhere to the convention of dividing the bit string into network and host address subcomponents. This makes it rather difficult to use in uniquely identifying networked hosts in an internetwork. Quite simply, it cannot define a network address. This is by design; this uniquely flat address space is not used for endpoint addressing. Instead, ***it serves as a code that lets a host send single stream of IP packets to numerous destination machines simultaneously.***

In other words, these addresses are called ***multicast*** addresses, and they are invalid for any workstation or host to use. ***[Multicast****: A communication between a single sender and multiple receivers on a network. No one host can have this address, but several can receive data by listening to it.****]***

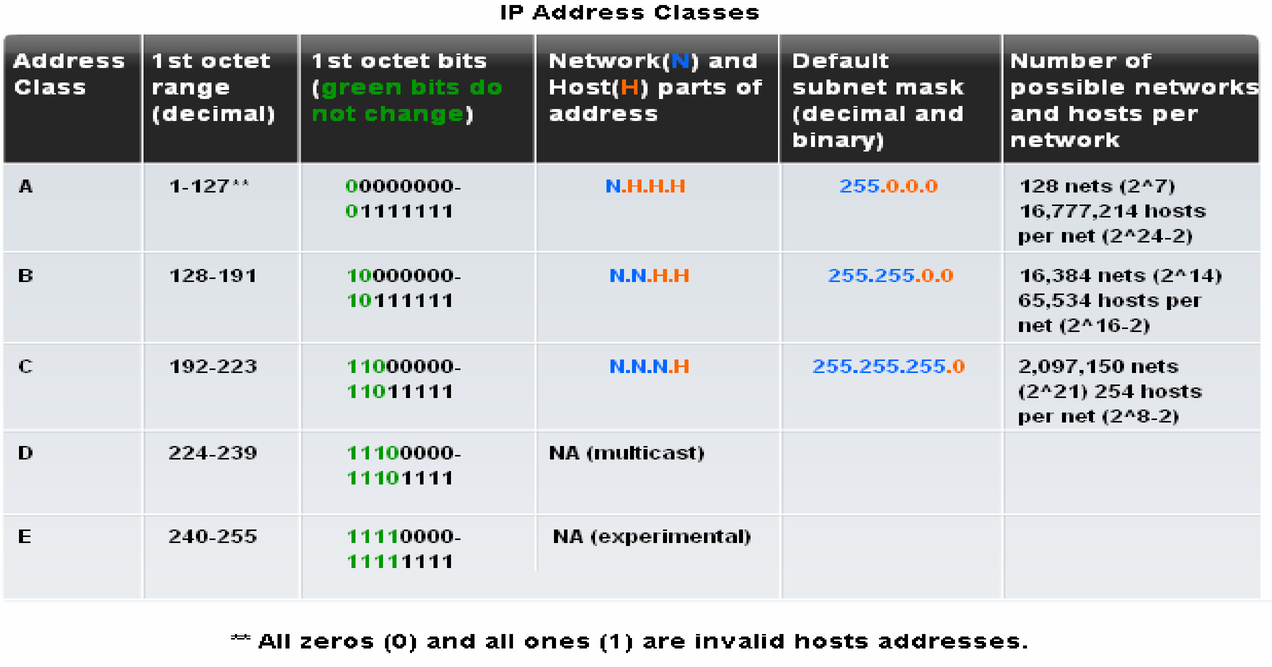
The purpose of a multicast address is *to enable a server somewhere to send data to a Class D address* that no one host has so that several hosts can listen to that address at the same time. When you are watching TV on the Internet or listening to the radio on the Internet, your computer is listening to a Class D address. No server is sending data directly to your workstation; instead, a server is sending data to the multicast address. Any host can use software to listen for data at that address, and many hosts can be listening at once.

The first 4 bits of a Class D address must be **1110**. Binary mathematics dictates that, given the location of this 0, the lowest address in this class can be 11100000, or 128 + 64 + 32 = 224. The highest mathematically possible address, given this constraint in the first octet, is 11101111, or 128 + 64 + 32 + 8 + 4 + 2 +

1 = 239. Thus, Class D multicast group addresses are from 224.0.0.0 to 239.255.255.255 .

5. Class E Addresses

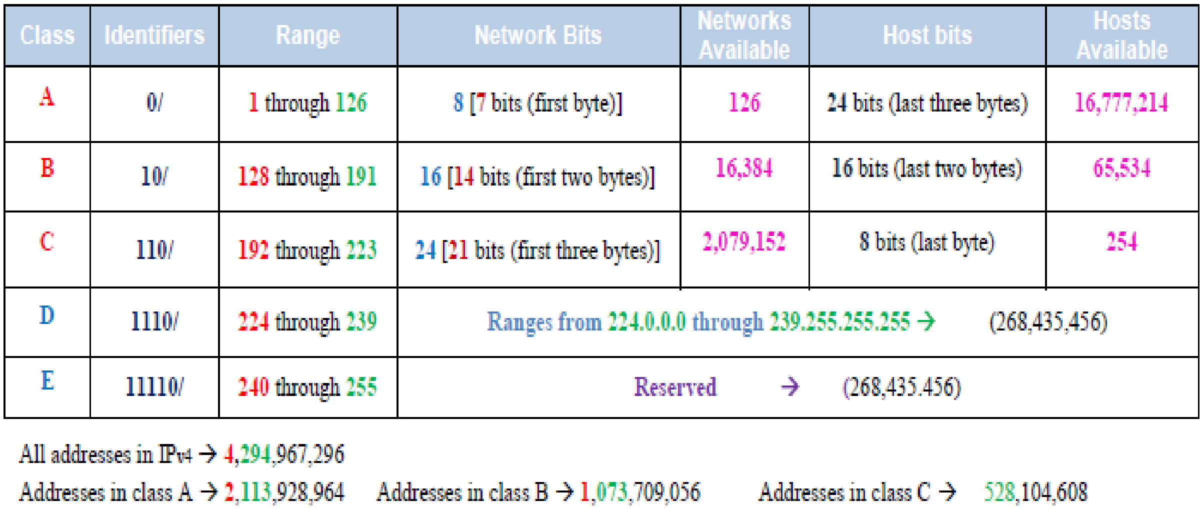
The last class of addresses is Class E. Class E addresses range from 240 to 255 in the first octet, and the 5 leftmost bits are **11110**. Class E addresses are reserved addresses and are invalid host addresses. They are used for experimental purposes by the IETF [**I**nternet **E**ngineering **T**ask **F**orce. *A governing body of the Internet*].



IP Address Classes

Address Class	1st octet range (decimal)	1st octet bits (green bits do not change)	Network(N) and Host(H) parts of address	Default subnet mask (decimal and binary)	Number of possible networks and hosts per network
A	1-127**	00000000- 01111111	N.H.H.H	255.0.0.0	128 nets (2^7) 16,777,214 hosts per net (2^24-2)
B	128-191	10000000- 10111111	N.N.H.H	255.255.0.0	16,384 nets (2^14) 65,534 hosts per net (2^16-2)
C	192-223	11000000- 11011111	N.N.N.H	255.255.255.0	2,097,150 nets (2^21) 254 hosts per net (2^8-2)
D	224-239	11100000- 11101111	NA (multicast)		
E	240-255	11110000- 11111111	NA (experimental)		

** All zeros (0) and all ones (1) are invalid hosts addresses.



Class	Identifiers	Range	Network Bits	Networks Available	Host bits	Hosts Available
A	0/	1 through 126	8 [7 bits (first byte)]	126	24 bits (last three bytes)	16,777,214
B	10/	128 through 191	16 [14 bits (first two bytes)]	16,384	16 bits (last two bytes)	65,534
C	110/	192 through 223	24 [21 bits (first three bytes)]	2,079,152	8 bits (last byte)	254
D	1110/	224 through 239	Ranges from 224.0.0.0 through 239.255.255.255 → (268,435,456)			
E	11110/	240 through 255	Reserved → (268,435.456)			

All addresses in IPv4 → 4,294,967,296

Addresses in class A → 2,113,928,964 Addresses in class B → 1,073,709,056 Addresses in class C → 528,104,608

All addresses are placed in a particular class based on the decimal values of their first octets. In the first octet, an IP address can start with a decimal value between 1 and 255. The system of class addresses has been set up to help ensure assignment of unique IP addresses.

Protocols implemented at the Network layer include:

- Internet Protocol version 4 (IPv4)
- Internet Protocol version 6 (IPv6)
- Novell Internetwork Packet Exchange (IPX)
- AppleTalk
- Connectionless Network Service (CLNS/DECNet)

Notation

There are three common notations to show an IPv4 address:

- Binary Notation (Base 2),
- Dotted-Decimal Notation (Base 256), and
- Hexadecimal Notation (Base 16).

The most prevalent, however, is base 256.

- ***Binary Notation:*** In binary notation, the IPv4 address is displayed as 32 bits. Each octet is often referred to as a byte. So it is common to hear an IPv4 address referred to as a 32-bit address or a 4-byte address. The following is an example of an IPv4 address in binary notation:

  01110101 10010101 00011101 00000010
- ***Dotted-Decimal Notation:*** To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes. The following is the Dotted-Decimal notation of the above address: 117.149.29.2
- ***Hexadecimal Notation:*** We sometimes see an IPv4 address in **hexadecimal notation.** Each hexadecimal digit is equivalent to four bits. This means that a 32-bit address has 8 hexadecimal digits. This notation is often used in network programming.

Within the address range of each IPv4 network, we have three types of addresses:

Network address - The address by which we refer to the network.

Broadcast address - A special address used to send data to all hosts in the network.

Host addresses - The addresses assigned to the end devices in the network.

Each network has an Internet address. Each network also must know the address of every other network with which it communicates.

After the network is identified, the specific host or node must be specified. A unique host address for the particular network is added to the end of the IP address.

Network Masks

A network mask helps you know which portion of the address identifies the network and which portion of the address identifies the node. Class A, B, and C networks have default masks, also known as natural masks, as shown here:

Class A: 255.0.0.0

Class B: 255.255.0.0

Class C: 255.255.255.0

Private Addresses

The private address blocks are:

10.0.0.0	**to**	**10.255.255.255 (10.0.0.0 /8)**	**for class A**
172.16.0.0	**to**	**172.31.255.255 (172.16.0.0 /12)**	**for class B**
192.168.0.0	**to**	**192.168.255.255 (192.168.0.0 /16)**	**For class C**

2. Classless Addressing

To overcome address depletion and give more organizations access to the Internet, classless addressing was designed and implemented. In this scheme, there are no classes, but the addresses are still granted in blocks.

Classless **I**nter-**D**omain **R**outing (**CIDR**) is an IP addressing scheme that was developed after the class system of A, B, C, D, and E [*uses a slash followed by a number to highlight the network portion of an address instead of using a subnet mask*].

The traditional class system considers an IP address as four octets with the network portion of the address highlighted by a subnet mask. The standard network portion is the first octet, the first two octets, or the first three octets. CIDR addressing still represents IP addresses in the traditional dotted decimal notation, but highlights the network portion with a slash followed by a number. For example:

192.168.3.15/26

172.21.165.1/19

The number after the slash is the number of bits that represent the network portion of the IP address. ***CIDR was developed to increase the efficiency of address allocation and to alleviate overloaded Internet routers.***

Computer Networks Fundamentals II

Lecture2:

Prefix and Subnet Mask

Assist Prof Dr. Ahmed Mahdi Al-salih

Assist Prof Dr. Suad Alasadi

IP Address

- An IP address is a 32-bit, two-level hierarchical number.it is uniquely defined by a network layer address.
- It is hierarchical because the first portion of the address represents the **network**, and the second portion of the address represents the **node (or host).**
- The 32 bits are grouped into four octets, with 8 bits per octet. The value of each octet ranges from 0 to 255 decimal, or 00000000 to 11111111 binary. IP addresses are usually written in dotted decimal notation, which means that each octet is written in decimal notation and dots are placed between the octets.

Network Prefixes

- "How do you know how many bits of this address represent the network portion and how many bits represent the host portion?"
- The answer is the *prefix*.

- When an IPv4 network address is expressed, you add a ***prefix length*** to the network address.

- **This prefix length is** the number of bits in the address that gives the network portion. This prefix length is written in ***slash format***. That is a forward slash (/) followed by the number of network bits.

Network Prefixes(Con.)

- For example, in 172.16.4.0 **/24**,

  the **/24** is the prefix length.

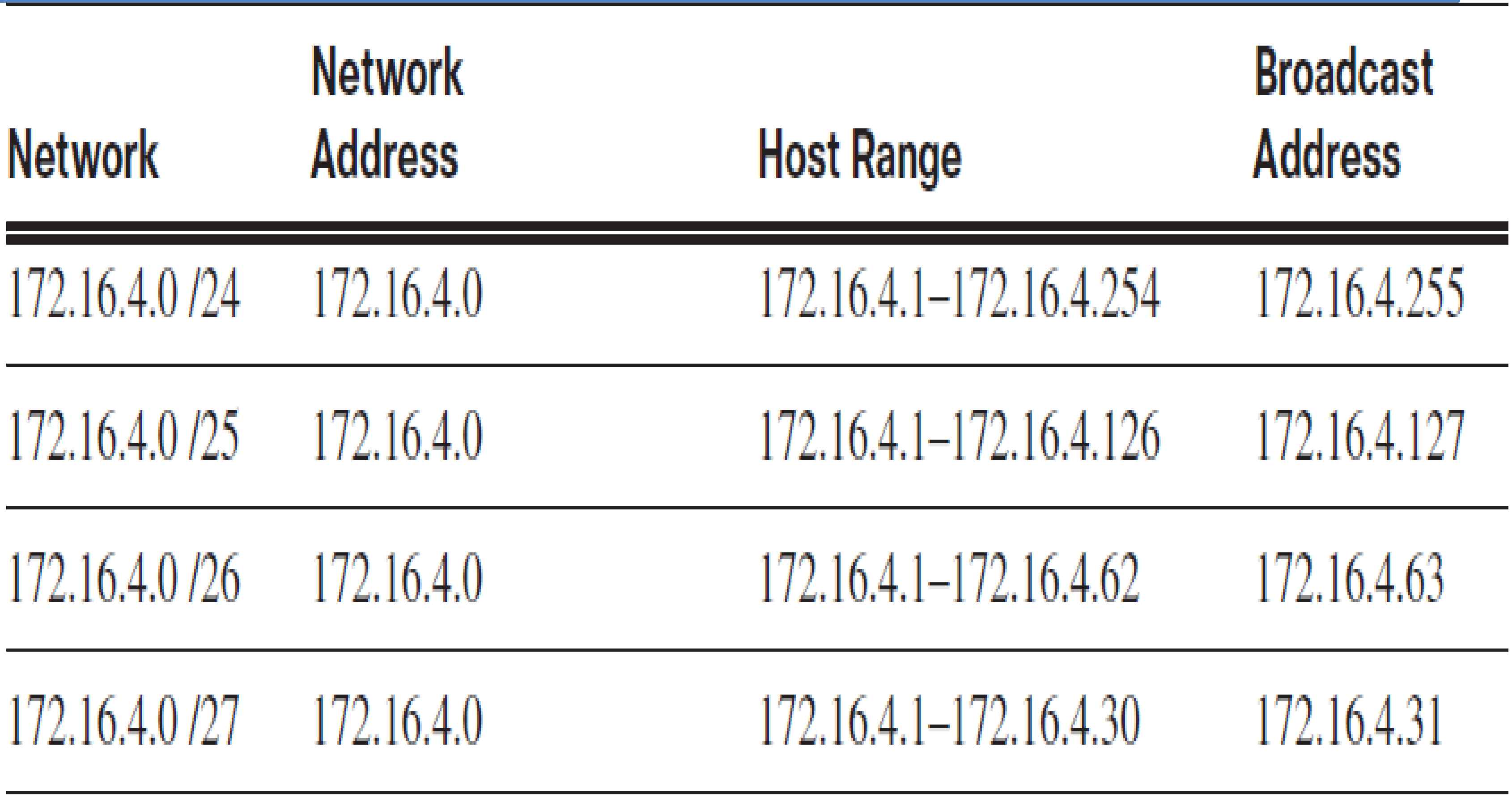
This tells you that the first **24** bits are the **network address**. The remaining **8 bits**, the last octet, are the **host portion.**

Network Prefixes (Con.)

- Depending on **the number of hosts** on the network, the prefix assigned can be different. Having a different prefix number changes the host range and broadcast address for each network.
- Notice that the network addresses in **Table 1** remain the same, but the host range and the broadcast address are different for the different prefix lengths. You can also see that the number of hosts that can be addressed on the network changes as well.

Network Prefixes (Con.)

Table 1: Using Different Prefixes for the 172.16.4.0 Network



Network	Network Address	Host Range	Broadcast Address
172.16.4.0 /24	172.16.4.0	172.16.4.1–172.16.4.254	172.16.4.255
172.16.4.0 /25	172.16.4.0	172.16.4.1–172.16.4.126	172.16.4.127
172.16.4.0 /26	172.16.4.0	172.16.4.1–172.16.4.62	172.16.4.63
172.16.4.0 /27	172.16.4.0	172.16.4.1–172.16.4.30	172.16.4.31

Subnet Mask

- “How do the **network devices** know how many bits are the network portion and how many bits are the host portion?”

- The answer to this question is the **subnet mask.**

- **The subnet mask is used to Define the Network and Host Portions of the Address.**

Prefix and subnet

- The **prefix and the subnet mask** are different ways of representing the same information: the **network portion** of an address.

- **The prefix length** tells you the number of bits in the address that are the network portion in a way that is easier to communicate to humans.

- The **subnet mask** is used in data networks to define this network portion for the devices.

Subnet Mask(con.)

- The **subnet mask is** a **32-bit value** used with the IPv4 address that specifies the **network portion** of the address to the network devices.
- The subnet mask uses 1s and 0s to indicate which bits of the IPv4 address are network bits and which bits are hosts bits.
- The subnet mask is expressed in the same dotted decimal format as the IPv4 address.

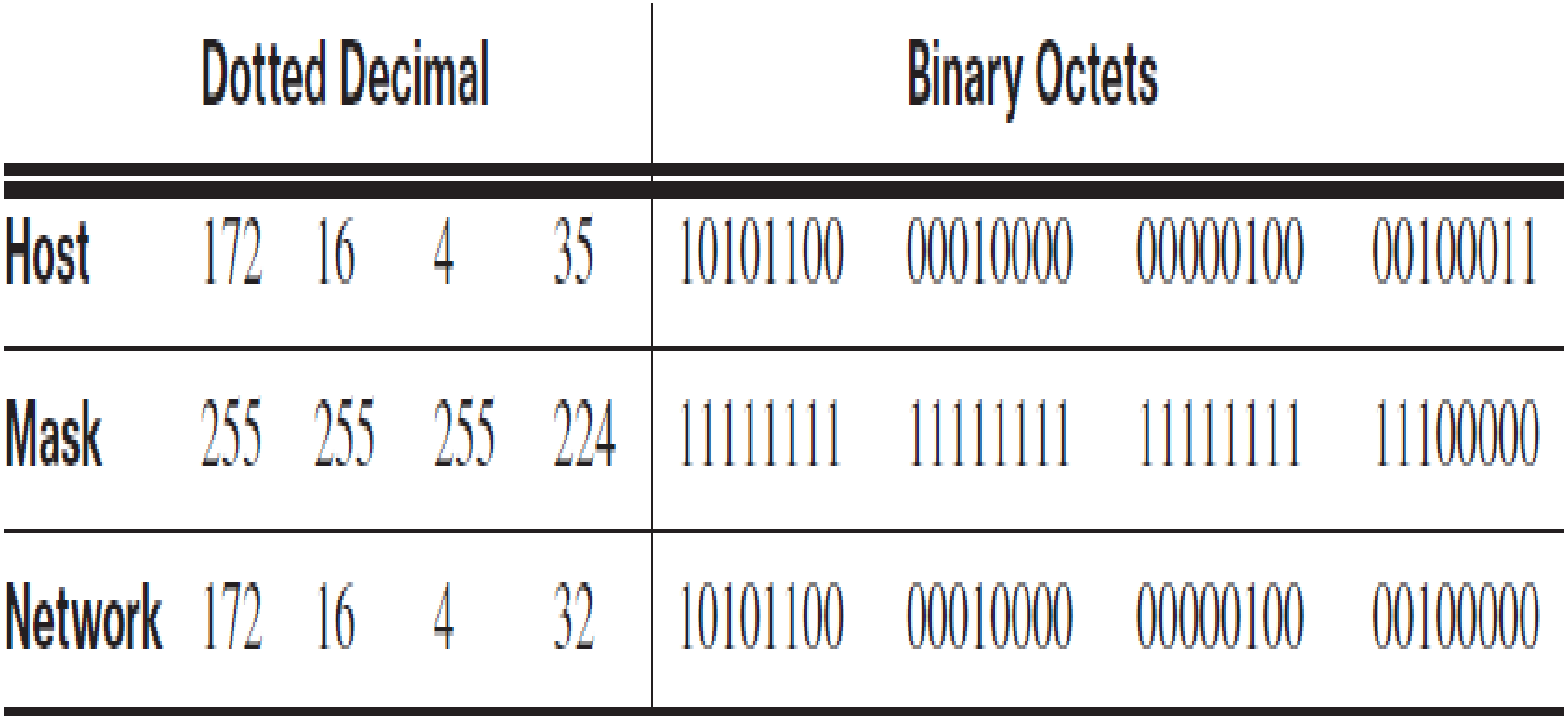
Subnet Mask(con.)

- For example **A /24** prefix represents a **subnet mask** of **255.255.255.0**
- (11111111.11111111.11111111.00000000).

- The first three octets, the higher-order 24 bits, are all 1s. The remaining low-order bits of the subnet mask are 0s, indicating the host address within the network.

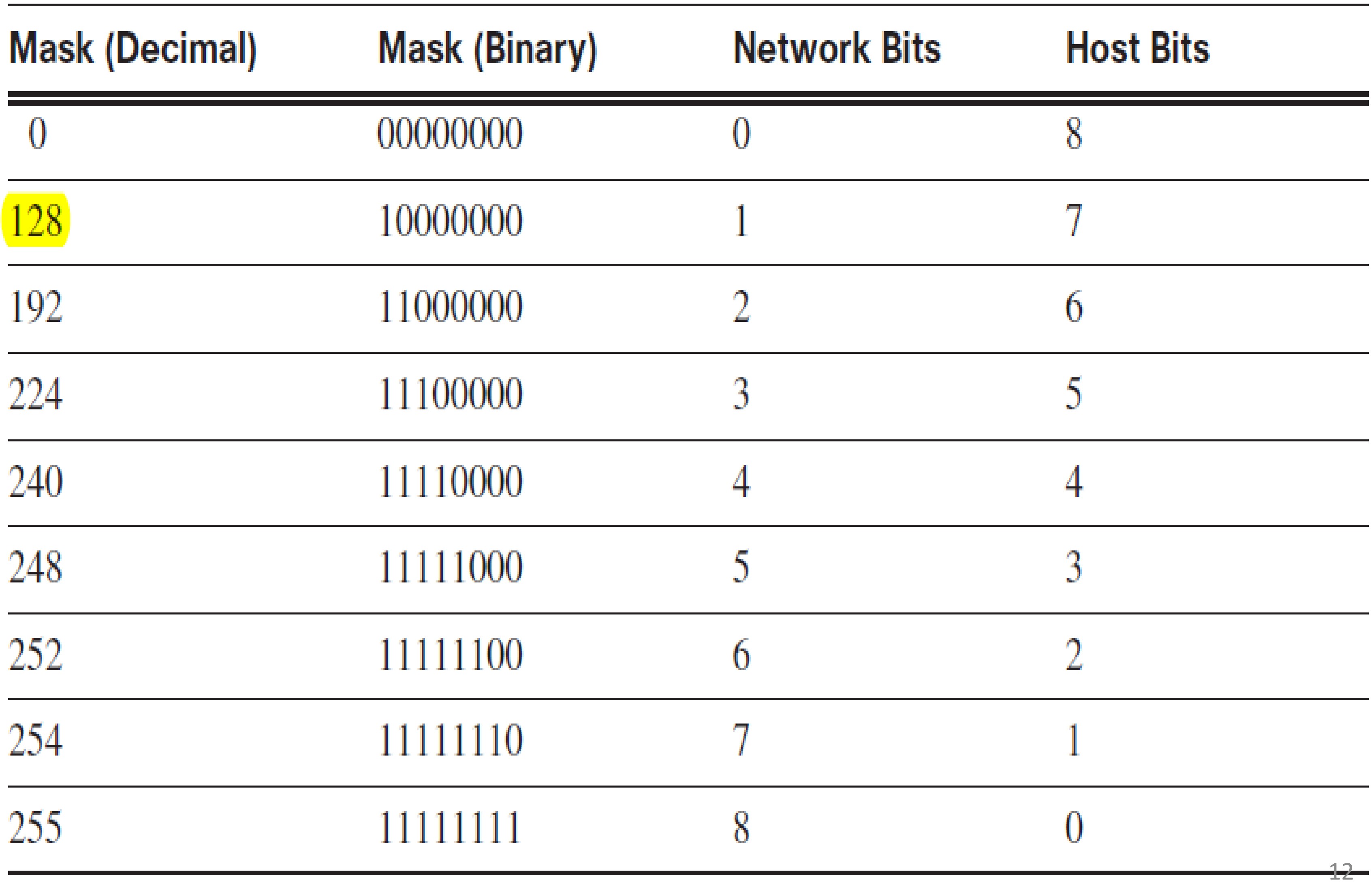
Subnet Mask(con.)

- For example, examine the host 172.16.4.35/27 shown in Table 2.



	Dotted Decimal				Binary Octets			
Host	172	16	4	35	10101100	00010000	00000100	00100011
Mask	255	255	255	224	11111111	11111111	11111111	11100000
Network	172	16	4	32	10101100	00010000	00000100	00100000

Subnet Mask(con.)



Mask (Decimal)	Mask (Binary)	Network Bits	Host Bits
0	00000000	0	8
128	10000000	1	7
192	11000000	2	6
224	11100000	3	5
240	11110000	4	4
248	11111000	5	3
252	11111100	6	2
254	11111110	7	1
255	11111111	8	0

Subnet Mask

- Using Subnet Mask to determine the Network address , Broadcast Address, First IP Address, Last IP Address, from host :

172 . 16 . 132 . 70 /20

172 . 16 . 132 . 70 **Host IP Add.**

10101100 . 00010000 . 10000100 . 01000110

255. 255. 240. 0 **Subnet Mask** **Anding**

11111111 . 11111111 . 11110000 . 00000000

10101100 . 00010000 . 10000000 . 00000000

172 . 16 . 128 . 0 **Network Address**

Subnet Mask

Broadcast Address

172 . 16 . 128 . 0

+

0 . 0. 15 . 255

172 . 16 . 143 . 255

Network Address

Wild Cast Address

Broadcast Address

Note : **Wild Cast Address complement of Subnet Mask**

Network Address = 172 . 16 .128 . 0

First IP Address = 172.16.128.1

Range of IP Addresses = **172.16.128.1 To 172.16.143.254**

Broadcast Address = 172. 16. 143. 255

Thank You

Subnetting

Understanding Subnetting

Subnetting allows you to create multiple logical networks that exist within a single Class A, B, or C network. If you do not subnet, you will only be able to use one network from your Class A, B, or C network, which is unrealistic.

Formula for calculating subnets

Use this formula to calculate the number of subnets:

$2^{\wedge n}$ where n = the number of bits borrowed

The number of hosts

To calculate the number of hosts per network, we use the formula of $2^{\wedge h}$ - 2 where h = the number of bits left for hosts.

Each data link on a network must have a unique network ID, with every node on that link being a member of the same network. If you break a major network (Class A, B, or C) into smaller subnetworks, it allows you to create a network of interconnecting subnetworks. Each data link on this network would then have a unique network/subnetwork ID. Any device, or gateway, connecting *n* networks/subnetworks has *n* distinct IP addresses, one for each network / subnetwork that it interconnects.

To subnet a network, extend the natural mask using some of the bits from the host ID portion of the address to create a subnetwork ID. For example, given a Class C network of 204.15.5.0 which has a natural mask of 255.255.255.0, you can create subnets in this manner:

204.15.5.0 - 11001100.00001111.00000101.00000000
255.255.255.224 - 11111111.11111111.11111111.11100000
---|sub|------

By extending the mask to be 255.255.255.224, you have taken three bits (indicated by "sub") from the original host portion of the address and used them to make subnets. With these three bits, it is possible to create eight subnets. With the remaining five host ID bits, each subnet can have up to 32 host addresses, 30 of which can actually be assigned to a device *since host ids of all zeros or all ones are not allowed* (it is very important to remember this). So, with this in mind, these subnets have been created.

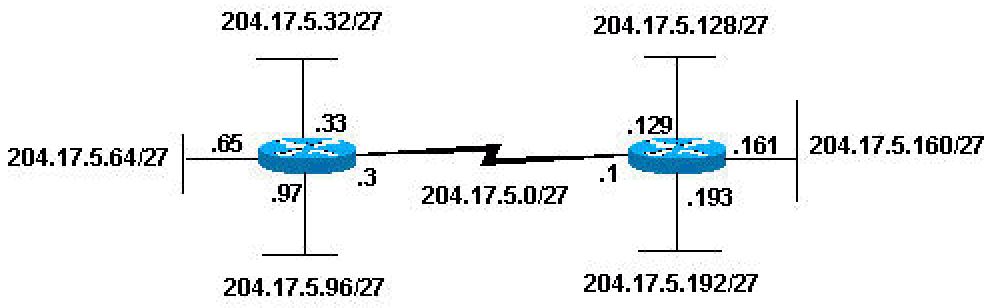
204.15.5.0	255.255.255.224	host address range 1 to 30
204.15.5.32	255.255.255.224	host address range 33 to 62
204.15.5.64	255.255.255.224	host address range 65 to 94
204.15.5.96	255.255.255.224	host address range 97 to 126
204.15.5.128	255.255.255.224	host address range 129 to 158
204.15.5.160	255.255.255.224	host address range 161 to 190
204.15.5.192	255.255.255.224	host address range 193 to 222
204.15.5.224	255.255.255.224	host address range 225 to 254

Note: There are two ways to denote these masks. First, since you are using three bits more than the "natural" Class C mask, you can denote these addresses as having a 3-bit subnet mask. Or, secondly, the mask of 255.255.255.224 can also be denoted as /27 as there are 27 bits that are set in the mask. This second method is used with CIDR. Using this method, one of thse networks can be described with the notation prefix/length. For example, 204.15.5.32/27 denotes the network

204.15.5.32 255.255.255.224. When appropriate the prefix/length notation is used to denote the mask throughout the rest of this document.

The network subnetting scheme in this section allows for eight subnets, and the network might appear as:

Figure 2



Notice that each of the routers in Figure 2 is attached to four subnetworks, one subnetwork is common to both routers. Also, each router has an IP address for each subnetwork to which it is attached. Each subnetwork could potentially support up to 30 host addresses.

This brings up an interesting point. The more host bits you use for a subnet mask, the more subnets you have available. However, the more subnets available, the less host addresses available per subnet. For example, a Class C network of 204.17.5.0 and a mask of 255.255.255.224 (/27) allows you to have eight subnets, each with 32 host addresses (30 of which could be assigned to devices). If you use a mask of 255.255.255.240 (/28), the break down is:

204.15.5.0 - 11001100.00001111.00000101.00000000

```
255.255.255.240 - 11111111.11111111.11111111.11110000
                  ---------------------------------------------------|sub |---
```

Since you now have four bits to make subnets with, you only have four bits left for host addresses. So in this case you can have up to 16 subnets, each of which can have up to 16 host addresses (14 of which can be assigned to devices).

Take a look at how a Class B network might be subnetted. If you have network 172.16.0.0 ,then you know that its natural mask is 255.255.0.0 or 172.16.0.0/16. Extending the mask to anything beyond 255.255.0.0 means you are subnetting. You can quickly see that you have the ability to create a lot more subnets than with the Class C network. If you use a mask of 255.255.248.0 (/21), how many subnets and hosts per subnet does this allow for?

```
172.16.0.0  -      10101100.00010000.00000000.00000000
255.255.248.0 - 11111111.11111111.11111000.00000000
                    --------------------------| sub |-------------------
```

You are using five bits from the original host bits for subnets. This will allow you to have 32 subnets (2^5). After using the five bits for subnetting, you are left with 11 bits for host addresses. This will allow each subnet so have 2048 host addresses (2^{11}), 2046 of which could be assigned to devices.

Note: In the past, there were limitations to the use of a subnet 0 (all subnet bits are set to zero) and all ones subnet (all subnet bits set to one). Some devices would not allow the use of these subnets. Cisco Systems devices will allow the use of these subnets when the **ip subnet zero** command is configured.

Examples

Sample Exercise 1

Now that you have an understanding of subnetting, put this knowledge to use. In this example, you are given two address / mask combinations, written with the prefix/length notation, which have been assigned to two devices. Your task is to determine if these devices are on the same subnet or different subnets. You can do this by using the address and mask of each device to determine to which subnet each address belongs.

Device A: 172.16.17.30/20
Device B: 172.16.28.15/20

Determining the Subnet for Device A:

```
172.16.17.30  -    10101100.00010000.00010001.00011110
255.255.240.0 -    11111111.11111111.11110000.00000000
                  ---------------------------| sub|---------------------
subnet =          10101100.00010000.00010000.00000000 = 172.16.16.0
```

Looking at the address bits that have a corresponding mask bit set to one, and setting all the other address bits to zero (this is equivalent to performing a logical "AND" between the mask and address), shows you to which subnet this address belongs. In this case, DeviceA belongs to subnet 172.16.16.0.

Determining the Subnet for Device B:

```
172.16.28.15  -   10101100.00010000.00011100.00001111
```

```
255.255.240.0 -   11111111.11111111.11110000.00000000
                  ---------------------------| sub|------------
subnet =        10101100.00010000.00010000.00000000 = 172.16.16.0
```

From these determinations, DeviceA and DeviceB have addresses that are part of the same subnet.

Subnetting

The procedure in which we browse some Host bits from the Host portion and add it in Network bits in Network Portion this procedure is called Subnetting. In Subnetting we increase the Network Portion and decrease the Host Portion.

It means as Network Portion increases option of Sub networks also increases but the no of Hosts in each network start to decrease.

For Example:

124.192.135.159 as we know an IP address is of 32 Bits

- **It Is an IP address of Class A (Because in First Octet Value range is 0 – 126)**
- **Its Network Portion is of 8 bits**
- **Its Host Portion is of 24 bits**
- **Its Subnet mask is 255.0.0.0**
- **Its Network ID is 124.0.0.0**

Note:

IP addresses are also represented as (IP/Network Bits)

Class A	→	124.192.135.159/8	→	In Class A Network Portion is of 8 Bits
Class B	→	189.200.191.239/16	→	In Class B Network Portion is of 16 Bits
Class C	→	193.220.164.223/24	→	In Class C Network Portion is of 24 Bits

200.100.100.0/24

(IP of Class C + Network Portion = 24 Bits + Host Portion = 8 Bits)

Let us Borrow 2-Bits from Host Portion (which have Host Portion = 8-Bits)

Now Host Portion have Total Bits = 6

Add it in Network Portion (which have Network Portion = 24-Bits)

Now Network Portion have Total Bits = 26

What is Custom Sub net Mask

After Subnetting Default Subnet Mask of the IP address becomes Custom Subnet Mask.

How many Subnets can form when we borrow some Bits from Host portion and add it in Network Portion?

We can produce 2^n (n = is equal to number of bits Borrow from Host Portion)

If we Borrow 1 Bit → 2^n → n = 1 → 2^1 → 2 → Subnets can form

If we Borrow 2 Bits → 2^n → n = 2 → 2^2 → 4 → Subnets can form

If we Borrow 3 Bits → 2^n → n = 3 → 2^3 → 8 → Subnets can form

If we Borrow 4 Bits → 2^n → n = 4 → 2^4 → 16 → Subnets can form

If we Borrow 5 Bits → 2^n → n = 5 → 2^5 → 32 → Subnets can form

If we Borrow 6 Bits → 2^n → n = 6 → 2^6 → 64 → Subnets can form

How may Maximum Bits be able to borrow from Host Portion?

We can borrow maximum 6 bits from Host Portion

Note:

If number of subnets increases congestion in the routing tables would be happened because size of routing tables increases but if Bandwidth is sufficient no congestion will take place.

Types of Subnetting

There are two types of Subnetting

1	**Fixed Length Subnetting**	**or**	**Fixed Length Subnet Masking**
2	**Variable Length Subnetting**	**or**	**Variable Length Subnet Masking**

1 <u>Fixed Length Subnet Masking</u>

If we are sure that we have limit of subnets and not increases from the limit it is called Fixed Length Subnetting

2 <u>Variable Length Subnet Masking</u>

Question: How Many Subnets can form using following IP Address we do sub netting of 2-Bits and How many Hosts Can be in each subnet?

<u>200.100.100.0</u>

Answer: Given IP address is 200.100.100.0

1	**Class**	➔	**C**
2	**Network Portion**	➔	**24-Bits**

3	Host Portion	→	08-Bits				
4	Representation of IP	→	200.100.100.0/24				
5	Subnetting	→	2-Bits				
6	After Subnetting Network Portion	→	26-Bits				
7	After Subnetting Host Portion	→	06-Bits				
8	After Subnetting IP	→	200.100.100.0/26				
9	Number of Subnets	→	2^n	n=2	2^2	→	4
10	Number of Hosts per Subnet	→	2^n	n=6	2^6	→	64

Default Subnet Mask

Default Subnet Mask of Class Full IP Address 200.100.100.0/24

As we know to find out the Default Subnet Mask we ON all 24-Bits of Network Portion of Class Full IP address

200 . 100 . 100 . 0/24

11111111. 11111111. 1111111. 0

255 . 255 . 255 . 0

Network ID of Class Full IP 200.100.100.0/24

200.100.100.0

Custom Subnet Mask

Custom Subnet Mask of Class Less IP Address 200.100.100.0/26

As we Know to find out the Custom Subnet Mask we ON all 26-Bits of Network Portion of class Less IP Address

200 . 100 . 100 . 0/26

11111111. 11111111. 11111111. 11000000

255 . 255 . 255 . 192

Subnet ID of Given Class Less IP 200.100.100.0/26

200 . 100 . 100 . 65

200 . 100 . 100 . 01000001

255 . 255 . 255 . 11000000

200 . 100 . 100 . 01000000

200 . 100 . 100 . 64

<u>200.100.100.65/26</u>

Custom Subnet Mask
200.100.100.65
100.100.100.01000001
255.255.255.11000000
255.255.255.192
100.100.100.01000000
200.100.100.6

Computer Networks Fundamentals

Lecture 10:

Route Summarization

Lecturer : Assist Prof Dr. Suad A. Alasadi

Hierarchical Addressing

** Each IP address is divided into a prefix and a suffix.

Prefix identifies network to which computer is attached -

- **Suffix** identifies computer within that network

** Address format makes routing efficient

Hierarchical Addressing

The IP addressing scheme is hierarchical, and IP routers make hierarchical decisions.

Recall that an IP address comprises a prefix part and a host part (suffix).

A router has to know only how to reach the next hop; it does not have to know the details of how to reach an end node that is not local.

Routers use the prefix to determine the path for a destination address that is not local. The host part is used to reach local hosts.

Route Summarization

With **route summarization**, also referred to as **route aggregation** or **supernetting**, one route in the routing table represents many other routes.

Summarizing routes reduces the routing update traffic and reduces the number of routes in the routing table and overall router overhead in the router receiving the routes.

In a hierarchical network design, effective use of route summarization can limit the impact of topology changes to the routers in one section of the network.

CIDR

Classless Inter-Domain Routing (CIDR) is a mechanism developed to help alleviate the problem of IP address exhaustion and growth of routing tables.

The idea behind CIDR is that blocks of multiple addresses (for example, blocks of Class C address) can be combined, or aggregated, to create a larger (that is, more hosts allowed) classless set of IP addresses. Blocks of Class C network numbers are allocated to each network service provider; organizations using the network service provider for Internet connectivity are allocated subsets of the service provider’s address space as required. These multiple Class C addresses can then be summarized in routing tables, resulting in fewer route advertisements.

The CIDR mechanism can be applied to blocks of Class A, B, and C addresses; it is not restricted to Class C.)

Route Summarization

For summarization to work correctly, the following requirements must be met:

■ Multiple IP addresses must share the same leftmost bits.

■ Routers must base their routing decisions on a 32-bit IP address and a prefix length of up to 32 bits.

■ Routing protocols must carry the prefix length with the 32-bit IP address.

Route Summarization

For example, assume that a router has the following networks behind it:

192.168.168.0/24
192.168.169.0/24
192.168.170.0/24
192.168.171.0/24
192.168.172.0/24
192.168.173.0/24
192.168.174.0/24
192.168.175.0/24

Each of these networks could be advertised separately; however, this would mean advertising eight routes. Instead, this router can summarize the eight routes into one route and advertise 192.168.168.0/21.

By advertising this one route, the router is saying, “Route packets to me if the destination has the first 21 bits the same as the first 21 bits of 192.168.168.0.”

Route Summarization

The following figure illustrates how this summary route is determined. The addresses all have the first 21 bits in common and include all the combinations of the other 3 bits in the network portion of the address; therefore, only the first 21 bits are needed to determine whether the router can route to one of these specific addresses.

192.168.168.0 =	11000000 10101000 10101 000 00000000
192.168.169.0 =	11000000 10101000 10101 001 00000000
192.168.170.0 =	11000000 10101000 10101 010 00000000
192.168.171.0 =	11000000 10101000 10101 011 00000000
192.168.172.0 =	11000000 10101000 10101 100 00000000
192.168.173.0 =	11000000 10101000 10101 101 00000000
192.168.174.0 =	11000000 10101000 10101 110 00000000
192.168.175.0 =	11000000 10101000 10101 111 00000000

Number of Common Bits = 21
Number of Non-Common Network Bits = **3**
Number of Host Bits = **8**

Benefits of Hierarchical Addressing

A network designer decides how to implement the IP addressing hierarchy based on the network's size, geography, and topology. In large networks, hierarchy within the IP addressing plan is mandatory for a stable network (including stable routing tables). For the following reasons, a planned, hierarchical IP addressing structure, with room for growth, is recommended for networks of all sizes:

1- Influence of IP addressing on routing: An IP addressing plan influences the network's overall routing. Before allocating blocks of IP addresses to various parts of the network and assigning IP addresses to devices, consider the criteria for an appropriate and effective IP addressing scheme. Routing stability, service availability , and network scalability are some crucial and preferred network characteristics and are directly affected by IP address allocation and deployment.

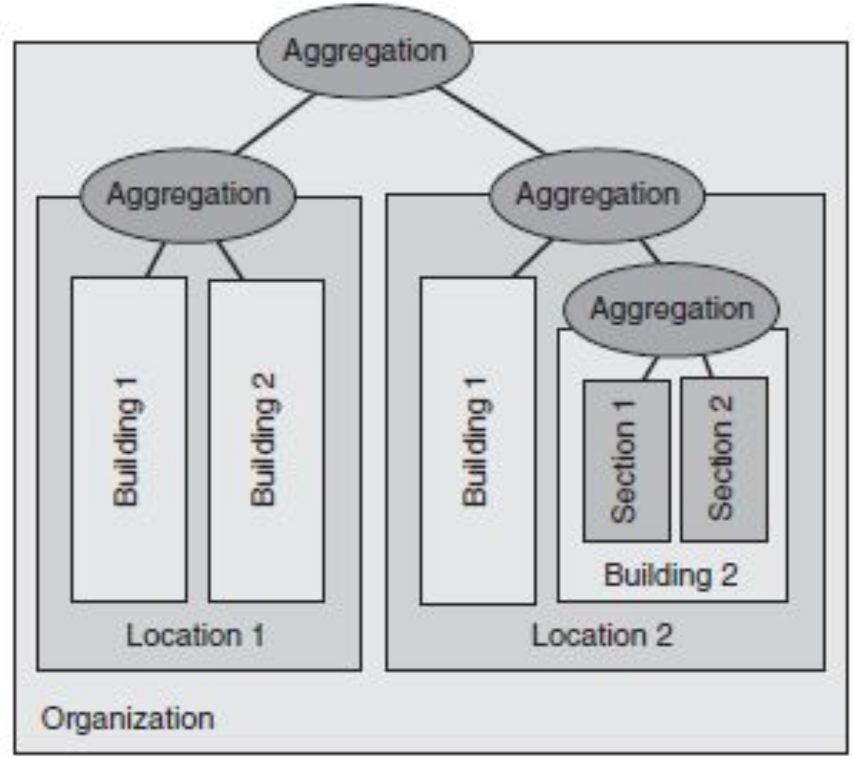
Benefits of Hierarchical Addressing

2- Modular design and scalable solutions: Whether building a new network or adding a new service on top of an existing infrastructure, a modular design helps to deliver a long-term, scalable solution. IP addressing modularity allows the aggregation of routing information on a hierarchical basis.

3- Route aggregation: Route aggregation is used to reduce routing overhead and improve routing stability and scalability. However, to implement route aggregation, a designer must be able to divide a network into contiguous IP address areas and must have a solid understanding of IP address assignment, route aggregation, and hierarchical routing.

Summarization Groups

To reduce the routing overhead in a large network, a multilevel hierarchy might be required. The depth of hierarchy depends on the network size and the size of the highest-level summarization group. The following figure shows an example of a network hierarchy.



Summarization Groups

A typical organization has up to three levels of hierarchy:

■ **First level:** **Network locations** typically represent the first level of hierarchy in enterprise networks. Each location typically represents a group of summarized subnets, known as a summarization group.

■ **Second level:** A second level of hierarchy can be done within first-level summarization groups. For example, a large location can be divided into smaller summarization groups that represent the **buildings within that location.** Not all first-level summarization groups require a second level of hierarchy.

■ **Third level:** To further minimize the potential routing overhead and instability, a third level of hierarchy can exist within the second-level summarization group. For example, **sections or floors within individual buildings** can represent the third-level summarization group.

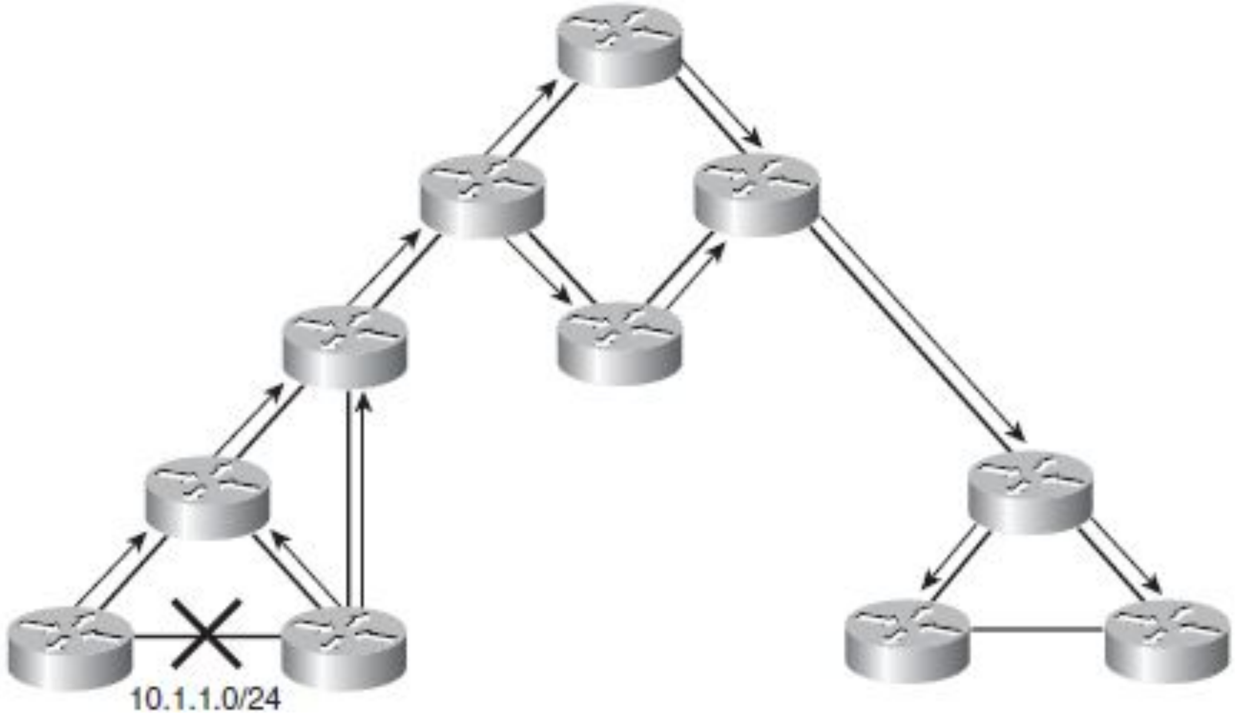
Impact of Poorly Designed IP Addressing

A poorly designed IP addressing scheme usually results in IP addresses that are randomly assigned on an as-needed basis. In this case, the IP addresses are most likely dispersed through the network with no thought as to whether they can be grouped or summarized. A poor design provides no opportunity for dividing the network into contiguous address areas, and therefore no means of implementing route summarization.

The next figure is a sample network with poorly designed IP addressing; it uses a dynamic routing protocol. Suppose that a link in the network is flapping (changing its state from UP to DOWN, and vice versa) ten times per minute. Because dynamic routing is used, the routers that detect the change send routing updates to their neighbors, those neighbors send it to their neighbors, and so on. Because aggregation is not possible, the routing update is propagated throughout the entire network, even if there is no need for a distant router to have detailed knowledge of that link.

Impact of Poorly Designed IP Addressing

A Poorly Designed IP Addressing Scheme Results in Excess Routing Traffic



Impact of Poorly Designed IP Addressing

Impacts of poorly designed IP addressing include the following:

■ **Excess routing traffic consumes bandwidth**: When any route changes, routers send routing updates. Without summarization, more updates are sent, and the routing traffic consumes more bandwidth.

■ **Increased routing table recalculation**: Routing updates require routing table recalculation, which affects the router's performance and ability to forward traffic.

■ **Possibility of routing loops**: When too many routing changes prevent routers from converging with their neighbors, routing loops might occur, which might have global consequences for an organization.

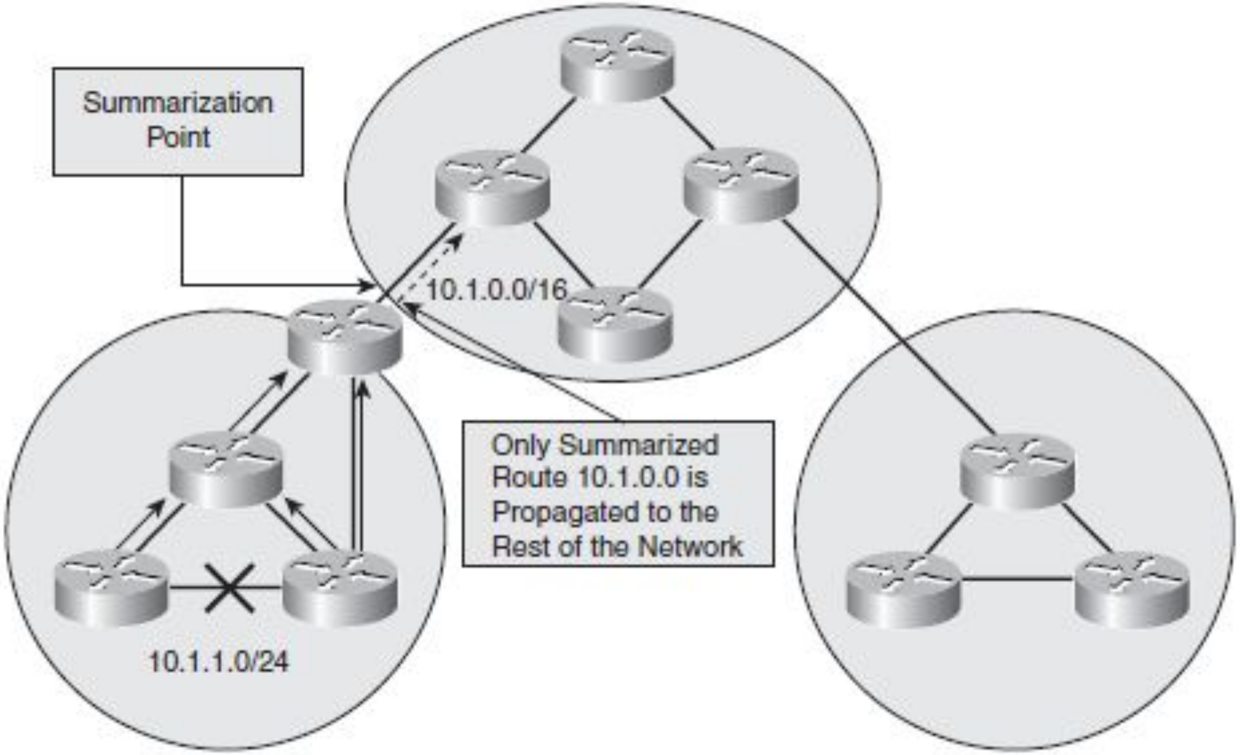
Benefits of Route Aggregation

Implementing route aggregation on border routers between contiguously addressed areas controls routing table size. The following figure shows an example of implementing route summarization (aggregation) on the area borders in a sample network. If a link within an area fails, routing updates are not propagated to the rest of the network, because only the summarized route is sent to the rest of the network, and it has not changed; the route information about the failed link stays within the area. This reduces bandwidth consumption related to routing overhead and relieves routers from unnecessary routing table recalculation.

Efficient aggregation of routing advertisements narrows the scope of routing update propagation and significantly decreases the cumulative frequency of routing updates.

Benefits of Route Aggregation

- *A Hierarchical IP Addressing Plan Results in Reduced Routing Traffic*


Summarization Point
10.1.0.0/16
Only Summarized Route 10.1.0.0 is Propagated to the Rest of the Network
10.1.1.0/24

Routing Protocol Considerations

To use VLSM, the routing protocol in use must be classless.

Classful routing protocols permit only FLSM.

Classful Routing Protocols

The following rules apply when classful routing protocols are used:

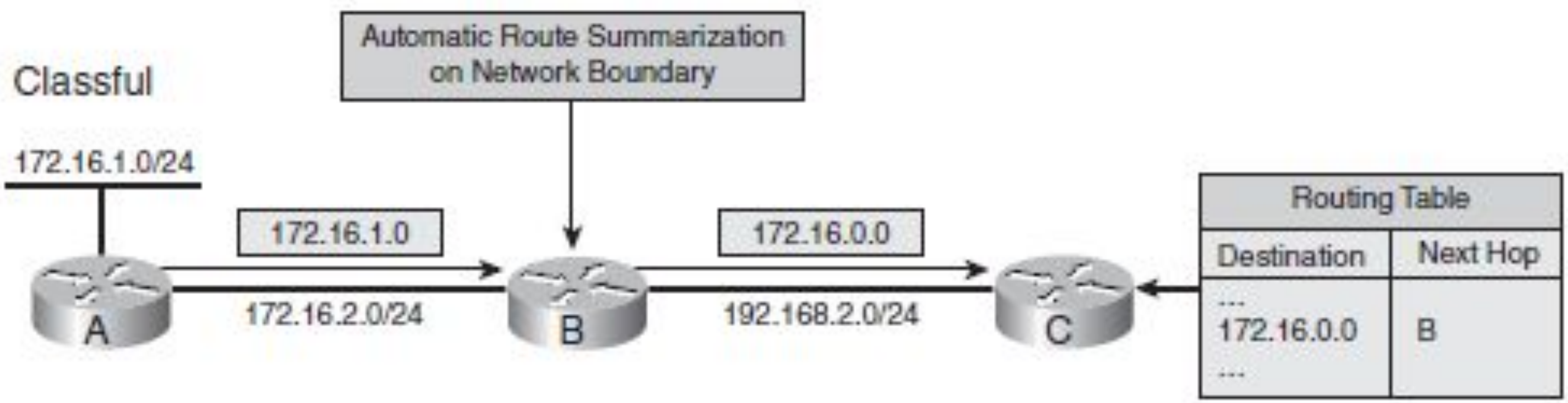
■ The routing updates do not include subnet masks.

■ When a routing update is received and the routing information is about one of the following:

— Routes within the same major network as configured on the receiving interface, the subnet mask configured on the receiving interface is assumed to apply to the received routes also. Therefore, the mask must be the same for all subnets of a major network. In other words, subnetting must be done with FLSM.

— Routes in a different major network than configured on the receiving interface, the default major network mask is assumed to apply to the received routes. Therefore, automatic route summarization is performed across major network (Class A, B, or C) boundaries, and subnetted networks must be contiguous.

Classful Routing Protocols



Routing Table

Destination	Next Hop
...	
172.16.0.0	B
...	

Classful Routing Protocols

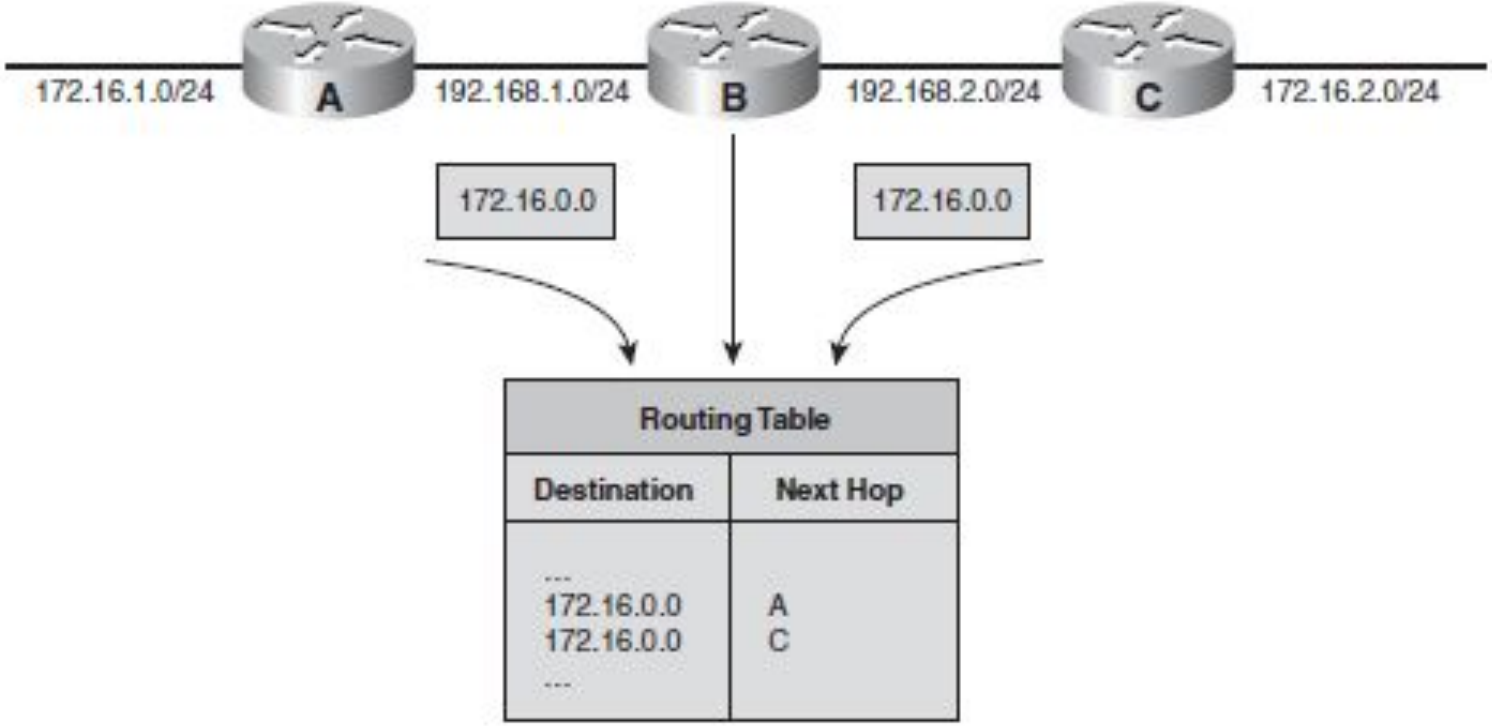
The following figure illustrates a sample network with a discontiguous 172.16.0.0 network that runs a classful routing protocol. Routers A and C automatically summarize across the major network boundary, so both send routing information about 172.16.0.0 rather than the individual subnets (172.16.1.0/24 and 172.16.2.0/24).

Consequently, Router B receives two entries for the major network 172.16.0.0, and it puts both entries into its routing table. Router B therefore might make incorrect routing decisions.

Because of these constraints, classful routing is not often used in modern networks. Routing Information Protocol (RIP) version 1 (RIPv1) is an example of a classful routing protocol.

Classful Routing Protocols

Classful Routing Protocols Do Not Send the Subnet Mask in the Routing Update



Routing Table

Destination	Next Hop
...	
172.16.0.0	A
172.16.0.0	C
...	

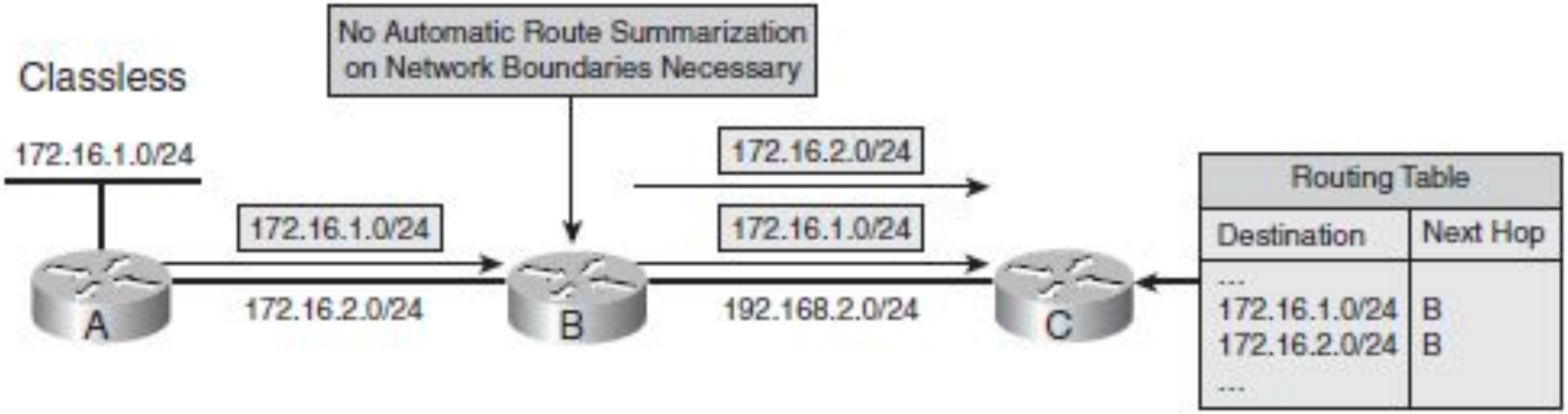
Classless Routing Protocols

The following rules apply when classless routing protocols are used:

- The routing updates include subnet masks.
- VLSM is supported.
- Automatic route summarization at the major network boundary is not required, and route summarization can be manually configured.
- Subnetted networks can be discontiguous.

Consequently, all modern networks should use classless routing. Examples of classless routing protocols include RIP version 2 (RIPv2), Enhanced Interior Gateway Routing Protocol (EIGRP), and Border Gateway Protocol (BGP).

Classless Routing Protocols



Routing Table

Destination	Next Hop
...	
172.16.1.0/24	B
172.16.2.0/24	B
...	

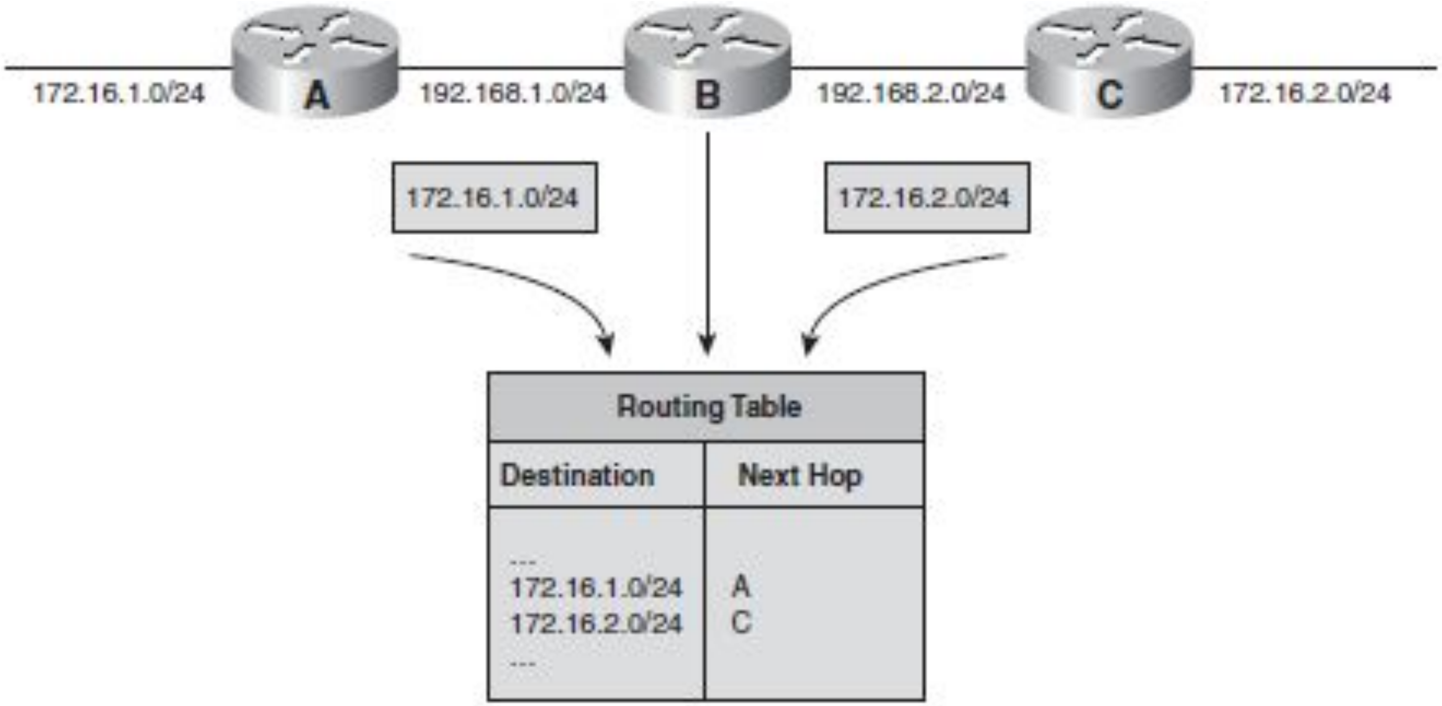
Classless Routing Protocols

The following figure illustrates how discontiguous networks are handled by a classless routing protocol.

Within this network, the classless routing protocol is running that does not automatically summarize at the network boundary. In this example, Router B learns about both subnetworks 172.16.1.0/24 and 172.16.2.0/24, one from each interface; routing is performed correctly.

Classless Routing Protocols

Classless Routing Protocols Send the Subnet Mask in the Routing Update



Routing Table	
Destination	Next Hop
...	
172.16.1.0/24	A
172.16.2.0/24	C
...	

Computer Networks Fundamentals

Lecture11:

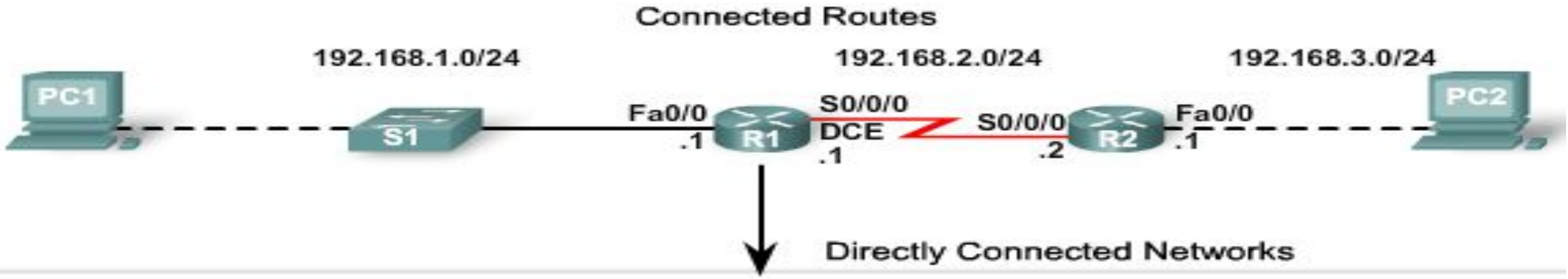
Static & Dynamic routing

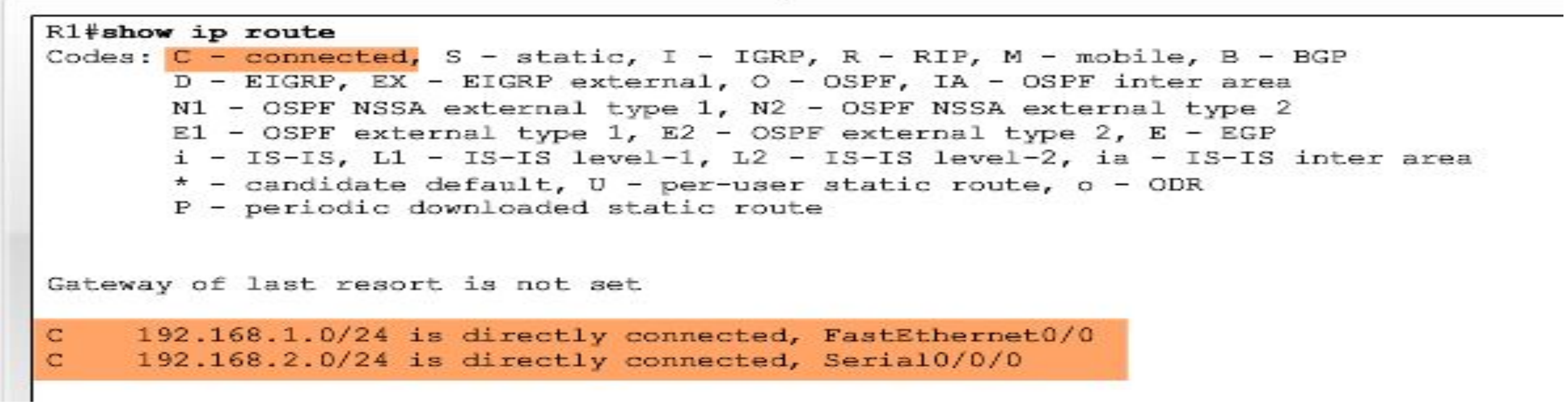
The **Router** can be configured by the :

1- static

2- dynamic

Before any static or dynamic routing is configured on a router, the router only knows about its own directly connected networks. These are the only networks that are displayed in the routing table until static or dynamic routing is configured. Directly connected networks are of prime importance for routing decisions. Static and dynamic routes cannot exist in the routing table without a router's own directly connected networks. The router cannot send packets out an interface if that interface is not enabled with an IP address and subnet mask, just as a PC cannot send IP packets out its Ethernet interface if that interface is not configured with an IP address and subnet mask.





```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route


Gateway of last resort is not set

C    192.168.1.0/24 is directly connected, FastEthernet0/0
C    192.168.2.0/24 is directly connected, Serial0/0/0
```

Introducing the Routing Table

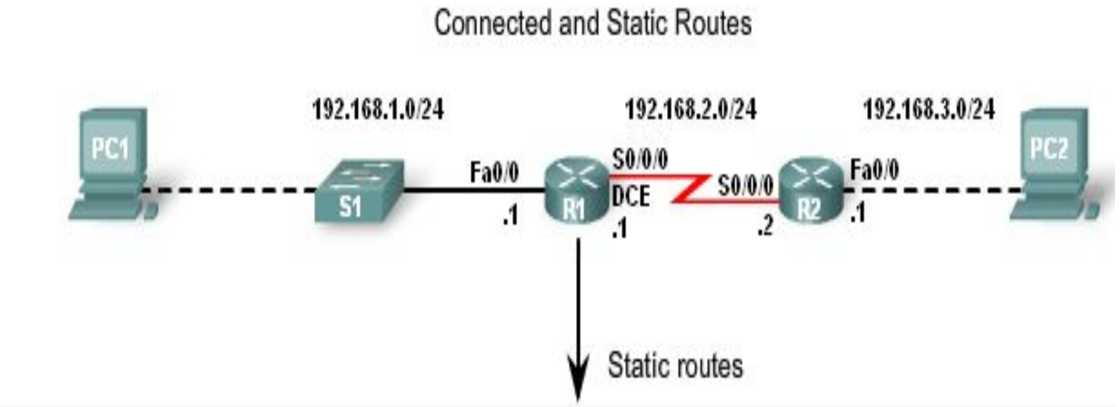
The primary function of a router is to forward a packet toward its destination network, which is the destination IP address of the packet. To do this, a router needs to search the routing information stored in its routing table.

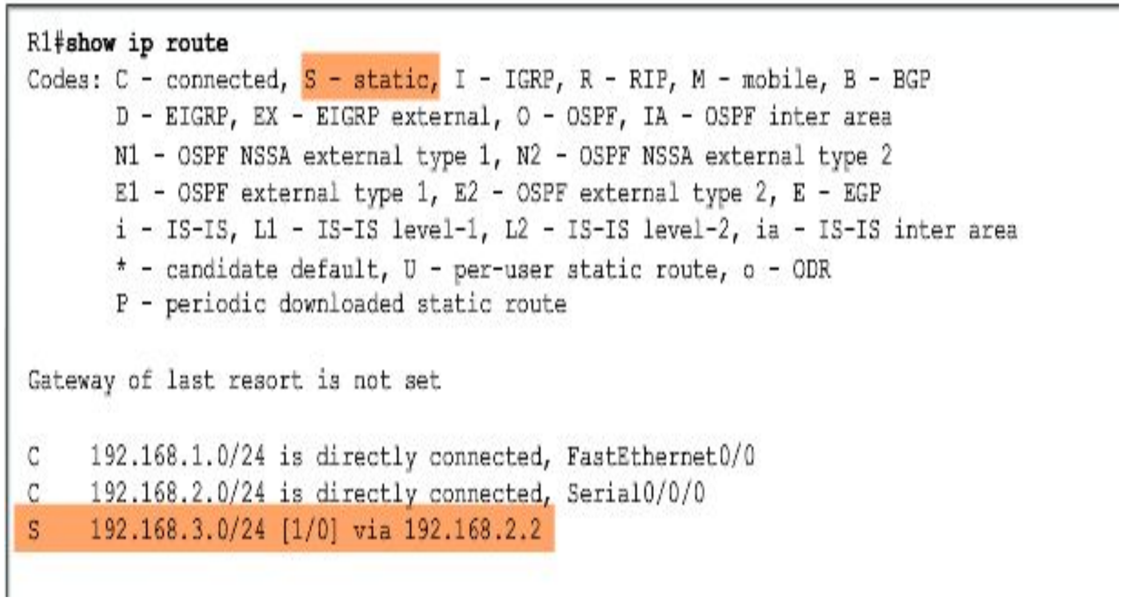
A **routing table** is a data file in RAM that is used to store route information about *directly connected* and *remote networks*. The routing table contains *network/next hop* associations. These associations tell a router that a particular destination can be optimally reached by sending the packet to a specific router that represents the "next hop" on the way to the final destination. The next hop association can also be the outgoing or exit interface to the final destination.

1- Static Routing

Remote networks are added to the routing table either by *configuring* static routes or *enabling* a dynamic routing protocol. When the IOS learns about a remote network and the interface that it will use to reach that network, it adds that route to the routing table as long as the exit interface is enabled.

A static route includes the network address and subnet mask of the remote network, along with the IP address of the next-hop router or exit interface. Static routes are denoted with the code **S** in the routing table as shown in the figure.



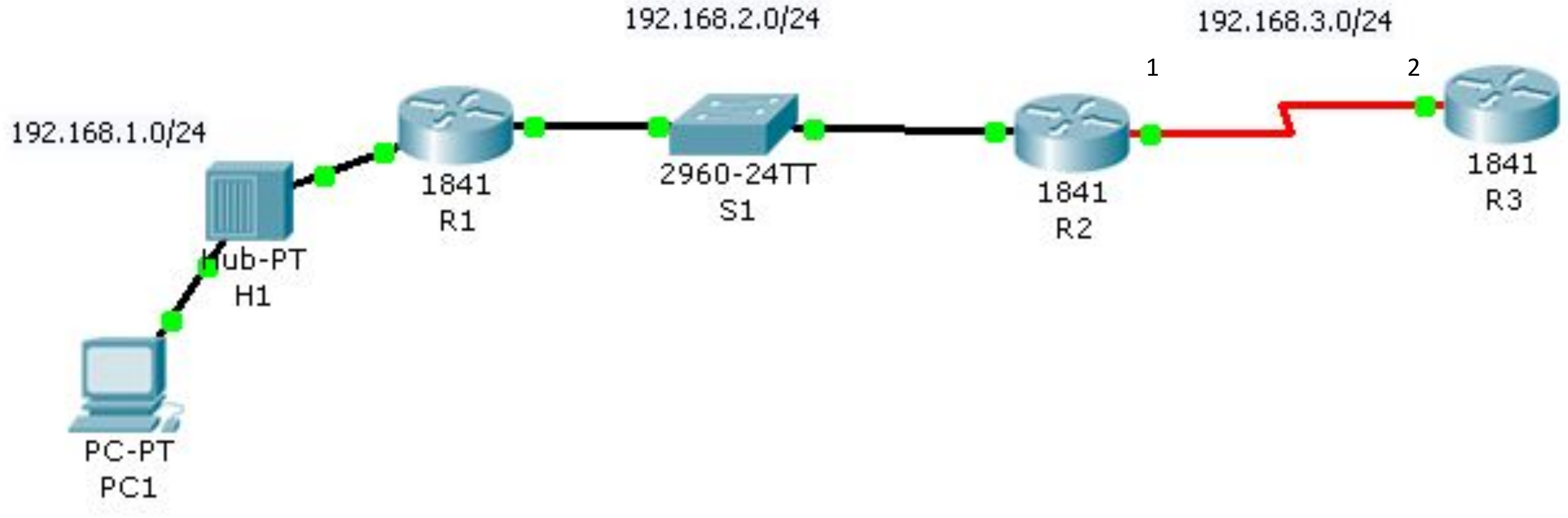


```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.1.0/24 is directly connected, FastEthernet0/0
C    192.168.2.0/24 is directly connected, Serial0/0/0
S    192.168.3.0/24 [1/0] via 192.168.2.2
```

General command of Static routing

R3(config)# IP ROUTE IP ADDRESS SUBNETMASK EXIT INTERFACE

R3(config)#**ip route 192.168.1.0 255.255.255.0 192.168.3.1**



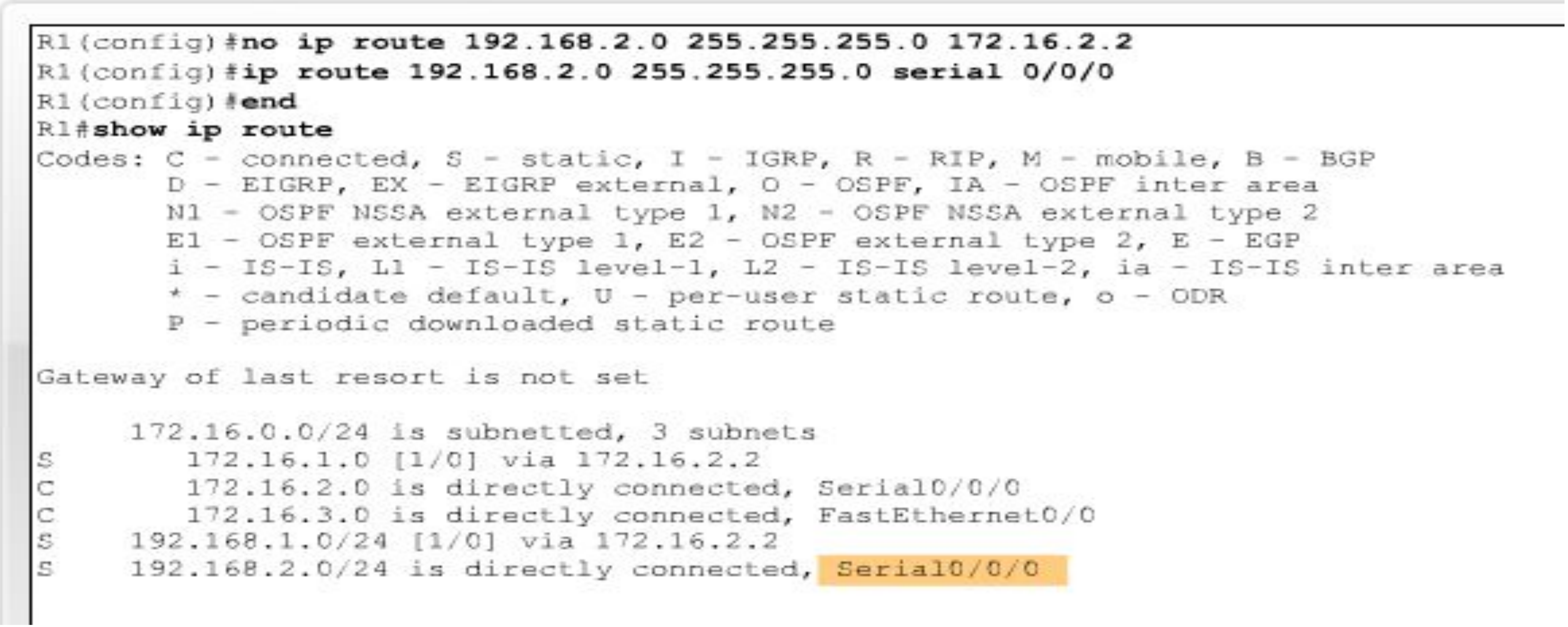
Modifying Static Routes

There are times when a previously configured static route needs to be modified:

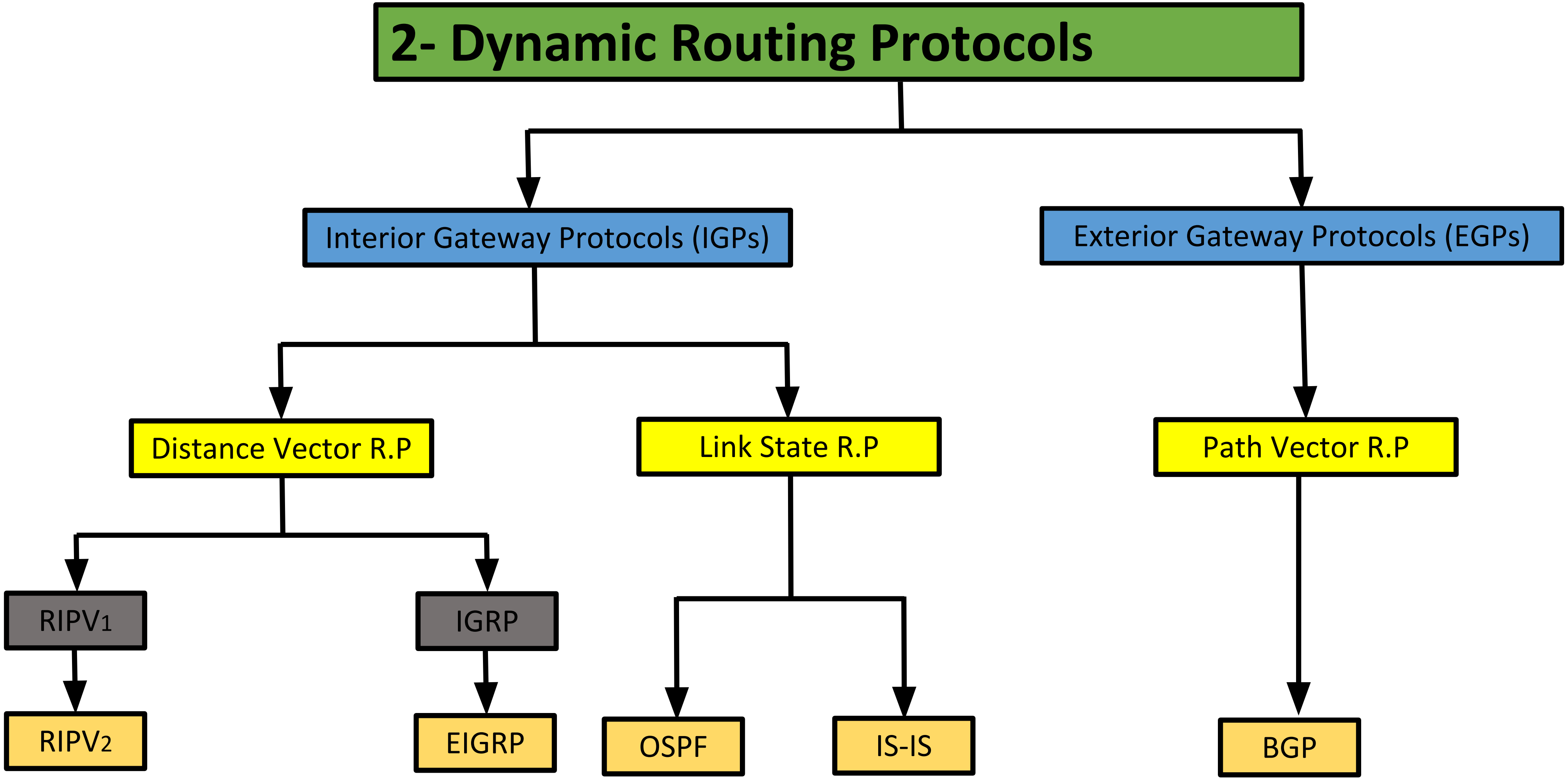
- The destination network no longer exists, and therefore the static route should be deleted.
- There is a change in the topology, and either the intermediate address or the exit interface has to be changed.
- There is no way to modify an existing static route. The static route **must be deleted** and a new one configured.
- To delete a static route, add **no** in front of the ip route command, followed by the rest of the static route to be removed.
- *It is more efficient* for the routing table lookup process to have static routes with exit interfaces - at least for serial point-to-point outbound networks.

Configuring a Static Route with an Exit Interface

Note :To delete a route we should use no ip route command and then add a new route.



```
R1(config)#no ip route 192.168.2.0 255.255.255.0 172.16.2.2
R1(config)#ip route 192.168.2.0 255.255.255.0 serial 0/0/0
R1(config)#end
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 3 subnets
S       172.16.1.0 [1/0] via 172.16.2.2
C       172.16.2.0 is directly connected, Serial0/0/0
C       172.16.3.0 is directly connected, FastEthernet0/0
S    192.168.1.0/24 [1/0] via 172.16.2.2
S    192.168.2.0/24 is directly connected, Serial0/0/0
```



2- Dynamic Routing Protocols

Interior Gateway Protocols (IGPs)

Exterior Gateway Protocols (EGPs)

Distance Vector R.P

Link State R.P

Path Vector R.P

RIPV1

RIPV2

IGRP

EIGRP

OSPF

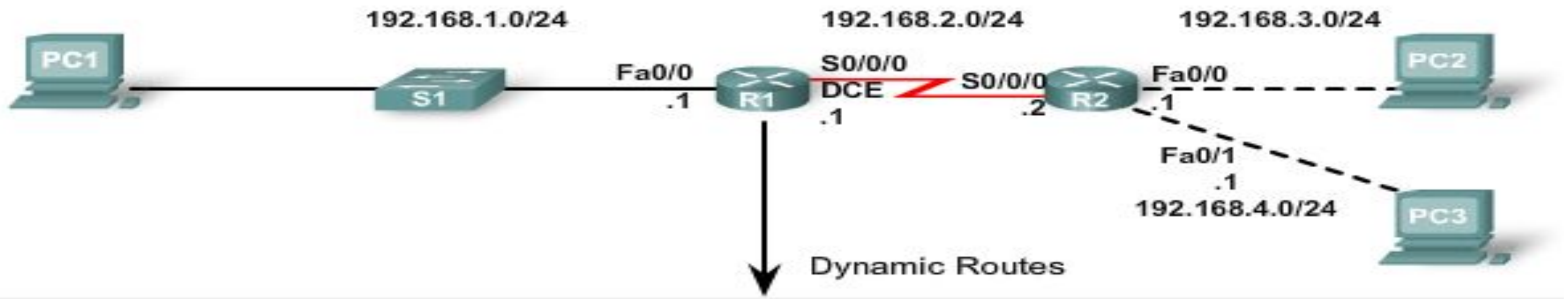
IS-IS

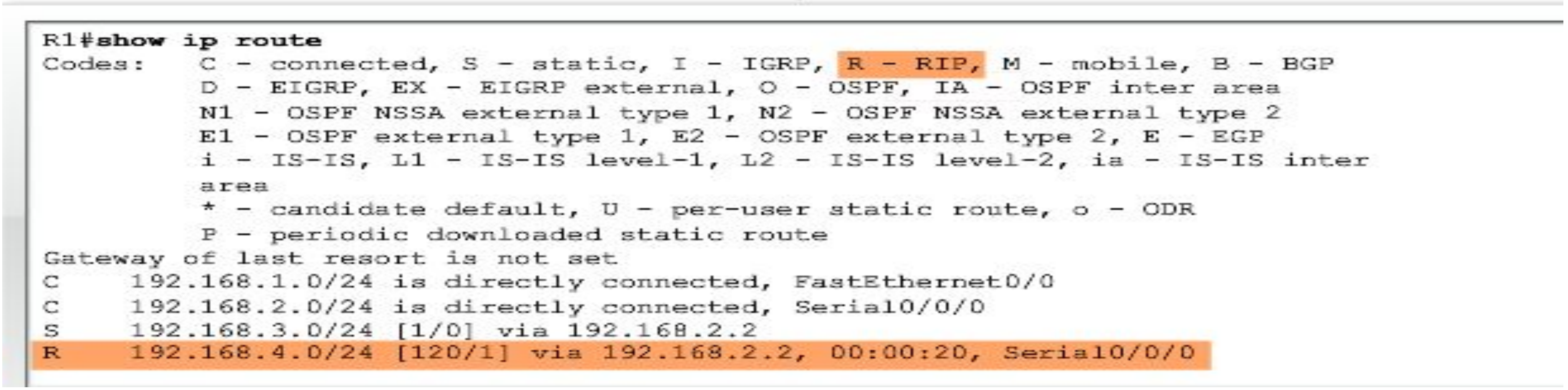
BGP

Dynamic Routing

- Remote networks can also be added to the routing table by using a dynamic routing protocol. In the figure, R1 has automatically learned about the 192.168.4.0/24 network from R2 through the dynamic routing protocol, **RIP** (Routing Information Protocol). RIP was one of the first IP routing protocols.

Connected, Static and Dynamic Routes



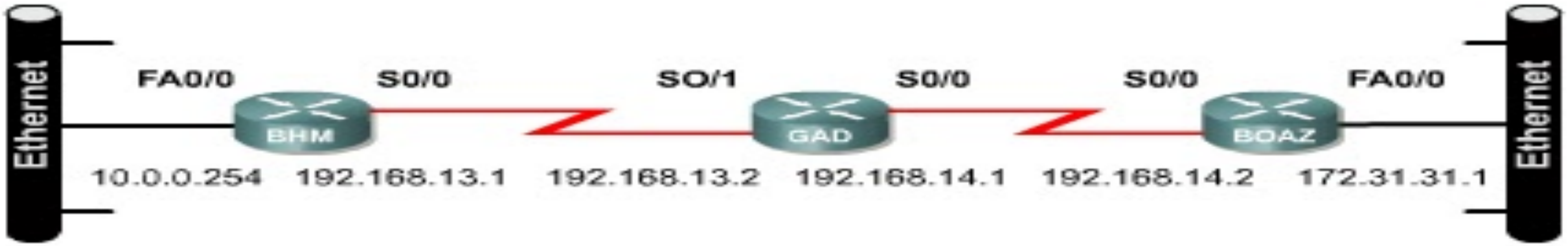


```
R1#show ip route
Codes:    C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
          D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
          N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
          E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
          i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
          area
          * - candidate default, U - per-user static route, o - ODR
          P - periodic downloaded static route
Gateway of last resort is not set
C      192.168.1.0/24 is directly connected, FastEthernet0/0
C      192.168.2.0/24 is directly connected, Serial0/0/0
S      192.168.3.0/24 [1/0] via 192.168.2.2
R      192.168.4.0/24 [120/1] via 192.168.2.2, 00:00:20, Serial0/0/0
```

Configuring RIP

FIGURE

1



```
BHM(config)#router rip
BHM(config-router)#network 10.0.0.0
BHM(config-router)#network 192.168.13.0
```

```
GAD(config)#router rip
GAD(config-router)#network 192.168.14.0
GAD(config-router)#network 192.168.13.0
```

```
BOAZ(config)#router rip
BOAZ(config-router)#network 192.168.14.0
BOAZ(config-router)#network 172.31.0.0
```

Maintaining Routing Tables

After the initial network discovery, dynamic routing protocols update and maintain the networks in their routing tables. Dynamic routing protocols not only make a best path determination to various networks, they will also determine a new best path if the initial path becomes unusable (or if the topology changes). For these reasons, dynamic routing protocols have an advantage over static routes. Routers that use dynamic routing protocols automatically share routing information with other routers and compensate for any topology changes without involving the network administrator.

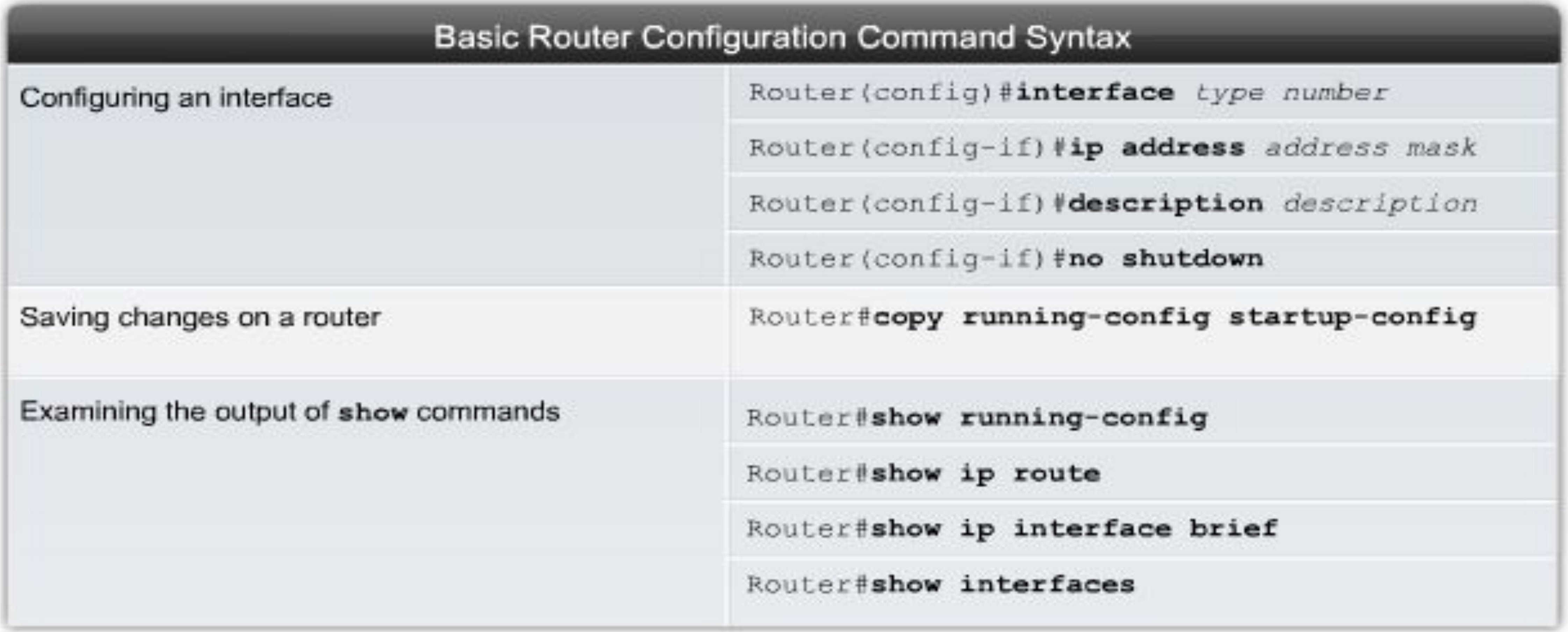
Routing Table Principles

1. Every router makes its decision alone, based on the information it has in its own routing table. After making its routing decision, router R1 forwards the packet destined for PC2 to router R2. R1 only knows about the information in its own routing table, which indicates that router R2 is the next-hop router. R1 does not know whether or not R2 actually has a route to the destination network.

2. The fact that one router has certain information in its routing table does not mean that other routers have the same information. It is the responsibility of the network administrator to make sure that all routers within their control have complete and accurate routing information so that packets can be forwarded between any two networks. This can be done using static routes, a dynamic routing protocol, or a combination of both.

3. Routing information about a path from one network to another does not provide routing information about the reverse, or return, path. Router R2 was able to forward the packet toward PC2's destination network. However, the packet from PC2 to PC1 was dropped by R2. Although R2 has information in its routing table about the destination network of PC2, we do not know if it has the information for the return path back to PC1's network.

Router Interface Configuration



Basic Router Configuration Command Syntax	
Configuring an interface	Router(config)#**interface** *type number*
	Router(config-if)#**ip address** *address mask*
	Router(config-if)#**description** *description*
	Router(config-if)#**no shutdown**
Saving changes on a router	Router#**copy running-config startup-config**
Examining the output of **show** commands	Router#**show running-config**
	Router#**show ip route**
	Router#**show ip interface brief**
	Router#**show interfaces**

The Purpose of Dynamic Routing Protocols

A routing protocol is a set of algorithms, and messages that are used to exchange routing information and populate the routing table with the best paths.

The purpose of a routing protocol includes:

- Discovery of remote networks
- Maintaining up-to-date routing information
- Choosing the best path to destination networks
- Ability to find a new best path if the current path is no longer available

The components of a routing protocol

- **Data structures** - Some routing protocols use tables and/or databases for its operations. This information is kept in RAM.
- **Algorithm** - An algorithm is a finite list of steps used in accomplishing a task. Routing protocols use algorithms for facilitating routing information and for best path determination.
- **Routing protocol messages** - Routing protocols use various types of messages to discover neighboring routers, exchange routing information, and other tasks to learn and maintain accurate information about the network.

Static Routing Advantages and Disadvantages

Static routing advantages:

- Minimal CPU processing.
- Easier for administrator to understand.
- Easy to configure.

Static routing disadvantages:

- Configuration and maintenance is time-consuming.
- Configuration is error-prone, especially in large networks.
- Administrator intervention is required to maintain changing route information.
- Does not scale well with growing networks; maintenance becomes cumbersome.
- Requires complete knowledge of the whole network for proper implementation.

Dynamic Routing Advantages and Disadvantages

Dynamic routing advantages:

- Administrator has less work maintaining the configuration when adding or deleting networks.
- Protocols automatically react to the topology changes.
- Configuration is less error-prone.
- More scalable, growing the network usually does not present a problem.

Dynamic routing disadvantages:

- Router resources are used (CPU cycles, memory and link bandwidth).
- More administrator knowledge is required for configuration, verification, and troubleshooting.

In the table, dynamic and static routing features are directly compared. From this comparison, we can list the advantages of each routing method. The advantages of one method are the disadvantages of the other.

Static versus Dynamic Routing

Static Routing	Dynamic Routing	Feature
Increases with network size	Generally independent of the network size	Configuration Complexity
No extra knowledge required	Advanced knowledge required	Requires administration knowledge
Administration intervention required	Automatically adapts to topology changes	Topology changes
Suitable for simple topologies	Suitable for simple and complex topologies	Scaling
More Security	Less Security	Security
No extra resources needed	Uses CPU, memory, link bandwidth	Resource usage
Route to destination is always the same	Route depends on the current topology	Predictability

Thanks for listening

Distance Vector Routing Protocols

Distance vector routing protocols include RIPs, IGRP, and EIGRP.

The Meaning of Distance Vector

As the name implies, distance vector means that routes are advertised as vectors of distance and direction. Distance is defined in terms of a metric such as hop count and direction is simply the next-hop router or exit interface.

A router using a distance vector routing protocol does not have the knowledge of the entire path to a destination network. Instead the router knows only:

- The direction or interface in which packets should be forwarded and
- The distance or how far it is to the destination network

(RIP)Routing Information Protocol was originally specified in RFC 1058. It has the following key characteristics:

* Hop count is used as the metric for path selection.

* If the hop count for a network is greater than 15, RIP cannot supply a route to that network.

* Routing updates are broadcast or multicast every 30 seconds, by default.

IGRP

- Interior Gateway Routing Protocol (IGRP) is a proprietary protocol developed by Cisco. IGRP has the following key design characteristics:
- Bandwidth, delay, load and reliability are used to create a composite metric.
- Routing updates are broadcast every 90 seconds, by default.
- IGRP is the predecessor of EIGRP and is now obsolete.

EIGRP

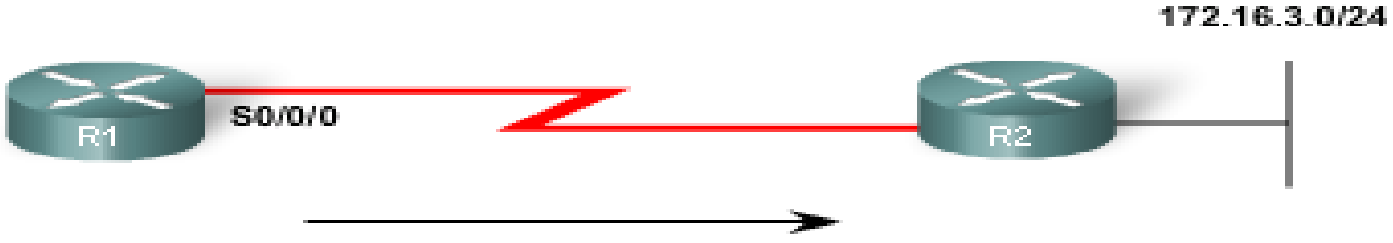
Enhanced IGRP (EIGRP) is a Cisco proprietary distance vector routing protocol. EIGRP has these key characteristics:

- It can perform unequal cost load balancing.
- It uses Diffusing Update Algorithm (DUAL) to calculate the shortest path.
- There are no periodic updates as with RIP and IGRP. Routing updates are sent only when there is a change in the topology.

For example, in the figure, R1 knows that the distance to reach network 172.16.3.0/24 is 1 hop and that the direction is out the interface S0/0/0 toward R2.

The Meaning of Distance Vector

Distance = How Far



Vector = Direction

For R1, 172.16.3.0/24 is one hop away (distance).
It can be reached through R2 (vector).

Operation of Distance Vector Routing Protocols

Some distance vector routing protocols, periodically broadcast the entire routing table to each of its neighbors. This method is inefficient because the updates not only consume bandwidth but also consume router CPU resources to process the updates.

Periodic Updates are sent at regular intervals (30 seconds for RIP and 90 seconds for IGRP). Even if the topology has not changed in several days, periodic updates continue to be sent to all neighbors.

Neighbors are routers that share a link and are configured to use the same routing protocol. The router is only aware of the network addresses of its own interfaces and the remote network addresses it can reach through its neighbors. It has no broader knowledge of the network topology. **Routers using distance vector routing are not aware of the network topology.**

Broadcast Updates are sent to 255.255.255.255. Neighboring routers that are configured with the same routing protocol will process the updates. All other devices will also process the update up to Layer 3 before discarding it. Some distance vector routing protocols use multicast addresses instead of broadcast addresses.

Entire Routing Table Updates are sent, with some exceptions to be discussed later, periodically to all neighbors. Neighbors receiving these updates must process the entire update to find related information and discard the rest. Some distance vector routing protocols like EIGRP do not send periodic routing table updates.

Routing Protocol Algorithms

At the core of the distance vector protocol is the algorithm. The algorithm is used to calculate the best paths and then send that information to the neighbors.

An algorithm is a procedure for accomplishing a certain task, starting at a given initial state and terminating in a defined end state.

Different routing protocols use different algorithms to:

install routes in the routing table,

send updates to neighbors, and

make path determination decisions.

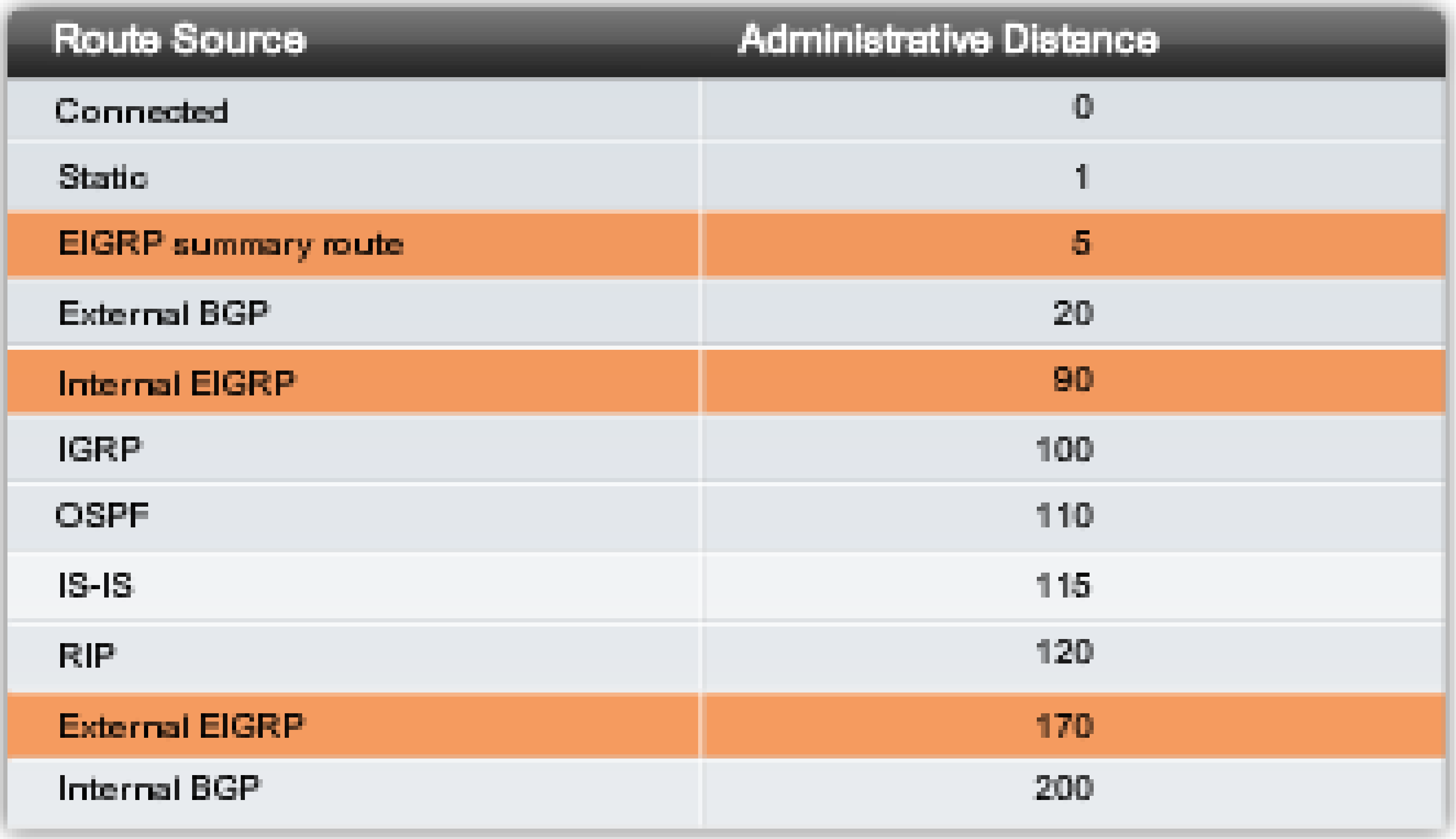
The algorithm used for the routing protocols defines the following processes:

- Mechanism for sending and receiving routing information.
- Mechanism for calculating the best paths and installing routes in the routing table.
- Mechanism for detecting and reacting to topology changes.

Routing protocols can be compared based on the following characteristics:

- **Time to Convergence** - Time to convergence defines how quickly the routers in the network topology share routing information and reach a state of consistent knowledge. The faster the convergence, the more preferable the protocol. *Routing loops* can occur when inconsistent routing tables are not updated due to slow convergence in a changing network.
- **Scalability** - Scalability defines how large a network can become based on the routing protocol that is deployed. The larger the network is, the more scalable the routing protocol needs to be.
- **Classless (Use of VLSM) or Classful** - *Classless* routing protocols *include the subnet mask in the updates*. This feature supports the use of Variable Length Subnet Masking (VLSM) and better route summarization. *Classful* routing protocols *do not include* the subnet mask and cannot support VLSM.
- **Resource Usage** - Resource usage includes the requirements of a routing protocol such as memory space, CPU utilization, and link bandwidth utilization. Higher resource requirements require more powerful hardware to support the routing protocol operation in addition to the packet forwarding processes.
- **Implementation and Maintenance** - Implementation and maintenance describes the level of knowledge that is required for a network administrator to implement and maintain the network based on the routing protocol deployed.

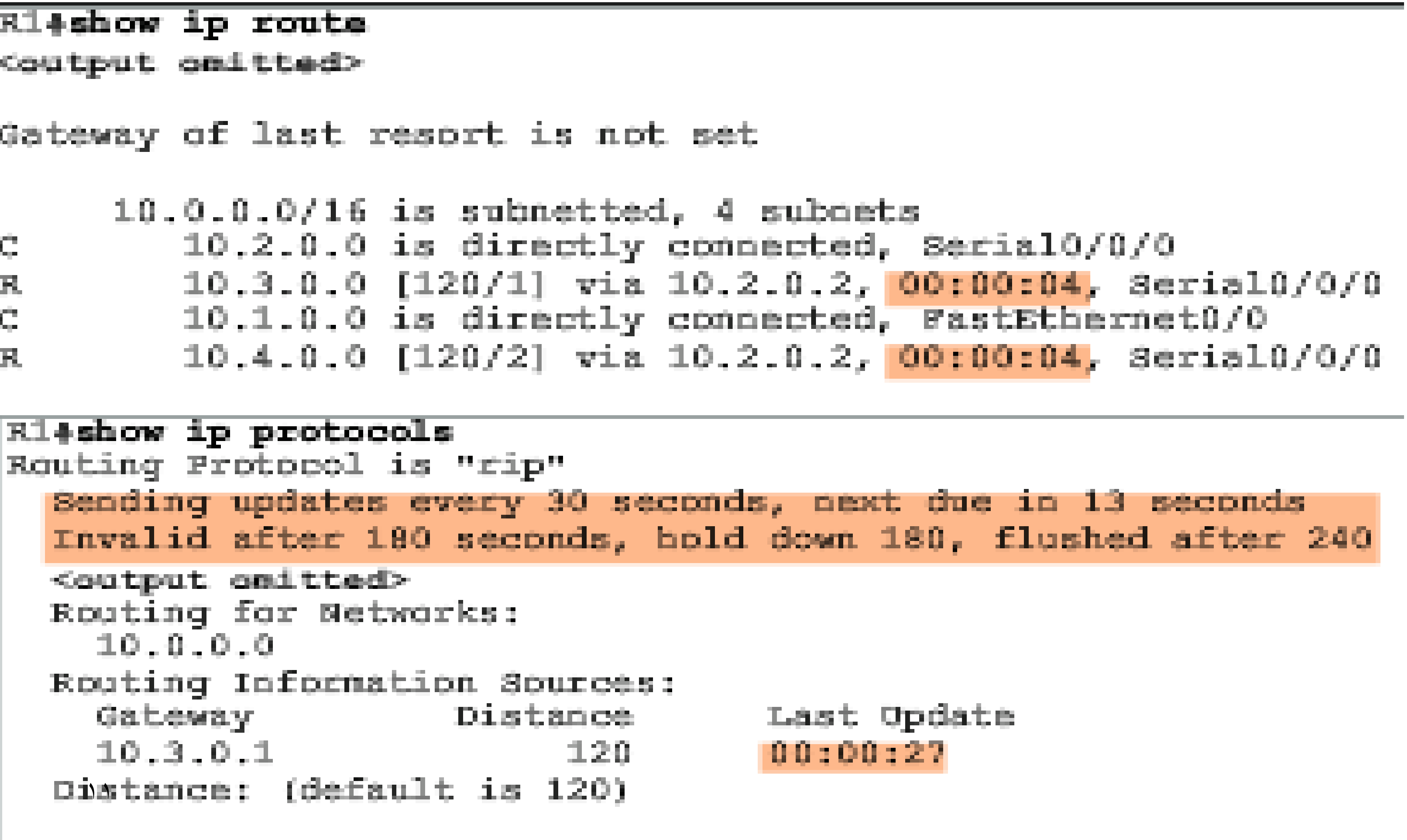
Feature	Distance Vector				Link state	
	RIPv1	RIPv2	IGRP	EIGRP	OSPF	IS-IS
• Speed of Convergence	Slow	Slow	Slow	Fast	Fast	Fast
• Scalability (size of Network)	Small	Small	Small	Large	Large	Large
• Use of VLSM	No	Yes	No	Yes	Yes	Yes
• Resource Usage	Low	Low	Low	Medium	High	High
• Implementation and Maintenance	Simple	Simple	Simple	Complex	Complex	Complex



Route Source	Administrative Distance
Connected	0
Static	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
External EIGRP	170
Internal BGP	200

Advantages and disadvantages of distance vector routing protocols

Advantages	Disadvantages
Simple implementation and maintenance: The level of knowledge required to deploy and later maintain a network with distance vector protocol in not high.	**Slow convergence:** The use of periodic updates can cause slower convergence. Even if some advanced techniques are used, like triggered updates, which are discussed later, the overall convergence is still slower compared to a link state routing protocol.
Low resource requirements: distance vector protocols typically do not need amounts of memory to store the information. Nor do they require a powerfull CPU. Depending on the network size and the IP addressing implemented, they also typically do not require a high level of link bandwidth to send routing updates. However , this can ecome an issue if you deploy a distance vetor protocol in a large network.	**Limited Scalability :** Slow Convergence may limit the size of the network because larger networks require more time to propagate routing information.
	Routing loops: Routing loops can occur when inconsistent routing tables are not update due to slow convergence in a changing network.



```
R1#show ip route
<output omitted>

Gateway of last resort is not set

     10.0.0.0/16 is subnetted, 4 subnets
C       10.2.0.0 is directly connected, Serial0/0/0
R       10.3.0.0 [120/1] via 10.2.0.2, 00:00:04, Serial0/0/0
C       10.1.0.0 is directly connected, FastEthernet0/0
R       10.4.0.0 [120/2] via 10.2.0.2, 00:00:04, Serial0/0/0
```

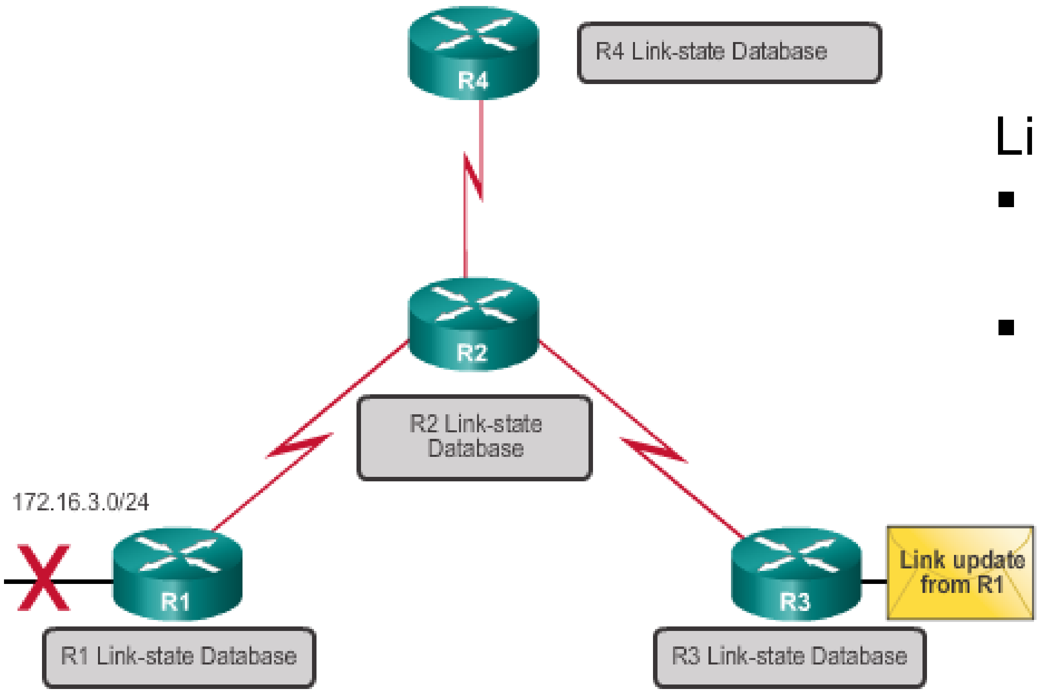
```
R1#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 13 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  <output omitted>
  Routing for Networks:
    10.0.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.3.0.1             120      00:00:27
  Distance: (default is 120)
```

Link State Routing Protocols

Types of Routing Protocols

Link-State Routing Protocols

Link-State Protocol Operation



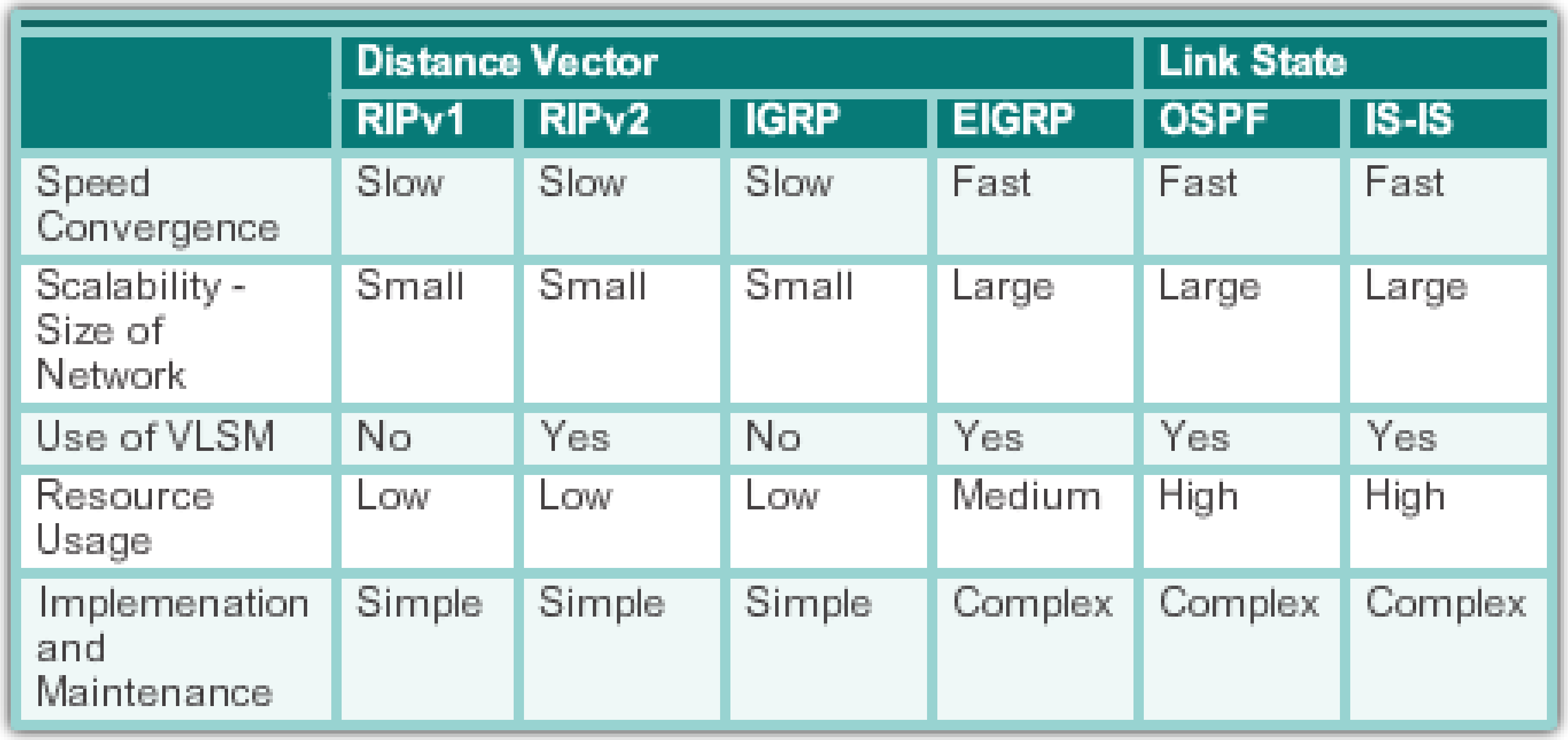
Link-state protocols forward updates when the state of a link changes.

Link-state IPv4 IGPs:

- **OSPF** - Popular standards based routing protocol
- **IS-IS** - Popular in provider networks.

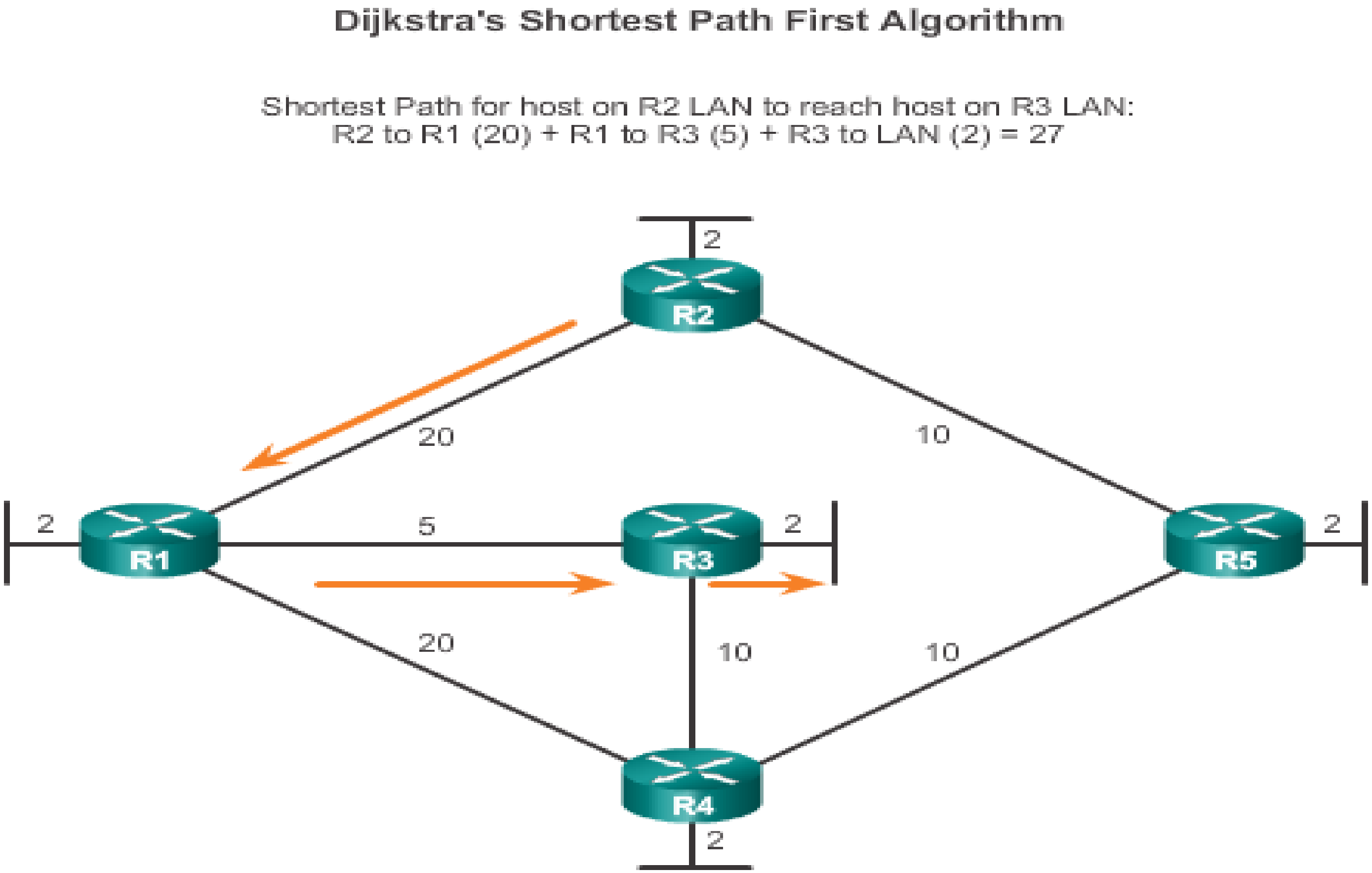
Types of Routing Protocols

Routing Protocol Characteristics



	Distance Vector				Link State	
	RIPv1	RIPv2	IGRP	EIGRP	OSPF	IS-IS
Speed Convergence	Slow	Slow	Slow	Fast	Fast	Fast
Scalability - Size of Network	Small	Small	Small	Large	Large	Large
Use of VLSM	No	Yes	No	Yes	Yes	Yes
Resource Usage	Low	Low	Low	Medium	High	High
Implemenation and Maintenance	Simple	Simple	Simple	Complex	Complex	Complex

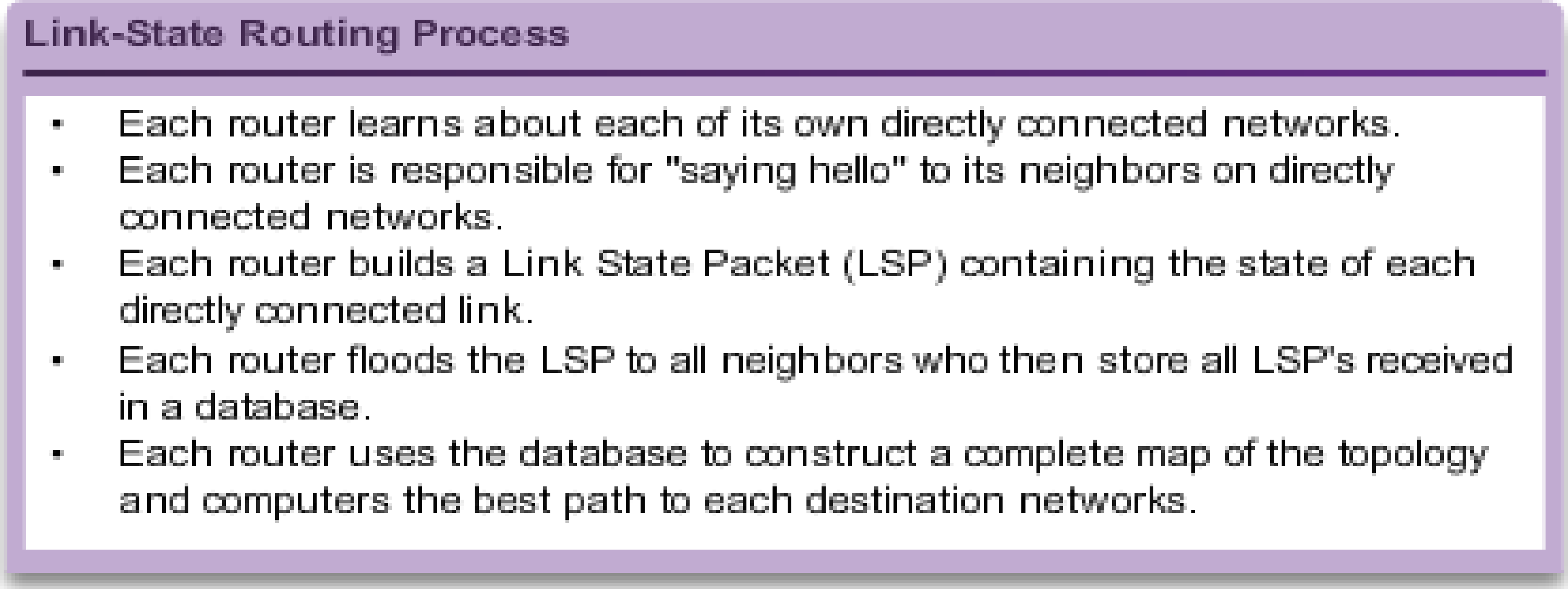
Link-State Routing Protocol Operation (SPF)
Dijkstra’s Algorithm



Dijkstra's Shortest Path First Algorithm
Shortest Path for host on R2 LAN to reach host on R3 LAN:
R2 to R1 (20) + R1 to R3 (5) + R3 to LAN (2) = 27
2
R2
20
10
2
R1
5
R3
2
R5
2
20
10
10
R4
2

Link-State Updates

Link-State Routing Process



Link-State Routing Process

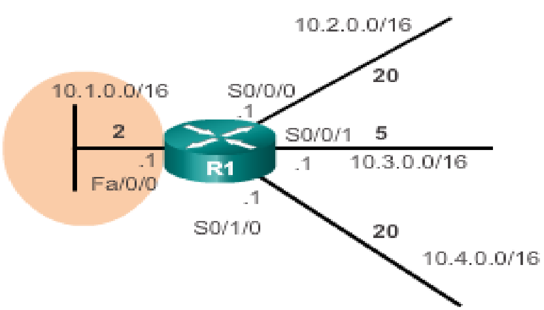
- Each router learns about each of its own directly connected networks.
- Each router is responsible for "saying hello" to its neighbors on directly connected networks.
- Each router builds a Link State Packet (LSP) containing the state of each directly connected link.
- Each router floods the LSP to all neighbors who then store all LSP's received in a database.
- Each router uses the database to construct a complete map of the topology and computers the best path to each destination networks.

Link-State Updates

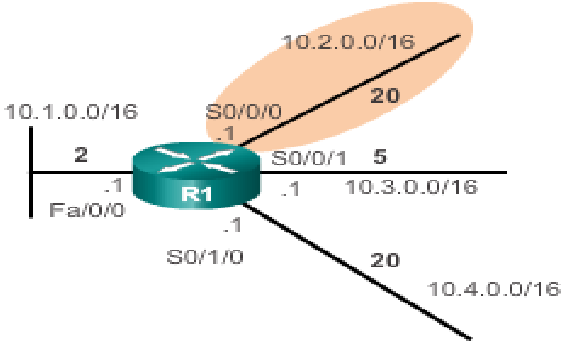
Link and Link-State

The first step in the link-state routing process is that each router learns about its own links and its own directly connected networks.

Link-State of Interface Fa0/0



Link-State of Interface S0/0/0



Link 1

- Network: **10.1.0.0/16**
- IP address: **10.1.0.1**
- Type of network: **Ethernet**
- Cost of that link: **2**
- Neighbors: **None**

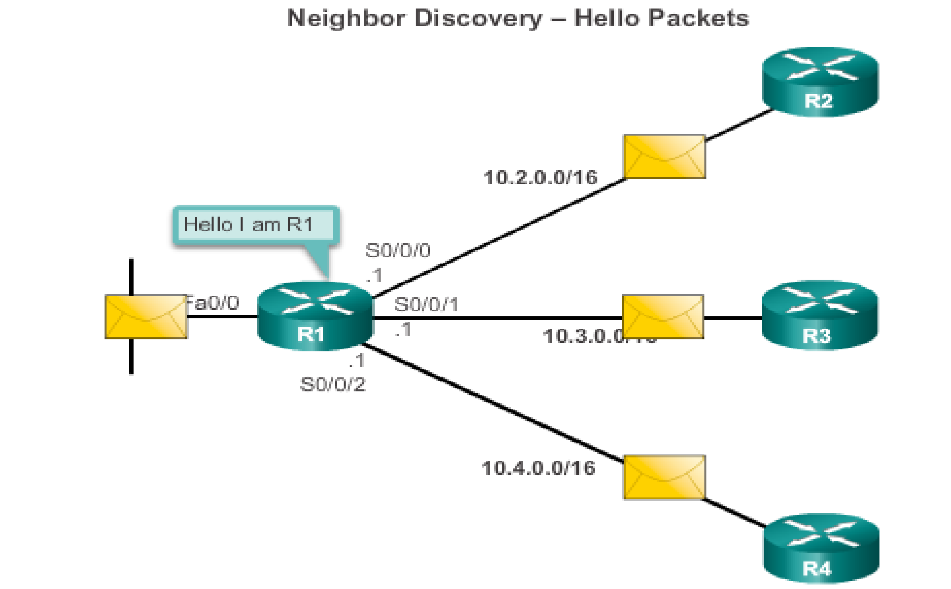
Link 2

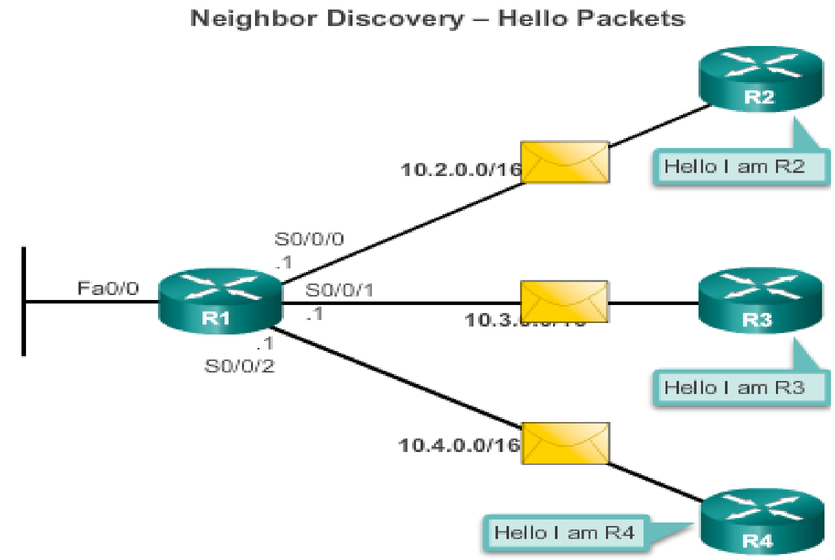
- Network: **10.2.0.0/16**
- IP address: **10.2.0.1**
- Type of network: **Serial**
- Cost of that link: **20**
- Neighbors: **R2**

Link-State Updates

Say Hello

The second step in the link-state routing process is that each router is responsible for meeting its neighbors on directly connected networks.


Neighbor Discovery – Hello Packets

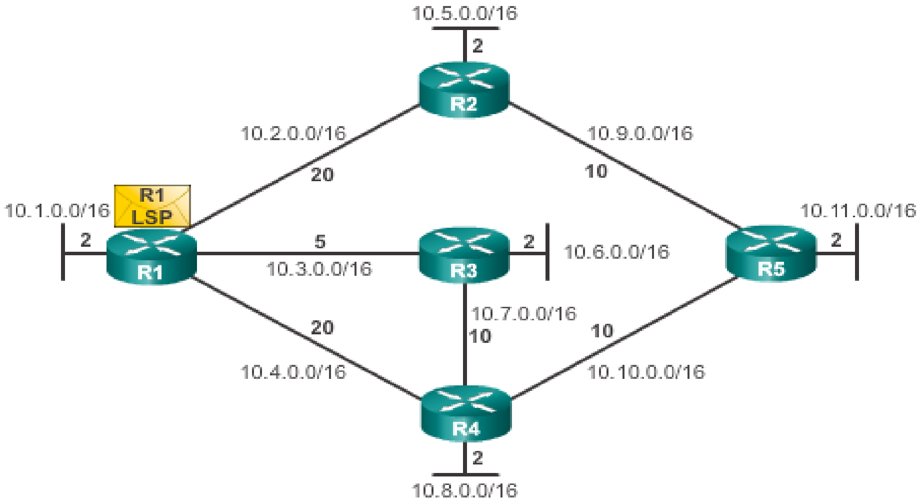

Neighbor Discovery – Hello Packets

Link-State Updates

Say Hello

The third step in the link-state routing process is that each router builds a link-state packet (LSP) containing the state of each directly connected link.

Building the LSP

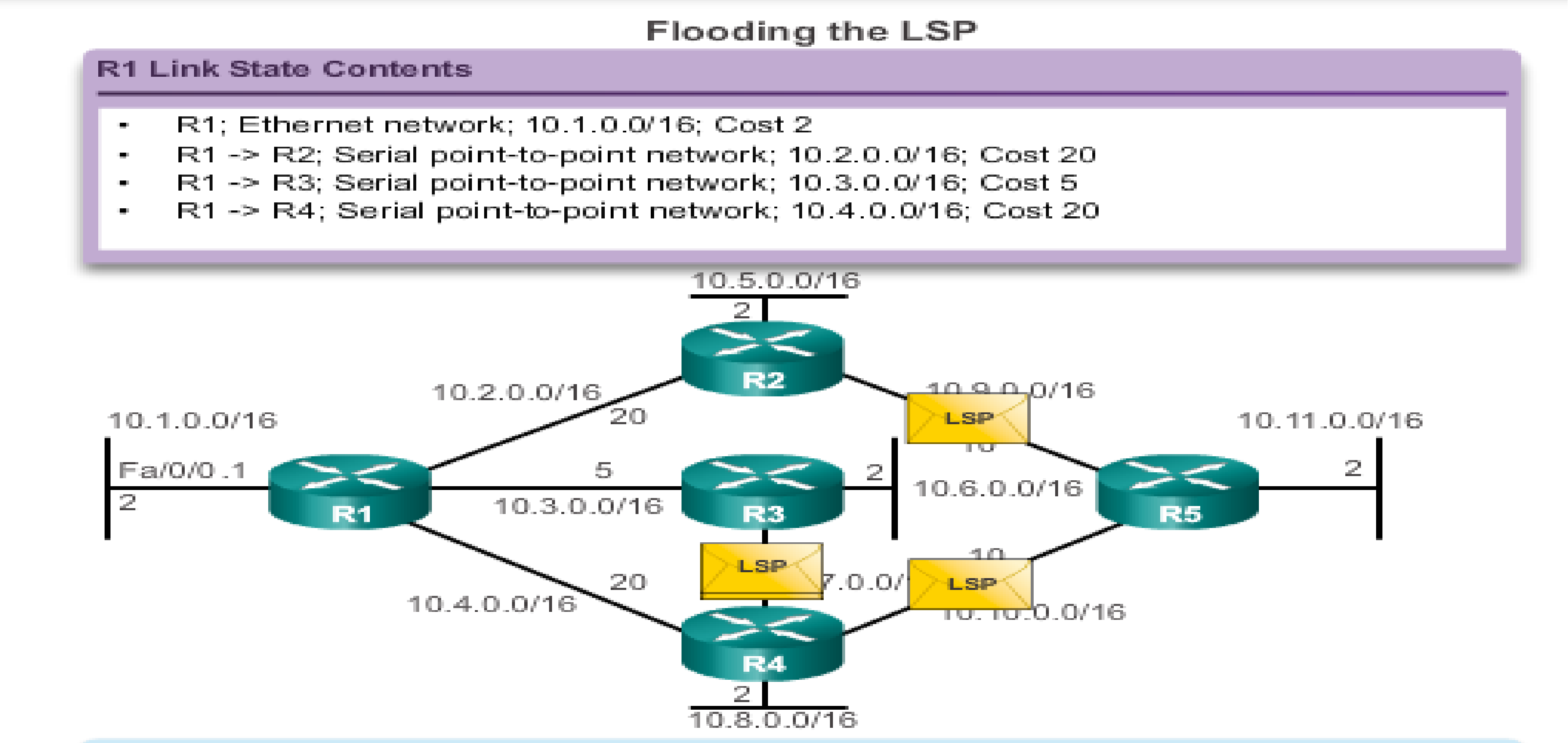


1. R1; Ethernet network 10.1.0.0/16; Cost 2
2. R1 -> R2; Serial point-to-point network; 10.2.0.0/16; Cost 20
3. R1 -> R3; Serial point-to-point network; 10.7.0.0/16; Cost 5
4. R1 -> R4; Serial point-to-point network; 10.4.0.0/16; Cost 20

Link-State Updates

Flooding the LSP

The fourth step in the link-state routing process is that each router floods the LSP to all neighbors, who then store all LSPs received in a database.



Flooding the LSP

R1 Link State Contents

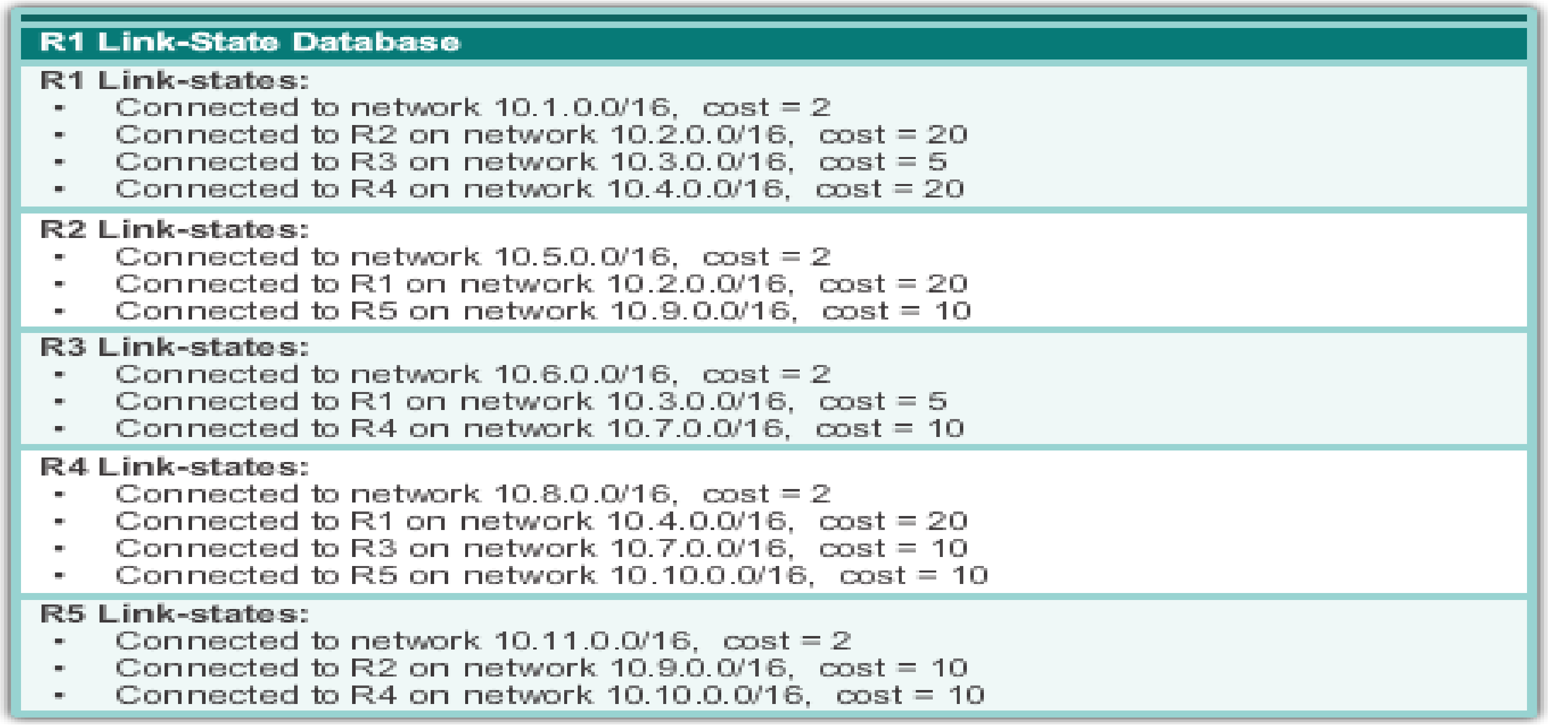
- R1; Ethernet network; 10.1.0.0/16; Cost 2
- R1 -> R2; Serial point-to-point network; 10.2.0.0/16; Cost 20
- R1 -> R3; Serial point-to-point network; 10.3.0.0/16; Cost 5
- R1 -> R4; Serial point-to-point network; 10.4.0.0/16; Cost 20

Link-State Updates

Building the Link-State Database

The final step in the link-state routing process is that each router uses the database to construct a complete map of the topology and computes the best path to each destination network.

Contents of the Link-State Database



R1 Link-State Database

R1 Link-states:

- Connected to network 10.1.0.0/16, cost = 2
- Connected to R2 on network 10.2.0.0/16, cost = 20
- Connected to R3 on network 10.3.0.0/16, cost = 5
- Connected to R4 on network 10.4.0.0/16, cost = 20

R2 Link-states:

- Connected to network 10.5.0.0/16, cost = 2
- Connected to R1 on network 10.2.0.0/16, cost = 20
- Connected to R5 on network 10.9.0.0/16, cost = 10

R3 Link-states:

- Connected to network 10.6.0.0/16, cost = 2
- Connected to R1 on network 10.3.0.0/16, cost = 5
- Connected to R4 on network 10.7.0.0/16, cost = 10

R4 Link-states:

- Connected to network 10.8.0.0/16, cost = 2
- Connected to R1 on network 10.4.0.0/16, cost = 20
- Connected to R3 on network 10.7.0.0/16, cost = 10
- Connected to R5 on network 10.10.0.0/16, cost = 10

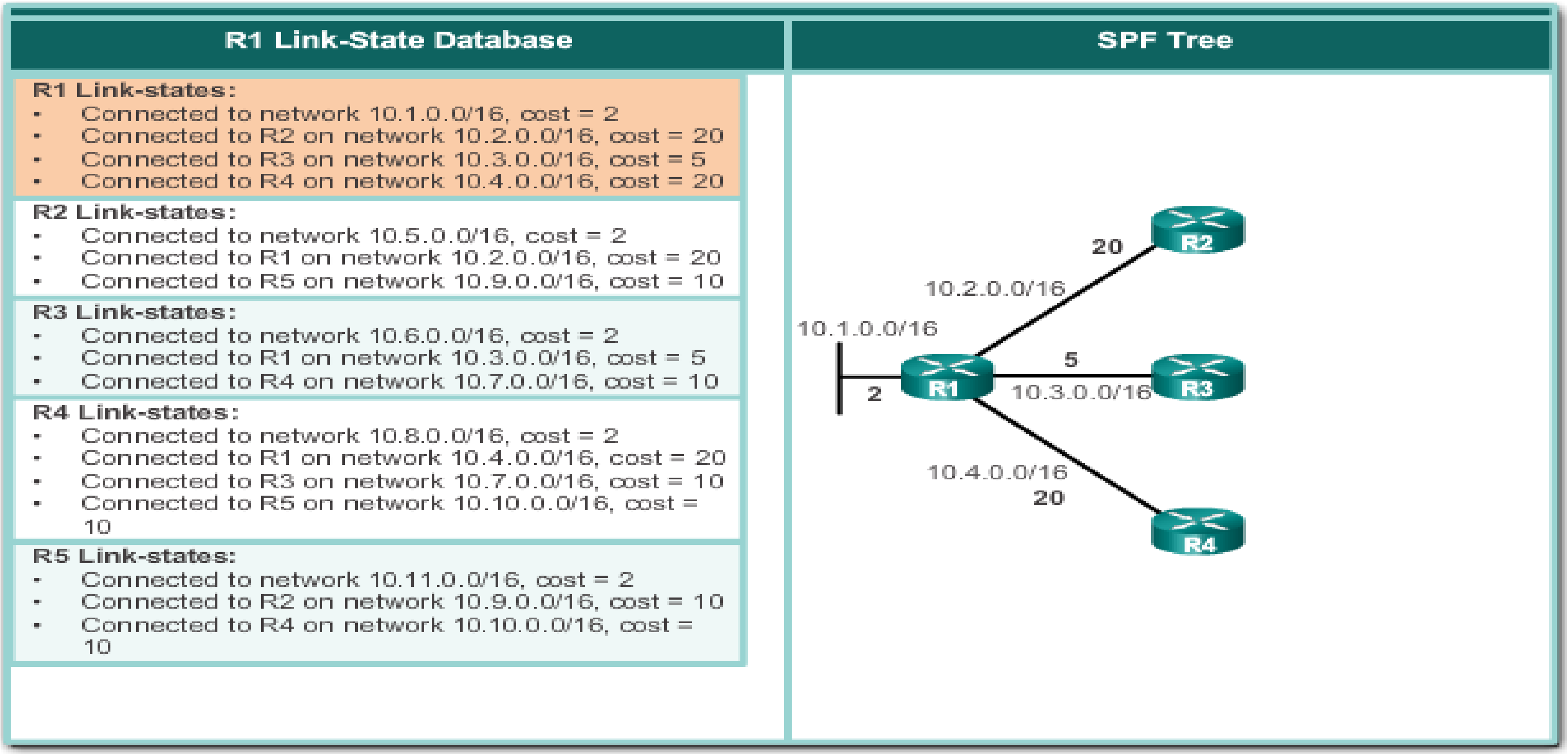
R5 Link-states:

- Connected to network 10.11.0.0/16, cost = 2
- Connected to R2 on network 10.9.0.0/16, cost = 10
- Connected to R4 on network 10.10.0.0/16, cost = 10

Link-State Updates

Building the SPF Tree

Identify the Directly Connected Networks



R1 Link-State Database

R1 Link-states:
- Connected to network 10.1.0.0/16, cost = 2
- Connected to R2 on network 10.2.0.0/16, cost = 20
- Connected to R3 on network 10.3.0.0/16, cost = 5
- Connected to R4 on network 10.4.0.0/16, cost = 20

R2 Link-states:
- Connected to network 10.5.0.0/16, cost = 2
- Connected to R1 on network 10.2.0.0/16, cost = 20
- Connected to R5 on network 10.9.0.0/16, cost = 10

R3 Link-states:
- Connected to network 10.6.0.0/16, cost = 2
- Connected to R1 on network 10.3.0.0/16, cost = 5
- Connected to R4 on network 10.7.0.0/16, cost = 10

R4 Link-states:
- Connected to network 10.8.0.0/16, cost = 2
- Connected to R1 on network 10.4.0.0/16, cost = 20
- Connected to R3 on network 10.7.0.0/16, cost = 10
- Connected to R5 on network 10.10.0.0/16, cost = 10

R5 Link-states:
- Connected to network 10.11.0.0/16, cost = 2
- Connected to R2 on network 10.9.0.0/16, cost = 10
- Connected to R4 on network 10.10.0.0/16, cost = 10

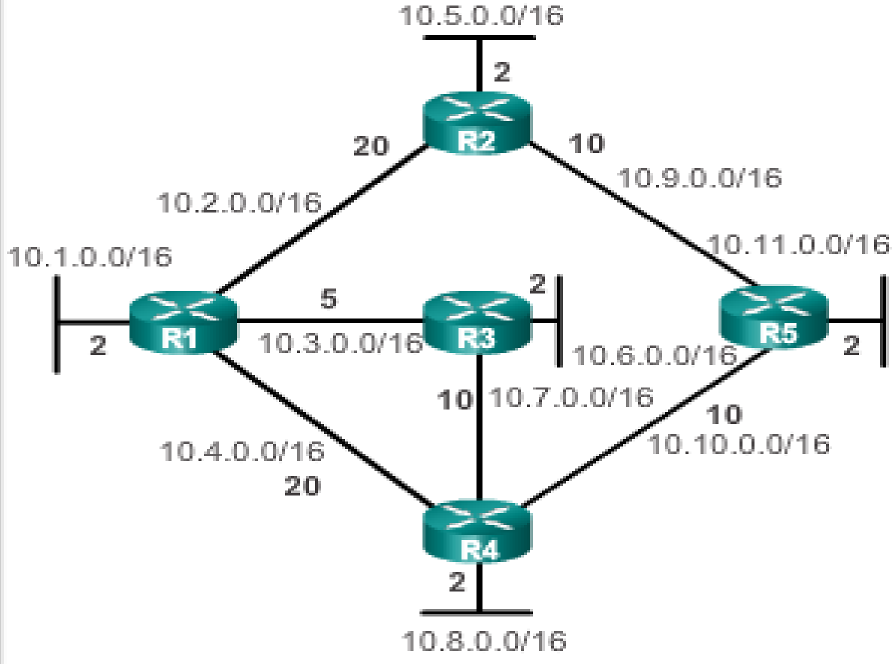
SPF Tree

Link-State Updates

Building the SPF Tree

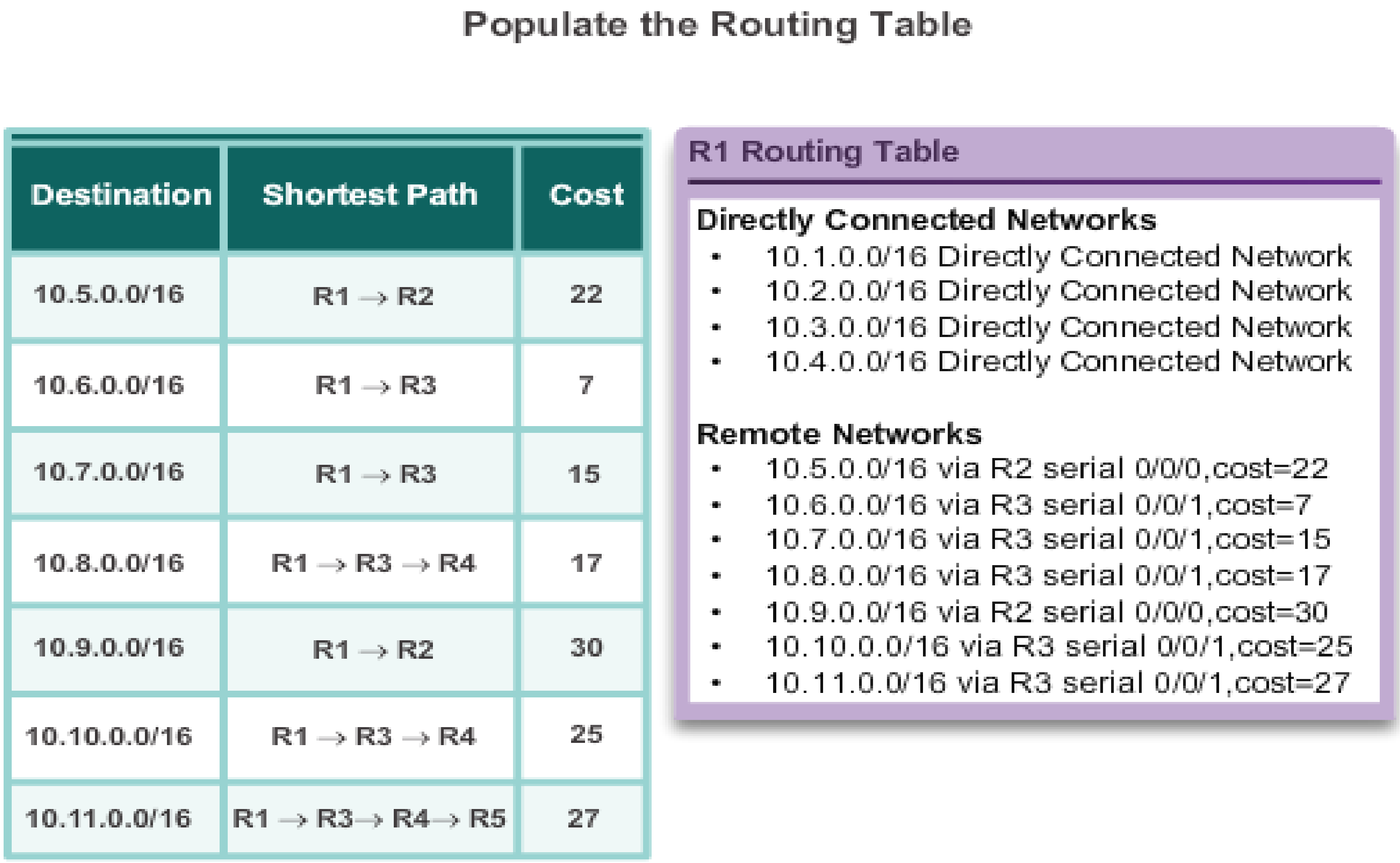
Resulting SPF Tree of R1

Destination	Shortest Path	Cost
10.5.0.0/16	R1 → R2	22
10.6.0.0/16	R1 → R3	7
10.7.0.0/16	R1 → R3	15
10.8.0.0/16	R1 → R3 → R4	17
10.9.0.0/16	R1 → R2	30
10.10.0.0/16	R1 → R3 → R4	25
10.11.0.0/16	R1 → R3→ R4→ R5	27



Link-State Updates

Adding OSPF Routes to the Routing Table



Populate the Routing Table

Destination	Shortest Path	Cost
10.5.0.0/16	R1 → R2	22
10.6.0.0/16	R1 → R3	7
10.7.0.0/16	R1 → R3	15
10.8.0.0/16	R1 → R3 → R4	17
10.9.0.0/16	R1 → R2	30
10.10.0.0/16	R1 → R3 → R4	25
10.11.0.0/16	R1 → R3→ R4→ R5	27

R1 Routing Table

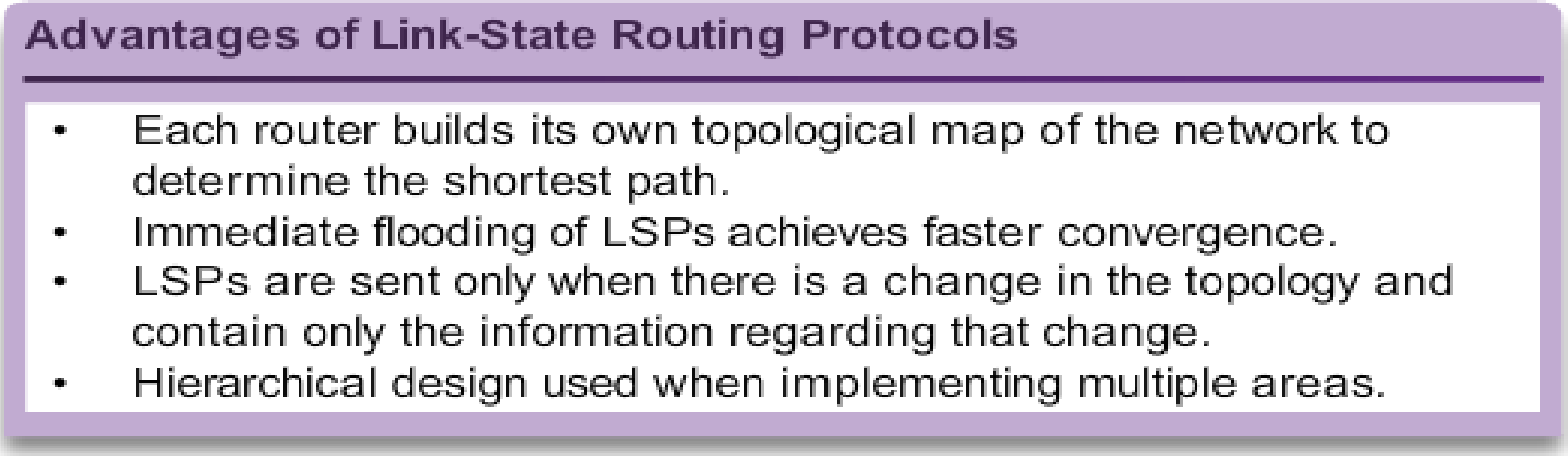
Directly Connected Networks

- 10.1.0.0/16 Directly Connected Network
- 10.2.0.0/16 Directly Connected Network
- 10.3.0.0/16 Directly Connected Network
- 10.4.0.0/16 Directly Connected Network

Remote Networks

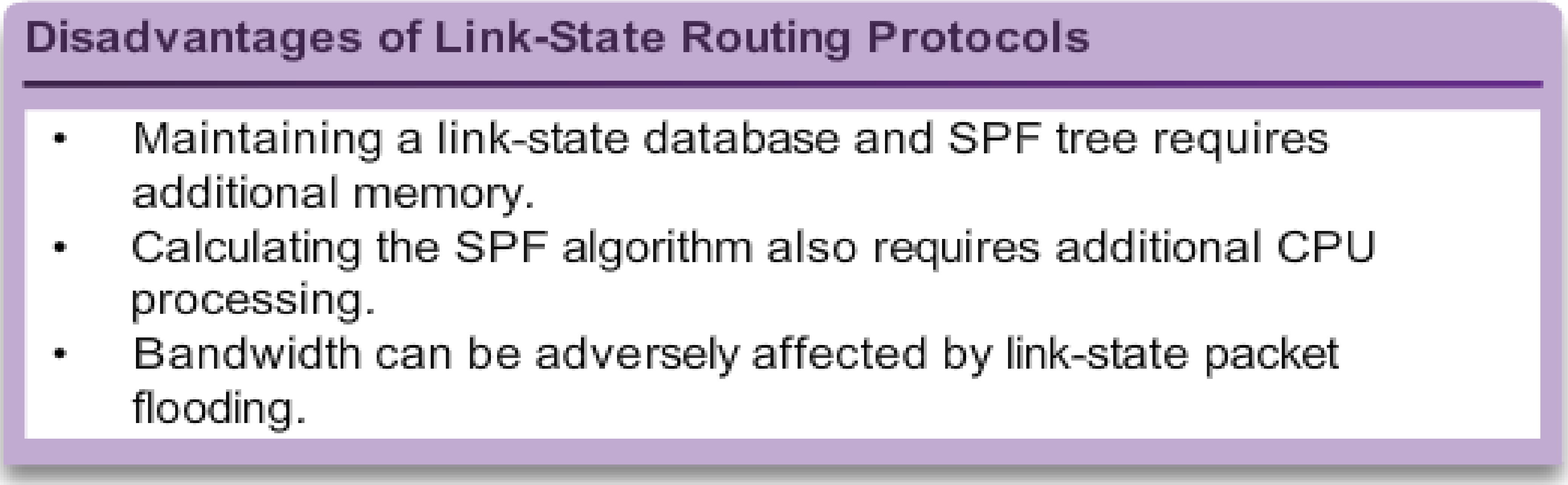
- 10.5.0.0/16 via R2 serial 0/0/0,cost=22
- 10.6.0.0/16 via R3 serial 0/0/1,cost=7
- 10.7.0.0/16 via R3 serial 0/0/1,cost=15
- 10.8.0.0/16 via R3 serial 0/0/1,cost=17
- 10.9.0.0/16 via R2 serial 0/0/0,cost=30
- 10.10.0.0/16 via R3 serial 0/0/1,cost=25
- 10.11.0.0/16 via R3 serial 0/0/1,cost=27

Why Use Link-State Routing Protocols ?



Advantages of Link-State Routing Protocols

- Each router builds its own topological map of the network to determine the shortest path.
- Immediate flooding of LSPs achieves faster convergence.
- LSPs are sent only when there is a change in the topology and contain only the information regarding that change.
- Hierarchical design used when implementing multiple areas.



Disadvantages of Link-State Routing Protocols

- Maintaining a link-state database and SPF tree requires additional memory.
- Calculating the SPF algorithm also requires additional CPU processing.
- Bandwidth can be adversely affected by link-state packet flooding.

IPv6 Features

The ability to scale networks for future demands requires a limitless supply of IP addresses; IPv6 combines expanded addressing with a more efficient and feature-rich header to meet these demands. IPv6 satisfies the increasingly complex requirements of hierarchical addressing that IPv4 does not support.

The main benefits of IPv6 include the following:

■ **Larger address space:** IPv6 addresses are 128 bits, compared to IPv4's 32 bits. This larger addressing space allows more support for addressing hierarchy levels, a much greater number of addressable nodes, and simpler auto configuration of addresses.

■ **Globally unique IP addresses:** Every node can have a unique global IPv6 address, which eliminates the need for NAT.

■ **Header format efficiency:** A simplified header with a fixed header size makes processing more efficient.

■ **Improved privacy and security:** IPsec is the IETF standard for IP network security, available for both IPv4 and IPv6. Although the functions are essentially identical in both environments, IPsec is mandatory in IPv6. IPv6 also has optional security headers.

■ **Flow labeling capability:** A new capability enables the labeling of packets belonging to particular traffic flows for which the sender requests special handling, such as non default quality of service (QoS) or real-time service.

■ **Increased mobility and multicast capabilities:** Mobile IPv6 allows an IPv6 node to change its location on an IPv6 network and still maintain its existing connections. With Mobile IPv6, the mobile node is always reachable through one permanent address. A connection is established with a specific permanent address assigned to the mobile node, and the node remains connected no matter how many times it changes locations and addresses.

IPv6 Address Format

Rather than using dotted-decimal format, IPv6 addresses are written as hexadecimal numbers with colons between each set of four hexadecimal digits (which is 16 bits); we like to call this the "coloned hex" format. The format is x:x:x:x:x:x:x:x, where x is a 16-bit hexadecimal field. A sample address is as follows:

2035:0001:2BC5:0000:0000:087C:0000:000A

Note:

1. We can shorten the written form of IPv6 addresses. Leading 0s within each set of four hexadecimal digits can be omitted, and a pair of colons (::) can be used, once within an address, to represent any number of successive 0s.

   For example, the previous address can be shortened to the following:

   2035:1:2BC5::87C:0:A

   An all-0s address can be written as :: .

2. A pair of colons (::) can be used only once within an IPv6 address. This is because an address parser identifies the number of missing 0s by separating the two parts and entering 0 until the 128 bits are complete. If two :: notations were to be placed in the address, there would be no way to identify the size of each block of 0s.

Example:

3FFE:**0**501:**000**8:**0000**:**0**260:97FF:FE40:EFAB

= 3FFE**:5**01**:8:0:2**60:97FF:FE40:EFAB

= 3FFE:501:8**::**260:97FF:FE40:EFAB

IPv6 Addressing in an Enterprise Network

An IPv6 address consists of two parts:

- **A subnet prefix** representing the network to which the interface is connected. Usually 64-bits in length.

- **An interface ID**, sometimes called a local identifier or a token. Usually 64-bits in length.

IPv6 uses the "/prefix-length" to denote how many bits in the IPv6 address represent the subnet.

The syntax is **ipv6-address/prefix-length**

- ipv6-address is the 128-bit IPv6 address.
- /prefix-length is a decimal value representing how many of the left most contiguous bits of the address comprise the prefix.

<u>For example:</u>

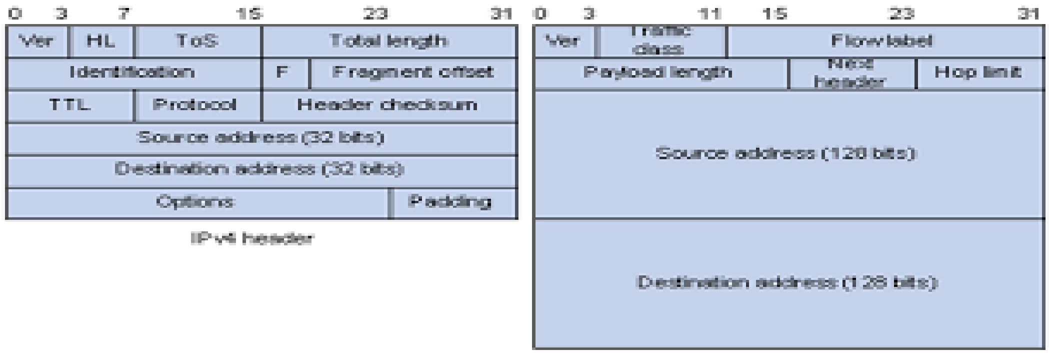
fec0:0:0:1::1234/64

is really

fec0:0000:0000:0001:**0000:0000:0000:1234**/64

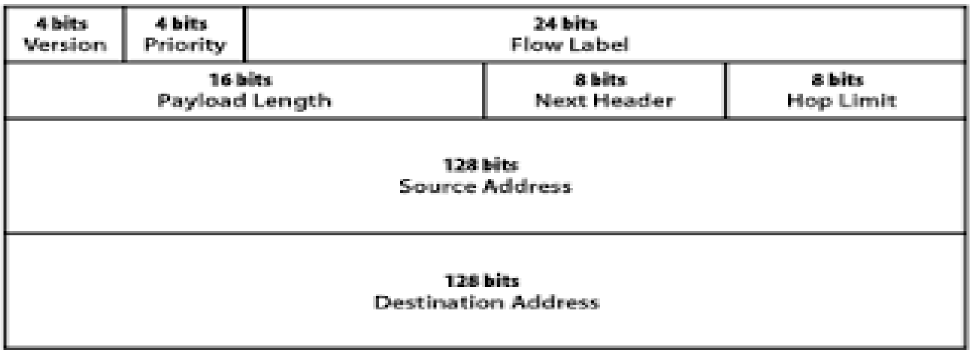
- The first 64-bits (**fec0:0000:0000:0001**) forms the address prefix.
- The last 64-bits (**0000:0000:0000:1234**) forms the Interface ID.

IPv4 and IPv6 Header Format :



IPv4 header

Basic IPv6 header



IPv6 Packet Header Format :

The IPv6 header has 40 octets, in contrast to the 20 octets in the IPv4 header. IPv6 has fewer fields, and the header is 64-bit-aligned to enable fast, efficient, hardware-based processing. The IPv6 address fields are four times larger than in IPv4.

The IPv4 header contains 12 basic header fields, followed by an options field and a data portion (which usually includes a transport layer segment). The basic IPv4 header has a fixed size of 20 octets; the variable-length options field increases the size of the total IPv4 header. IPv6 contains fields similar to 7 of the 12 IPv4 basic header fields (5 plus the source and destination address fields) but does not require the other fields.

The IPv6 header contains the following fields:

- **Version**: A 4-bit field, the same as in IPv4. For IPv6, this field contains the number 6; for IPv4, this field contains the number 4.
- **Traffic class**: An 8-bit field similar to the type of service (ToS) field in IPv4. This field tags the packet with a traffic class that it uses in differentiated services (DiffServ) QoS. These functions are the same for IPv6 and IPv4.
- **Flow label**: This 20-bit field is new in IPv6. It can be used by the source of the packet to tag the packet as being part of a specific flow, allowing multilayer switches and routers to handle traffic on a per-flow basis rather than per-packet, for faster packet-switching performance. This field can also be used to provide QoS.
- **Payload length**: This 16-bit field is similar to the IPv4 total length field.
- **Next header**: The value of this 8-bit field determines the type of information that follows the basic IPv6 header. It can be transport-layer information, such as Transmission Control Protocol (TCP) or User Datagram Protocol (UDP), or it can be an extension header. The *next header field* is similar to the *protocol field* of IPv4.
- **Hop limit**: This 8-bit field specifies the maximum number of hops that an IPv6 packet can traverse. Similar to the time to live (TTL) field in IPv4, each router decreases this field by 1. *Because there is no checksum in the IPv6 header*, an IPv6 router can decrease the field without recomputing the checksum; in IPv4 routers, the recomputation costs processing time. If this field ever reaches 0, a message is sent back to the source of the packet, and the packet is discarded.
- **Source address**: This field has 16 octets (128 bits). It identifies the source of the packet.

- **Destination address**: This field has 16 octets (128 bits). It identifies the destination of the packet.
- **Extension headers**: The extension headers, if any, and the data portion of the packet follow the other eight fields. The number of extension headers is not fixed, so the total length of the extension header chain is variable.

Special IPv6 Addresses

IPv6 Address	Description
::/0	• All routes and used when specifying a default static route. • It is equivalent to the IPv4 quad-zero (0.0.0.0).
::/128	• Unspecified address and is initially assigned to a host when it first resolves its local link address.
::1/128	• Loopback address of local host. • Equivalent to 127.0.0.1 in IPv4.
FE80::/10	• Link-local unicast address. • Similar to the Windows autoconfiguration IP address of 169.254.x.x.
FF00::/8	Multicast addresses.
All other addresses	Global unicast address.

IPv6 Address Scope Types

Similar to IPv4, a single source can address datagrams to either one or many destinations at the same time in IPv6.

Following are the types of IPv6 addresses:

■ **Unicast (one-to-one):** Similar to an IPv4 unicast address, an IPv6 unicast address is for a single source to send data to a single destination. A packet sent to a unicast IPv6 address goes to the interface identified by that address. The IPv6 unicast address space encompasses the entire IPv6 address range, with the exception of the FF00::/8 range (addresses starting with binary 1111 1111), which is used for multicast addresses.

■ **Anycast (one-to-nearest):** An IPv6 anycast address is a new type of address that is assigned to a set of interfaces on different devices; an anycast address identifies multiple interfaces. A packet that is sent to an anycast address goes to the closest interface (as determined by the routing protocol being used) identified by the anycast address. Therefore, all nodes with the same anycast address should provide uniform service. Anycast addresses are syntactically indistinguishable from global unicast addresses because anycast addresses are allocated from the global unicast address space. Nodes to which the anycast address is assigned must be explicitly configured to recognize the anycast address.

■ **Multicast (one-to-many):** Similar to IPv4 multicast, an IPv6 multicast address identifies a set of interfaces (in a given scope), typically on different devices.

A packet sent to a multicast address is delivered to all interfaces identified by the multicast address (in a given scope). IPv6 multicast addresses have a 4-bit scope identifier (ID) to specify how far the multicast packet may travel.

<u>Scope:</u>

- 1 (0001) = Node
- 2 (0010) = Link
- 5 (0101) = Site
- 8 (1000) = Organization
- E (1110) = Global

<u>Note:</u>

- IPv6 has no concept of broadcast addresses; multicast addresses are used instead.

- A single interface may be assigned multiple IPv6 addresses of any type (unicast, anycast, and multicast).

IPv6 Unicast Address

Following are the different unicast addresses that IPv6 supports:

■ Global aggregatable address (also called global unicast address)

■ Link-local address

■ IPv4-compatible IPv6 address

IPv4-to-IPv6 Transition Strategies and Deployments

IPv4-to-IPv6 migration does not happen automatically. The following sections first explore the differences between IPv4 and IPv6 and then discuss possible transition strategies and deployments.

Differences Between IPv4 and IPv6:

Regardless of which protocol is used, the communication between IPv4 and IPv6 domains must be transparent to end users. The major differences to consider between IPv4 and IPv6 include the following:

■ IPv4 addresses are 32 bits long, whereas IPv6 addresses are 128 bits long.

■ An IPv6 packet header is different from an IPv4 packet header. The IPv6 header is longer and simpler (new fields were added to the IPv6 header, and some old fields were removed).

■ IPv6 has no concept of broadcast addresses; instead, it uses multicast addresses.

■ Routing protocols must be changed to support native IPv6 routing.

IPv4-to-IPv6 Transition

The transition from IPv4 to IPv6 will take several years because of the high cost of upgrading equipment. In the meantime, IPv4 and IPv6 must coexist. The following are three primary mechanisms for the transition from IPv4 to IPv6:

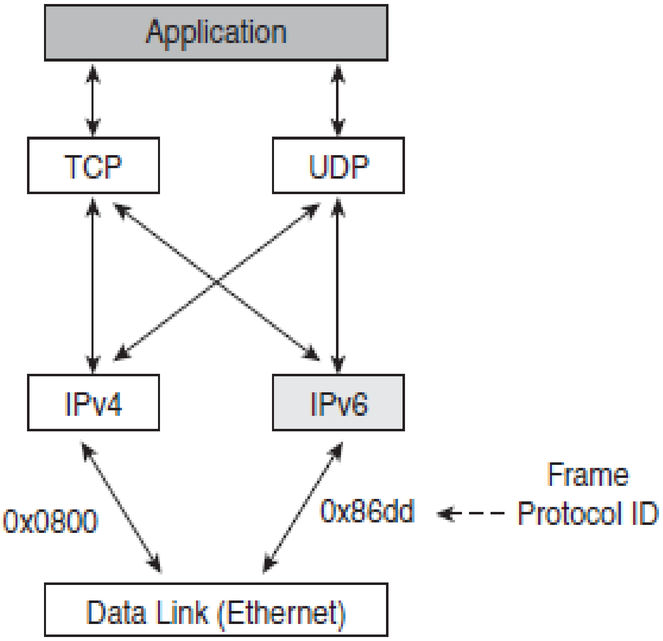
■ **Dual-stack**: Both the IPv4 and the IPv6 stacks run on a system that can communicate with both IPv6 and IPv4 devices.

■ **Tunneling**: Uses encapsulation of IPv6 packets to traverse IPv4 networks, and vice versa.

■ **Translation**: A mechanism that translates one protocol to the other to facilitate communication between the two networks.

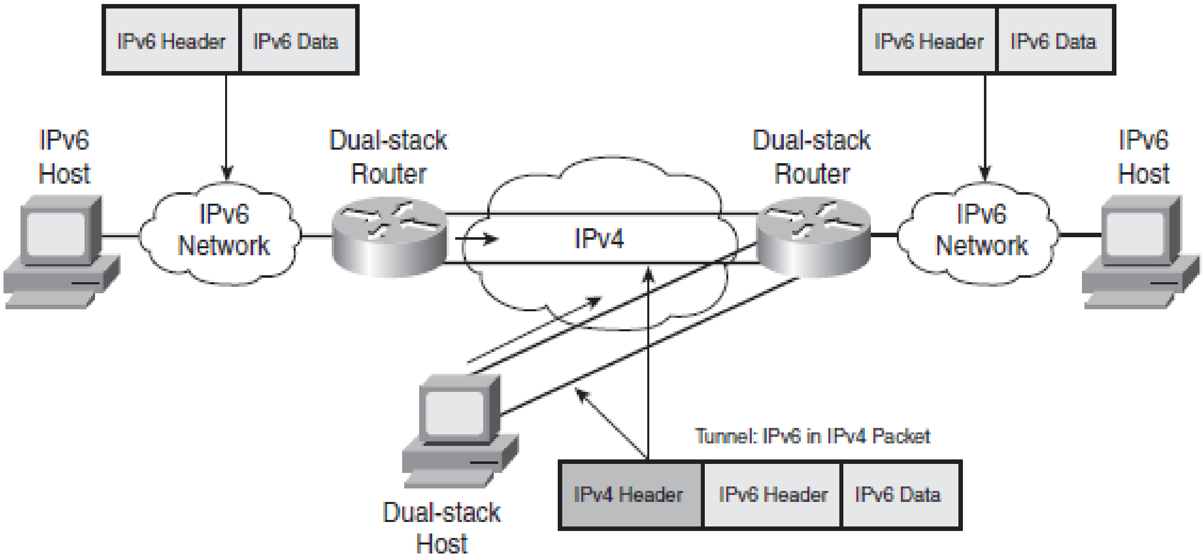
Dual-Stack Transition Mechanism

a dual-stack node enables both IPv4 and IPv6 stacks. Applications communicate with both IPv4 and IPv6 stacks; the IP version choice is based on name lookup and application preference. This is the most appropriate method for campus and access networks during the transition period, and it is the preferred technique for transitioning to IPv6. A dual-stack approach supports the maximum number of applications. Operating systems that support the IPv6 stack include FreeBSD, Linux, Sun Solaris, and Windows 2000, XP, and Vista



<u>Tunneling Transition Mechanism</u>

The purpose of tunneling is to encapsulate packets of one type in packets of another type. When transitioning to IPv6, tunneling encapsulates IPv6 packets in IPv4 packets, as shown in the following figure.



By using overlay tunnels, isolated IPv6 networks can communicate without having to upgrade the IPv4 infrastructure between them. Both routers and hosts can use tunneling. The following different techniques are available for establishing a tunnel:

■ **Manually configured:** For a manually configured tunnel, the tunnel source and tunnel destination are manually configured with static IPv4 and IPv6 addresses. Manual tunnels can be configured between border routers or between a border router and a host.

■ **Semi-automated:** Semi-automation is achieved by using a tunnel broker that uses a web based service to create a tunnel. A tunnel broker is a server on the IPv4 network that receives tunnel requests from dual-stack clients, configures the tunnel on the tunnel server or router, and associates the tunnel from the client to one of the tunnel servers or routers. A simpler model combines the tunnel broker and server onto one device.

■ **Automatic:** Various automatic mechanisms accomplish tunneling, including the following: