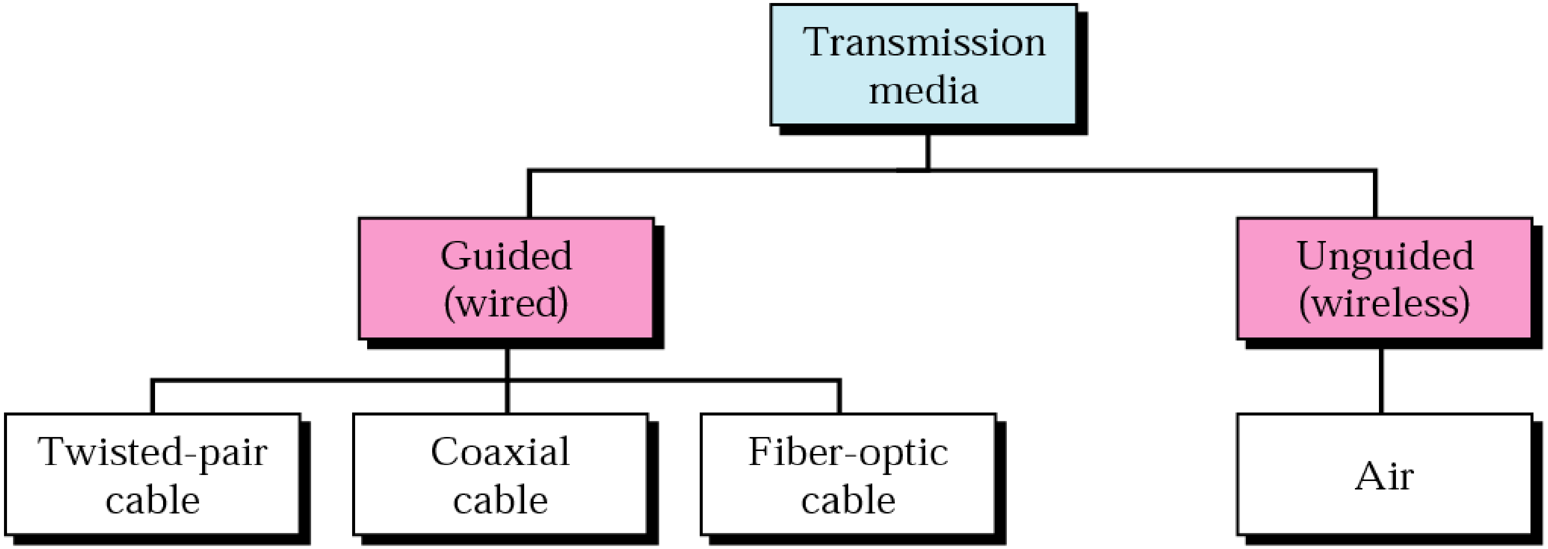
NETWORK

POWER⏻UNIT

Classes of transmission media



Transmission Media

- Guided media, which are those that provide a conduit from one device to another.
- Examples: twisted-pair, coaxial cable, optical fiber.
- Unguided media (or wireless communication) transport electromagnetic waves without using a physical conductor. Instead, signals are broadcast through air (or, in a few cases, water), and thus are available to anyone who has a device capable of receiving them.

Guided Media

There are three categories of guided media:

1. Twisted-pair cable
2. Coaxial cable
3. Fiber-optic cable

Twisted-pair cable

- Twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together.
- Twisted-pair cable comes in two forms: unshielded and shielded
- The twisting helps to reduce the interference (noise) and crosstalk.

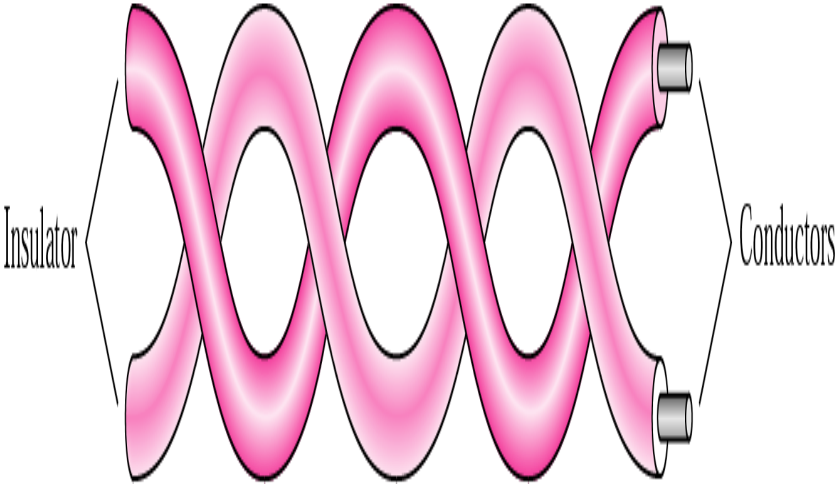


Figure 7.4 *Frequency range for twisted-pair cable*

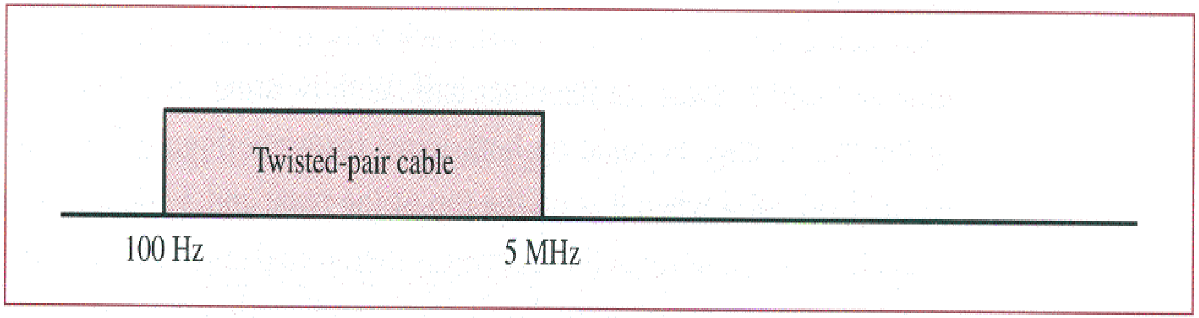
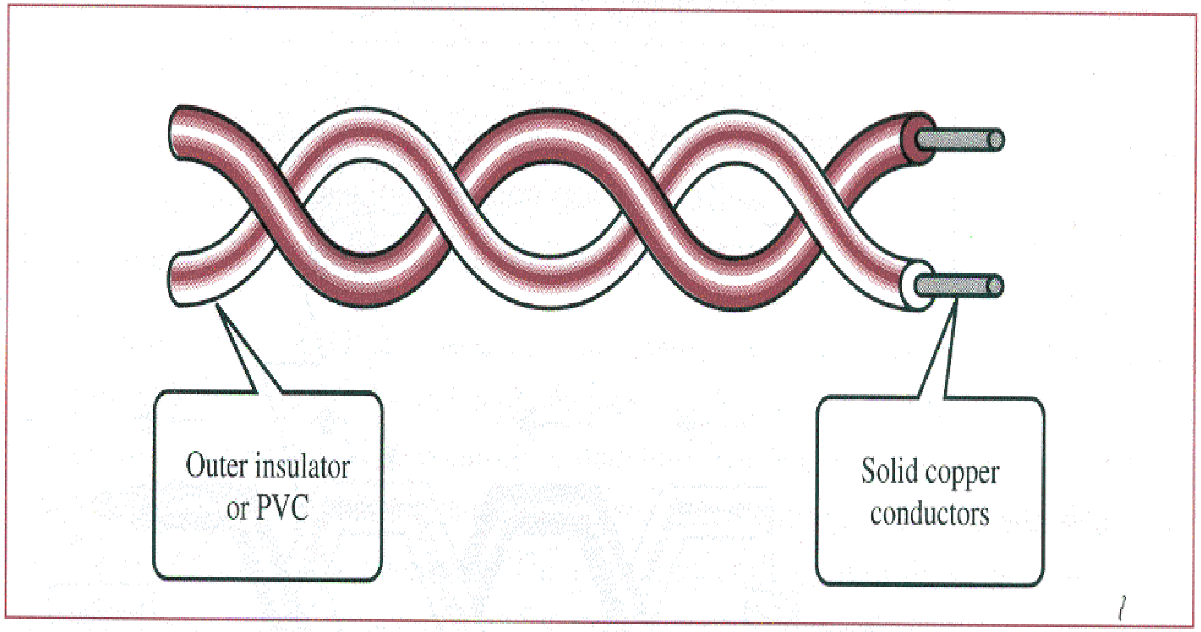
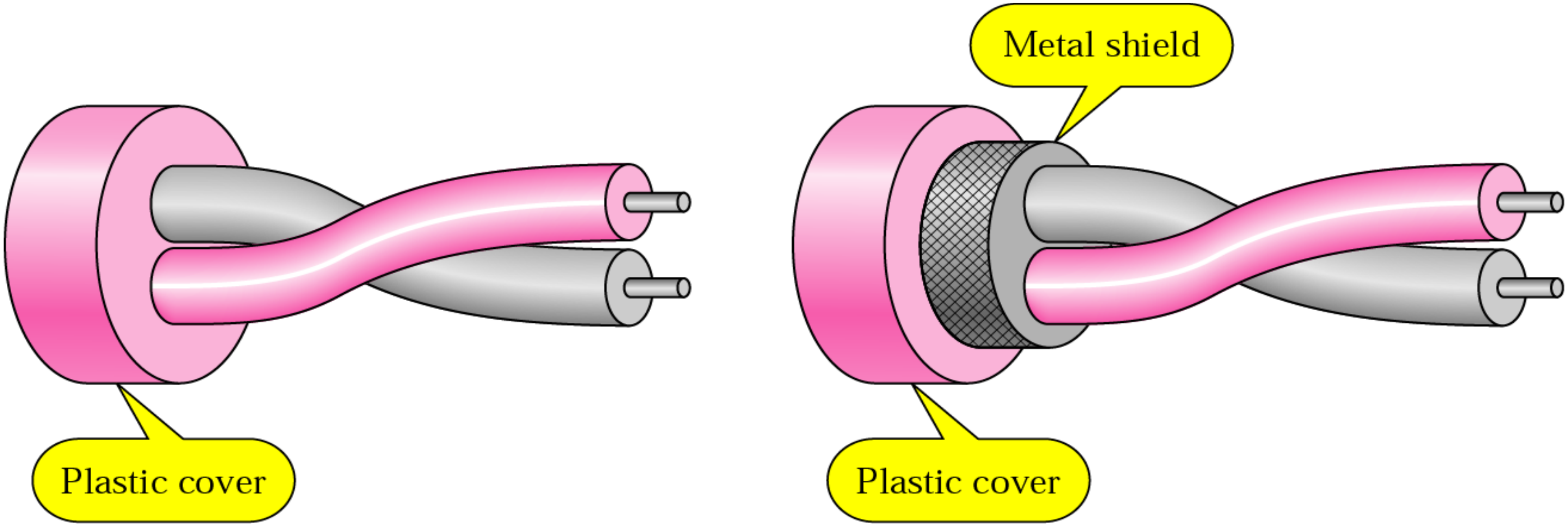


Figure 7.5 *Twisted-pair cable*



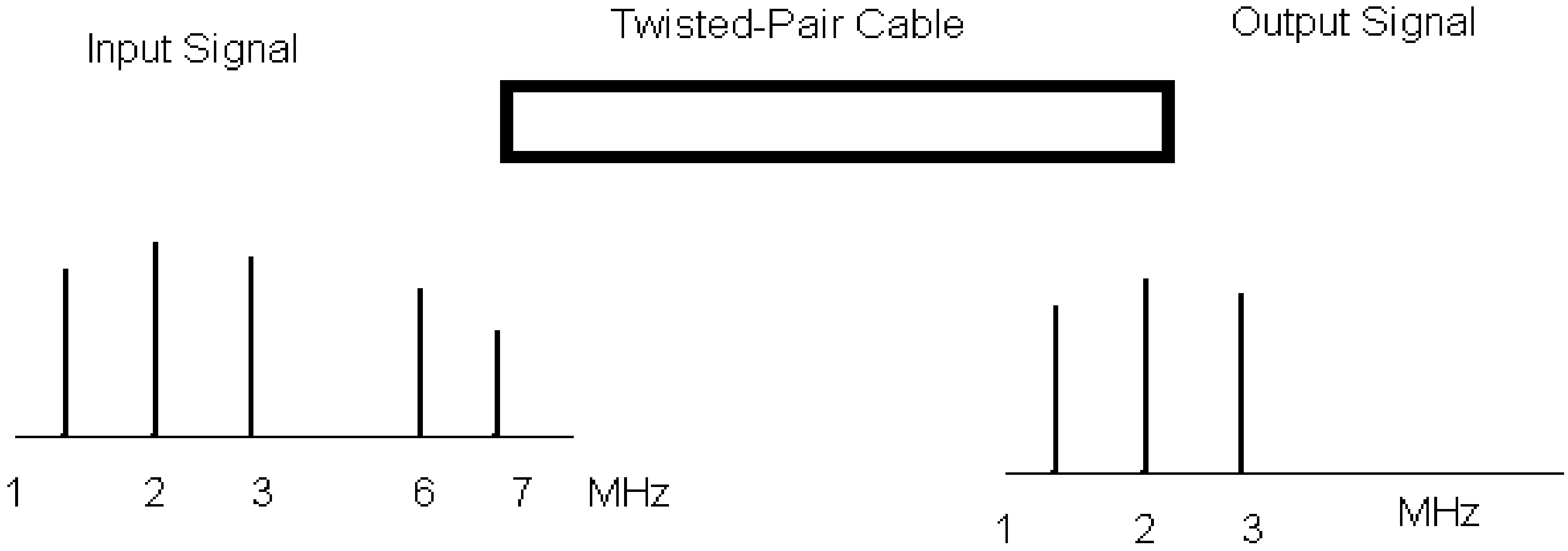
UTP and STP



a. UTP

b. STP

Frequency range for twisted-pair cable

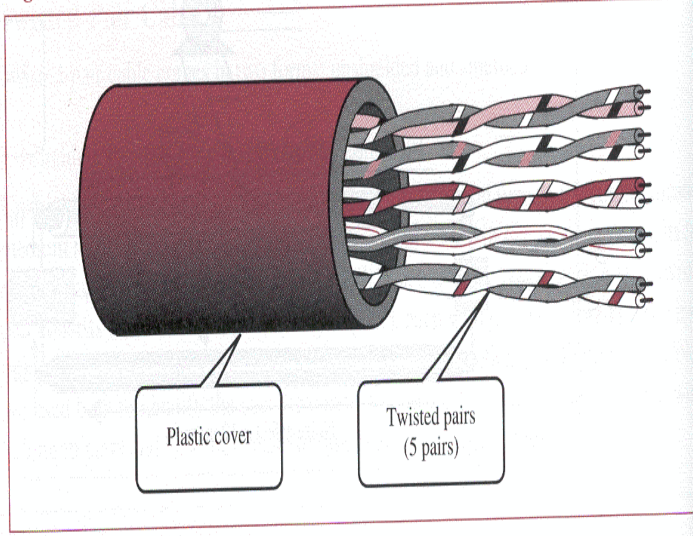


Unshielded Twisted-pair (UTP) cable

- Any medium can transmit only a fixed range of frequencies!
- UTP cable is the most common type of telecommunication medium in use today.

- The range is suitable for transmitting both data and video.
- Advantages of UTP are its cost and ease of use. UTP is cheap, flexible, and easy to install.

Figure 7.8 *Cable with five unshielded twisted pairs of wires*



The Electronic Industries Association (EIA) has developed standards to grade UTP.

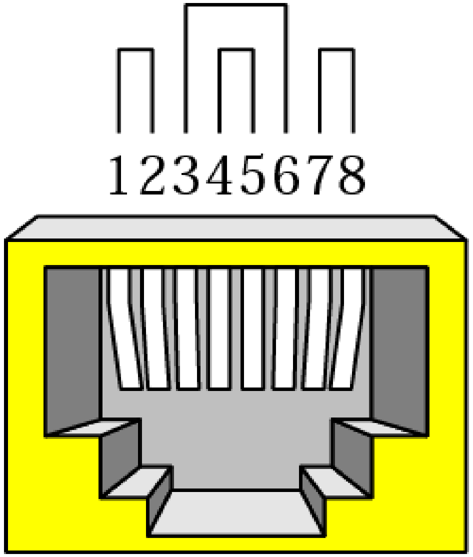
1. Category 1. The basic twisted-pair cabling used in telephone systems. This level of quality is fine for voice but inadequate for data transmission.
2. Category 2. This category is suitable for voice and data transmission of up to 2Mbps.
3. Category 3.This category is suitable for data transmission of up to 10 Mbps. It is now the standard cable for most telephone systems.
4. Category 4. This category is suitable for data transmission of up to 20 Mbps.
5. Category 5. This category is suitable for data transmission of up to 100 Mbps.

Table 7.1 Categories of unshielded twisted-pair cables

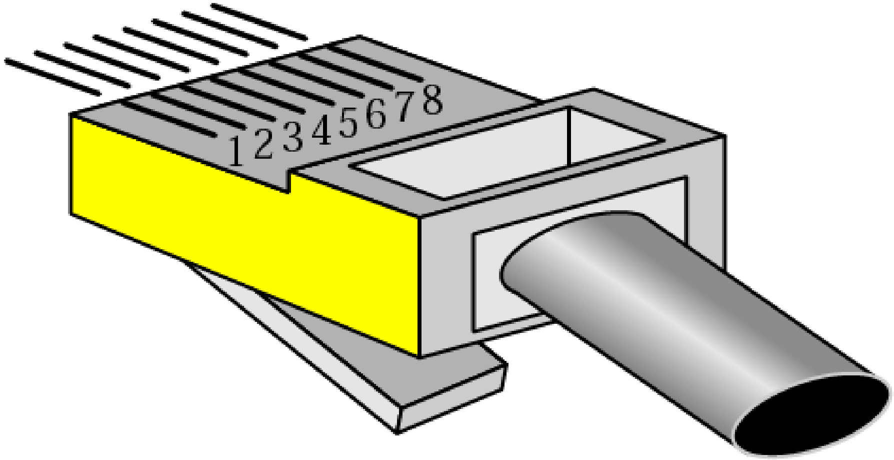
Category	Bandwidth	Data Rate	Digital/Analog	Use
1	very low	< 100 kbps	Analog	Telephone
2	< 2 MHz	2 Mbps	Analog/digital	T-1 lines
3	16 MHz	10 Mbps	Digital	LANs
4	20 MHz	20 Mbps	Digital	LANs
5	100 MHz	100 Mbps	Digital	LANs
6 (draft)	200 MHz	200 Mbps	Digital	LANs
7 (draft)	600 MHz	600 Mbps	Digital	LANs

UTP connectors

The most common UTP connector is RJ45 (RJ stands for Registered Jack).



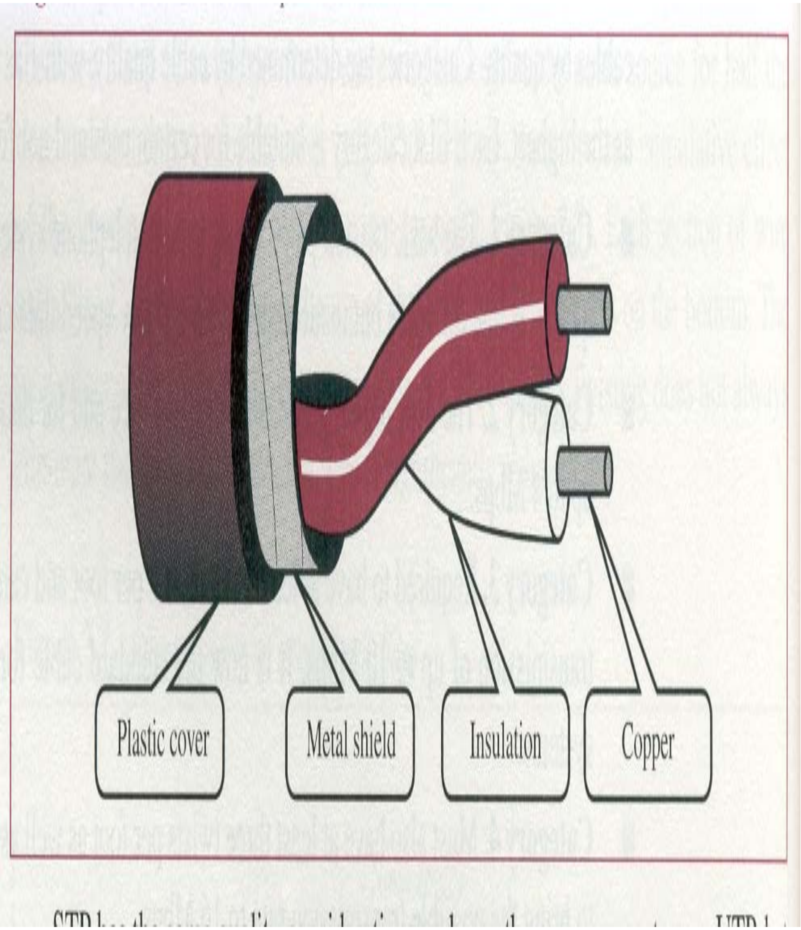
RJ-45 Female



RJ-45 Male

Shielded Twisted (STP) Cable

- STP cable has a metal foil or braided-mesh covering that enhances each pair of insulated conductors.
- The metal casing prevents the penetration of electromagnetic noise.
- Materials and manufacturing requirements make STP more expensive than UTP but less susceptible to noise.



Applications

- Twisted-pair cables are used in telephones lines to provide voice and data channels.
- The DSL lines that are used by the telephone companies to provide high data rate connections also use the high-bandwidth capability of unshielded twisted-pair cables.
- Local area networks, such as 10Base-T and 100Base-T, also used UTP cables.

Coaxial Cable (or coax)

- Coaxial cable carries signals of higher frequency ranges than twisted-pair cable.

- Coaxial Cable standards:

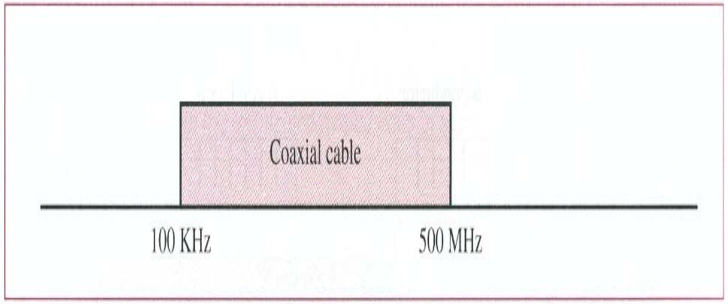
RG-8, RG-9, RG-11 are

used in thick Ethernet

RG-58 Used in thin Ethernet

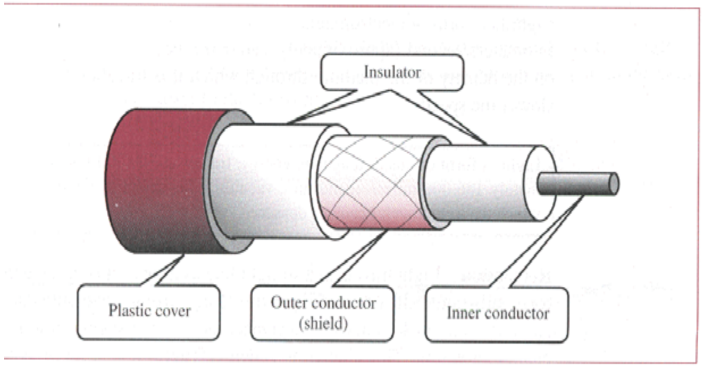
RG-59 Used for TV

Figure 7.11 *Frequency range of coaxial cable*



Coaxial Cable Standards

Figure 7.12 *Coaxial cable*

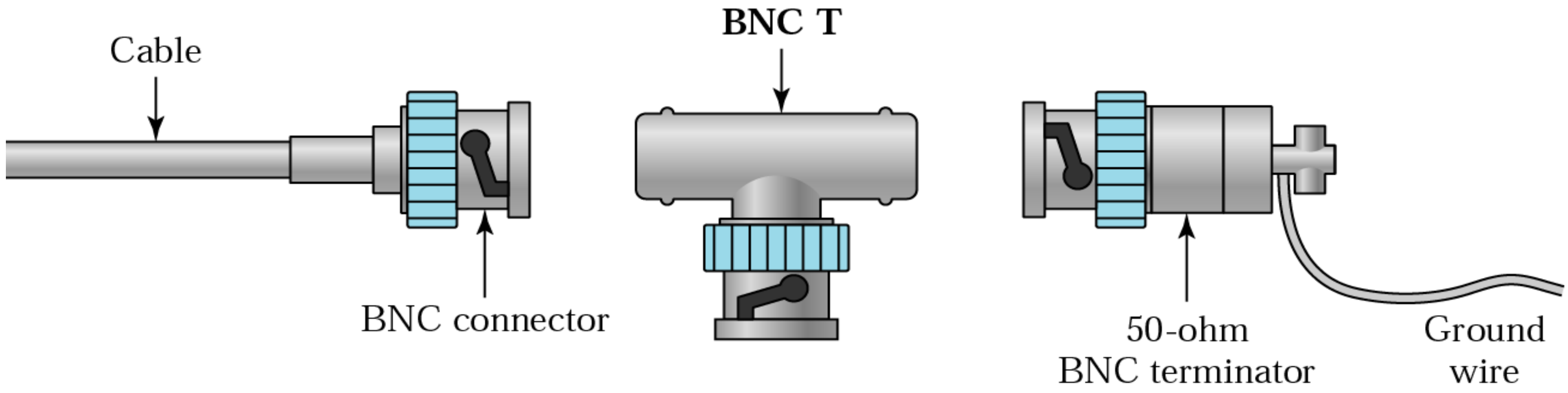


RG-58 Used in thin Ethernet

BNC connectors

•To connect coaxial cable to devices, it is necessary to use coaxial connectors. The most common type of connector is the Bayone-Neill-Concelman, or BNC, connectors. There are three types: the BNC connector, the BNC T connector, the BNC terminator.

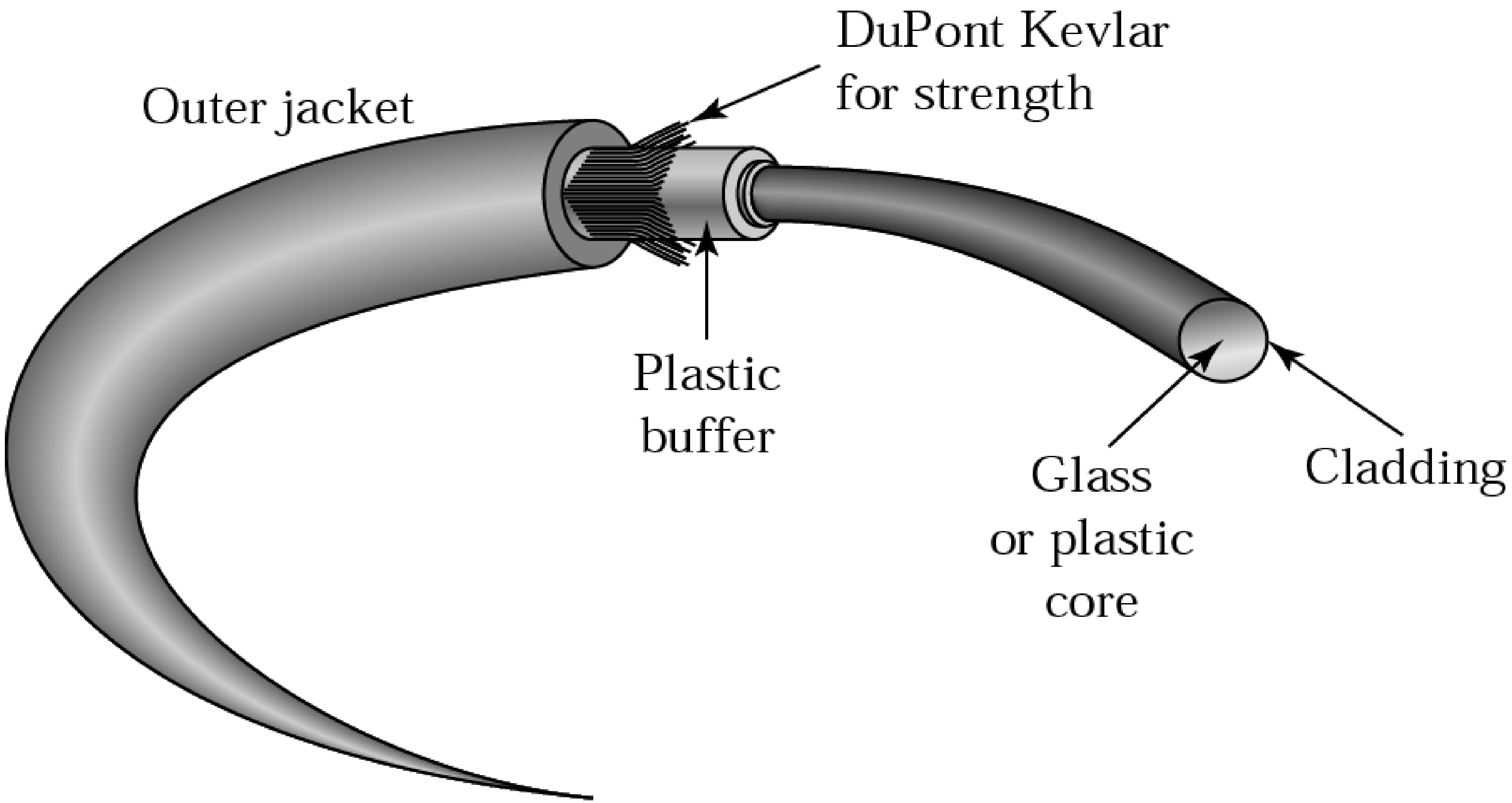
Applications include cable TV networks, and some traditional Ethernet LANs like 10Base-2, or 10-Base5.



Optical Fiber

- Metal cables transmit signals in the form of electric current.
- Optical fiber is made of glass or plastic and transmits signals in the form of **light**.
- Light, a form of electromagnetic energy, travels at 300,000 Kilometers/second (186,000 miles/second), in a vacuum.
- The speed of the light depends on the density of the medium through which it is traveling (the higher density, the slower the speed).

Fiber construction



Types of Optical Fiber

- **There are two basic types of fiber: multimode fiber and single-mode fiber.**

- **Multimode fiber is best designed for short transmission distances, and is suited for use in LAN systems**

- **Single-mode fiber is best designed for longer transmission distances, making it suitable for long-distance telephony and multichannel television broadcast systems.**

Advantages of Optical Fiber

- The major advantages offered by fiber-optic cable over twisted-pair and coaxial cable are **noise resistance, less signal attenuation, and higher bandwidth**.

- **Noise Resistance**: Because fiber-optic transmission uses light rather than electricity, noise is not a factor. External light, the only possible interference, is blocked from the channel by the outer jacket.

Advantages of Optical Fiber

- **Less signal attenuation**

Fiber-optic transmission distance is significantly greater than that of other guided media. A signal can run for miles without requiring regeneration.

- **Higher bandwidth**

Currently, data rates and bandwidth utilization over fiber-optic cable are limited not by the medium but by the signal generation and reception technology available.

Disadvantages of Optical Fiber

- The main disadvantages of fiber optics are **cost, installation/maintenance, and fragility**.
- Cost. Fiber-optic cable is expensive. Also, a laser light source can cost thousands of dollars, compared to hundreds of dollars for electrical signal generators.
- Installation/maintenance
- Fragility. Glass fiber is more easily broken than wire, making it less useful for applications where hardware portability is required.

Internet Access Wired Networks:

- Public Switched Telephone Network
    - ➢ Dial-up Internet Access
    - ➢ Digital Subscriber Line
- Integrated Services Digital Network
- TV-Cable Networks

1. Public Switched Telephone Network

- Historically, most telephone connections in the world have been made through the public switched telephone network (PSTN).
- Most PSTN calls are transmitted digitally except while in the local loop, the part of the telephone network between the telephone and the telephone company's central switching office. Within this loop, speech from a telephone is usually transmitted in analog format.
- Digital data from a computer must first be converted to analog by a modem.
- The modem is installed in the computer, connected to the computer by the serial port, or by a Universal Serial Bus connection.
- The data is converted at the receiving end by another modem, which changes the data from audio to its original data form.

A. Dial-up Internet Access

- **Dial-up Internet access** is a form of Internet access that uses the facilities of the public switched telephone network (PSTN) to establish a dialed connection to an Internet service provider (ISP) via telephone lines. To achieve that, a modem is required. The dial-up speed is 56Kbps.

B. Digital Subscriber Line

- **Digital subscriber line** (**DSL**; originally **digital subscriber loop**) is a family of technologies that provide internet access by transmitting digital data using a local telephone network which uses the public switched telephone network.
- When you connect to the Internet, you might connect through a regular modem, through a local-area network connection in your office, through a cable modem or through a **digital subscriber line** (DSL) connection.
- In telecommunications marketing, the term DSL is widely understood to mean asymmetric digital subscriber line (ADSL), the most commonly installed DSL technology.
- DSL service is delivered simultaneously with wired telephone service on the same telephone line. This is possible because DSL uses higher frequency bands for data.
- The bit rate of consumer DSL services typically ranges from 256 kbit/s to over 100 Mbit/s in the direction to the customer (downstream), depending on DSL technology, line conditions, and service-level implementation. Bit rates of 1 Gbit/s have been reached in trials. In ADSL, the data throughput in the upstream direction, (the direction to the service provider) is lower, hence the designation of *asymmetric* service.
- In symmetric digital subscriber line (SDSL) services, the downstream and upstream data rates are equal.

Disadvantages:

- A DSL connection works better when you are closer to the provider's central office. The farther away you get from the central office, the weaker the signal becomes.
- The connection is faster for receiving data than it is for sending data over the Internet.
- The service is not available everywhere.

2. Integrated Services Digital Network

- The need for high-speed telecommunications support within the existing telecommunications infrastructure has led to the development of new technologies, such as Integrated Services Digital Network (ISDN). ISDN is a digital phone service that is provided by regional and national phone companies, using existing copper telephone cabling.

- ISDN in concept is the integration of both analog or voice data together with digital data over the same network.

- There are two levels of service: the Basic Rate Interface (BRI), intended for the home and small enterprise, and the Primary Rate Interface (PRI), for larger users. Both rates include a number of B-channels and a D-channels. Each B-channel carries data, voice, and other services. Each D-channel carries control and signaling information.

- In many areas where DSL and cable modem service are now offered, ISDN is no longer as popular an option as it was formerly.

- To use ISDN, you need either an ISDN modem or an ISDN adapter. In other words, ISDN requires adapters at both ends of the transmission so your access provider also needs an ISDN adapter.

- ISDN modems are available in internal and external configurations. Internal ISDN modems, the more common of the two, are installed in the same manner as a network adapter card. External ISDN modems hook up to your computer through a serial port, just as regular modems do. Thus, because a serial port cannot exceed 115 kilobits per second (Kbps) (which is lower than the total effective bandwidth of the ISDN line), some throughput is lost if you are using the maximum ISDN bandwidth.

- ISDN is typically supplied by the same company that supports the public switched telephone network. However, ISDN differs from analog telephone service in several ways, including:

  - ✓ Data transfer rate
  - ✓ Available channels per call
  - ✓ Availability of service
  - ✓ Cost of service
  - ✓ Quality of connection

<u>Data Transfer Rate:</u>

- ➢ The Basic Rate Interface consists of two 64 Kbps B-channels and one 16 Kbps D-channel. Thus, a Basic Rate user can have up to 128 Kbps service. The Primary Rate consists of 23 B-channels and one 64 Kbps D-channel in the United States or 30 B-channels and 1 D-channel in Europe.

- ➢ These speeds are slower than those of local area networks (LANs) supported by high-speed data communications technology, but faster than those of analog telephone lines.

- ➢ In addition to the difference in data transfer rates, ISDN calls can be established much faster than analog phone calls. While an analog modem can take up to a minute to set up a connection, you usually can start transmitting data in about two seconds with ISDN.

- ➢ Because ISDN is fully digital, the lengthy process of analog modems is not required.

Channels:

- ISDN service is available in several configurations of multiple channels, each of which can support voice or digital communications.
- In addition to increasing data throughput, multiple channels eliminate the need for separate voice and data telephone lines.

Availability:

- ISDN and PSTN are available throughout the United States.

Cost:

- The cost of ISDN hardware and service is higher than for PSTN modems and service.

Connection Quality:

- ISDN transmits data digitally and, as a result, is less susceptible to static and noise than analog transmissions. Analog modem connections must dedicate some bandwidth to error correction and retransmission. This overhead reduces the actual throughput. In contrast, an ISDN line can dedicate all its bandwidth to data transmission.

3. TV-Cable Networks

- A cable modem is a device that enables you to hook up your PC to a local cable TV line and receive data at about 1.5 Mbps (start service rate). This data rate far exceeds that of 56 Kbps telephone modems and the up to 128 Kbps of Integrated Services Digital Network (ISDN) and is about the data rate available to subscribers of Digital Subscriber Line (DSL) telephone service.

- A cable modem has two connections: one to the cable wall outlet and the other to a PC.

- Many people who have cable TV can now get a high-speed connection to the Internet from their cable provider. Cable modems compete with technologies like asymmetrical digital subscriber lines (ADSL).

- Cable internet access is generally offered by the same companies that provide cable TV. It works on the same coaxial cable that the TV signal comes in on, but doesn't effect your TV signal. Therefore you can use the internet and watch TV at the same time.

- Typically, cable internet access provides a maximum of 1.5 - 150Mbps of bandwidth on the system. However, everyone on your network segment is sharing that bandwidth, so performance can be much lower, especially if a lot of people in your neighborhood use the service. They may also limit your individual bandwidth, so that you will never see the peak bandwidth even when your network segment is clear.

- Since you are sharing the network segment with other users, there can be security risks with cable modems.

- Since few office buildings are wired for cable TV, it's not a practical option for most businesses. As more cable providers reach out to the business market, however, cable Internet access will be a good alternative to DSL service.

Data Link Layer

Sublayers:

- Sublayers of the data link layer
  - Logical link control sublayer
  - Media access control sublayer

Data link layer services:

- Encapsulation of network layer data packets into frames
- Frame synchronization- Done by Media Access Control sublayer
- Logical link control (LLC) sublayer:
  - Error control (automatic repeat request, ARQ)
  - Flow control
- Media access control (MAC) sublayer:
  - Multiple access protocols for channel-access control, for example CSMA/CD protocols for collision detection and re-transmission in Ethernet bus networks and hub networks, or the CSMA/CA protocol for collision avoidance in wireless networks.
  - Physical addressing (MAC addressing)
  - LAN switching (packet switching) including MAC filtering and spanning tree protocol
  - Data packet queuing or scheduling
  - Store-and-forward switching or cut-through switching
  - Quality of Service (QoS) control
  - Virtual LANs (VLAN)

The uppermost sublayer, LLC, multiplexes protocols running atop the data link layer, and optionally provides flow control, acknowledgment, and error notification.

More about MAC Sublayer:

1. The Media Access Control sublayer also determines where one frame of data ends and the next one starts – frame synchronization. There are four means of frame synchronization: time based, character counting, byte stuffing and bit stuffing.
2. The Spanning Tree Protocol (STP) is a network protocol that ensures a loop-free topology for any bridged Ethernet local area network.
3. Store and forward is a telecommunications technique in which information is sent to an intermediate station where it is kept and sent at a later time to the final destination or to another intermediate station. The intermediate station, or node in a networking context, verifies the integrity of the message before forwarding it. This technique originates the delay-tolerant networks. No real-time services are available for these kinds of networks.

4. In computer networking, cut-through switching is a method for packet switching systems, wherein the switch starts forwarding a frame (or packet) before the whole frame has been received, normally as soon as the destination address is processed. Compared to store and forward, this technique reduces latency through the switch and relies on the destination devices for error handling.

Once More:

- In the seven-layer OSI model of computer networking, media access control (MAC) data communication protocol is a sublayer of the data link layer (layer 2). The MAC sublayer provides addressing and channel access control mechanisms that make it possible for several terminals or network nodes to communicate within a multiple access network that incorporates a shared medium, e.g. Ethernet. The hardware that implements the MAC is referred to as a *medium access controller*.
- The channel access control mechanisms provided by the MAC layer are also known as multiple access protocols.
- The most widespread multiple access protocol is the contention based CSMA/CD protocol used in Ethernet networks. This mechanism is only utilized within a network collision domain, for example an Ethernet bus network or a hub-based star topology network. An Ethernet network may be divided into several collision domains, interconnected by bridges and switches.

Data link layer protocols:

- Ethernet
- Token ring
- Fiber Distributed Data Interface (FDDI)
- Address Resolution Protocol (ARP)
- High-Level Data Link Control (HDLC)
- Serial Line Internet Protocol (SLIP)
- Point-to-Point Protocol (PPP)
- Asynchronous Transfer Mode (ATM)
- IEEE 802.11 wireless LAN
- Frame Relay

Common multiple access protocols:

Examples of common statistical time division multiplexing multiple access protocols for wired multi-drop networks are:

- CSMA/CD (used in Ethernet or IEEE 802.3)
- Token ring (IEEE 802.5)
- Token passing (used in FDDI)

Examples of common multiple access protocols that may be used in packet radio wireless networks are:

- CSMA/CA (used in IEEE 802.11/WiFi WLANs)
- Slotted ALOHA
- Dynamic TDMA
- Reservation ALOHA (R-ALOHA)
- Mobile Slotted Aloha (MS-ALOHA)
- CDMA
- OFDMA

Address Resolution Protocol (ARP):

- ARP is used to convert an IP address to a physical address such as an Ethernet address (also known as a MAC address).

- For example the computers *Matterhorn* and *Washington* are in an office, connected to each other on the office local area network by Ethernet cables and network switches, with no intervening gateways or routers. Matterhorn wants to send a packet to Washington. Through DNS, it determines that Washington's IP address is 192.168.0.55. In order to send the message, it also needs to know Washington's MAC address. First, Matterhorn uses a cached ARP table to look up 192.168.0.55 for any existing records of Washington's MAC address (00:eb:24:b2:05:ac). If the MAC address is found, it sends the IP packet encapsulated in a level 2 frame on the link layer to address 00:eb:24:b2:05:ac via the local network cabling. If the cache did not produce a result for 192.168.0.55, Matterhorn has to send a broadcast ARP message (destination FF:FF:FF:FF:FF:FF MAC address which is accepted by all computers) requesting an answer for 192.168.0.55. Washington responds with its MAC address (and its IP). Washington may insert an entry for Matterhorn into its own ARP table for future use. The response information is cached in Matterhorn's ARP table and the message can now be sent.

- There is also the Reverse Address Resolution Protocol (Reverse ARP or RARP). RARP is obsolete; it was replaced by BOOTP, which was later superseded by the Dynamic Host Configuration Protocol (DHCP).

- The **Dynamic Host Configuration Protocol** (**DHCP, application layer protocol**) is a standardized networking protocol used on Internet Protocol (IP) networks for dynamically distributing network configuration parameters, such as IP addresses for interfaces and services. With DHCP, computers request IP addresses and networking parameters automatically from a DHCP server, reducing the need for a network administrator or a user to configure these settings manually.

More about Dynamic Host Configuration Protocol:

DHCP assigns an IP address when a system is started, for example:

1. A user turns on a computer with a DHCP client.
2. The client computer sends a broadcast request (called a DISCOVER or DHCPDISCOVER), looking for a DHCP server to answer.
3. The router directs the DISCOVER packet to the correct DHCP server.
4. The server receives the DISCOVER packet. Based on availability and usage policies set on the server, the server determines an appropriate address (if any) to give to the client. The server then temporarily reserves that address for the client and sends back to the client an OFFER (or DHCPOFFER) packet, with that address information. The server also configures the client's DNS servers, WINS servers, NTP servers, and sometimes other services as well.
5. The client sends a REQUEST (or DHCPREQUEST) packet, letting the server know that it intends to use the address.
6. The server sends an ACK (or DHCPACK) packet, confirming that the client has a been given a lease on the address for a server-specified period of time.

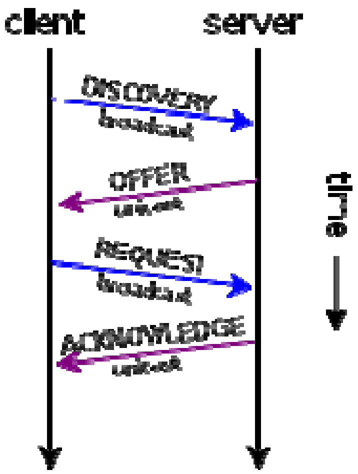


Diagram of a typical DHCP session

When a computer uses a static IP address, it means that the computer is manually configured to use a specific IP address. One problem with static assignment, which can result from user error or inattention to detail, occurs when two computers are configured with the same IP address. This creates a conflict that results in loss of service. Using DHCP to dynamically assign IP addresses minimizes these conflicts.

High Level Link Control (HDLC) Protocol:

- The HDLC protocol is a general purpose protocol which operates at the data link layer of the OSI reference model. The protocol uses the services of a physical layer, and provides either a *best effort* or *reliable* communications path between the transmitter and receiver (i.e. with acknowledged data transfer) (i.e., HDLC provides both connection-oriented and connectionless services).

- The type of service provided depends upon the HDLC mode which is used.

- Each piece of data is encapsulated in an HDLC frame by adding a trailer and a header. The header contains an HDLC address and an HDLC control field. The trailer is found at the end of the frame, and contains a Cyclic Redundancy Check (CRC) which detects any errors which may occur during transmission. The frames are separated by HDLC flag sequences which are transmitted between each frame and whenever there is no data to be transmitted.

- The two most prevalent HDLC modes are:

  - ✓ **The best-effort or datagram service**. In this mode, the packets are carried in a UI frame, and a best-effort delivery is performed (i.e. there is no guarantee that the packet carried by the frame will be delivered.) The link layer does not provide error recovery of lost frames. The best-effort service is provided through the use of U (un-numbered) frames.

  - ✓ **The Asynchronous Balanced Mode (ABM)**. This provides a reliable data point-to-point data link service and may be used to provide a service which supports either a datagram or reliable network protocol. In this mode, the packets are carried in numbered I-frames, which are acknowledged by the receiver using numbered supervisory frames. Error recovery (e.g. checkpoint or go-back-n error recovery) is employed to ensure a well-ordered and reliable flow of frames.

Communicating with Internet Service Providers: Protocols

- **SLIP Protocol**

  - ✓ Serial Line IP (SLIP) was the first protocol for relaying IP packets over dial-up lines. It defines an encapsulation mechanism, but little else. There is no support for dynamic address assignment, link testing, or multiplexing different protocols over a single link. SLIP has been largely supplanted by PPP.

- **<u>PPP Protocol</u>**

  - ✓ The Point-to-Point Protocol (PPP) is currently the best solution for dial-up Internet connections, including ISDN.
  - ✓ PPP (Point-to-Point Protocol) is a protocol for communication between two computers using a serial interface, typically a personal computer connected by phone line to a server. For example, your Internet server provider may provide you with a PPP connection so that the provider's server can respond to your requests, pass them on to the Internet, and forward your requested Internet responses back to you.

- **<u>Difference between SLIP and PPP:</u>**

  - ✓ Although they can be used with different types of media, the most typical use is with telephone lines for an Internet connection; used to establish digital communication between the user and the ISP.

  - ✓ The main difference between SLIP and PPP is in their current use. SLIP is the older of the two and had a very minimal feature set. This eventually led to the creation of PPP and its more advanced features, thus rendering SLIP obsolete.

  - ✓ One of the key features in PPP is its ability to auto-configure its connection settings during initialization. The client and host communicate during initialization and negotiate on the best settings to be used. This is unlike SLIP which needs the settings coded beforehand in order to establish a successful connection. Auto-configuration significantly simplifies setup since most settings do not need to be entered manually.

  - ✓ Another essential feature added into PPP is error detection and recovery. In the process of transmitting data, it is very possible that a packet or two gets lost along the way. PPP is able to detect errors and automatically initiate the recovery of the lost packets. SLIP does not have any provisions for error detection so it needs to be implemented on a higher level. Not only does this add complexity, it also increases the processing needed.

  - ✓ Although SLIP is obsolete and is no longer used in most computer systems, it still enjoys some use in certain systems like microcontrollers. This is because of the relatively small amount of overhead that it adds.

- ✓ In order to transmit a packet, PPP adds a header as well as padding information in the end. In comparison, SLIP simply adds an end character at the end of each packet.

- **In summary:**

  - ✓ SLIP is obsolete and has been replaced by PPP in most applications.
  - ✓ PPP can auto-configure settings while SLIP cannot.
  - ✓ PPP provides error detection and recovery while SLIP doesn’t.
  - ✓ SLIP has very minimal overhead compared to PPP.

FRAMING

Need for Framing:

- Data communications is based on the exchange of data units (usually called frames), with a known structure (format)

- The problem of framing is solved in different ways depending on the frames having a fixed (known) length or a variable length.

  ✓ For frames of fixed length, it is only necessary to identify the start of the frame and add the frame size to locate the end of the frame.

  ✓ For frames of variable size, special synchronization characters or bit patterns are used to identify the start of a frame, while different explicit or implicit methods can be used for identifying the end of a frame.

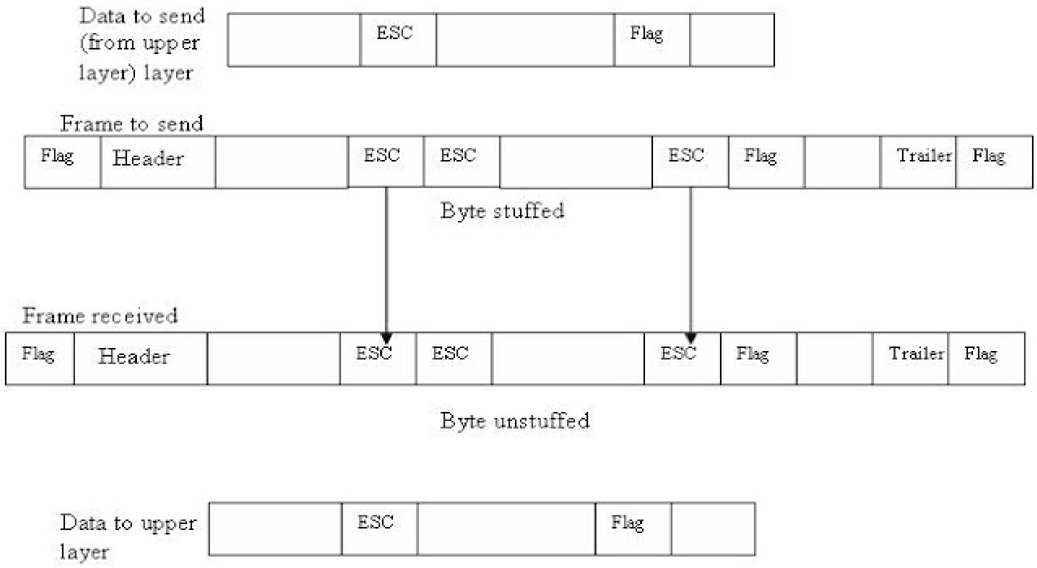
Frame Synchronization:

1. Character Count

This method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow, and hence where the end of the frame is. The disadvantage is that if the count is garbled by a transmission error, the destination will lose synchronization and will be unable to locate the start of the next frame. So, this method is rarely used.

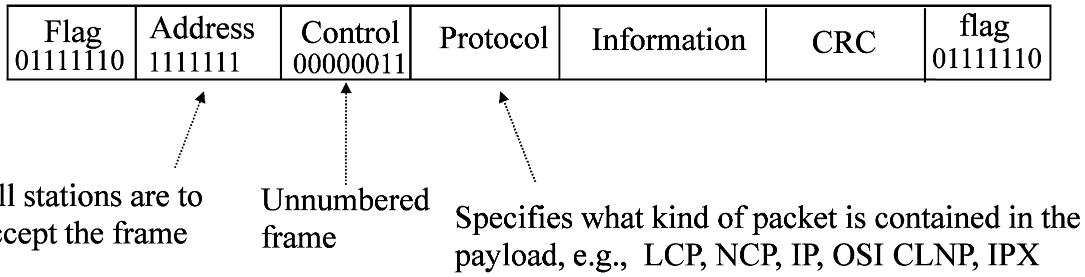
2. Byte Stuffing (Character Stuffing)

In byte stuffing a special byte is add to the data part (upon appearance of certain characters), this is known as **escape character** (ESC). The escape characters have a predefined pattern. The receiver removes the escape character and keeps the data part. It cause to another problem, if the text contains escape characters as part of data. To deal with this, an escape character is prefixed with another escape character. The following figure explains everything we discussed about character stuffing. However, character stuffing is closely associated with 8-bit characters and this is a major hurdle in transmitting arbitrary sized characters.



Note: PPP (point-to-point protocol) uses byte stuffing for frame synchronization.

PPP (Point-to-Point Protocol) Frame Format



Flag 01111110	Address 1111111	Control 00000011	Protocol	Information	CRC	flag 01111110

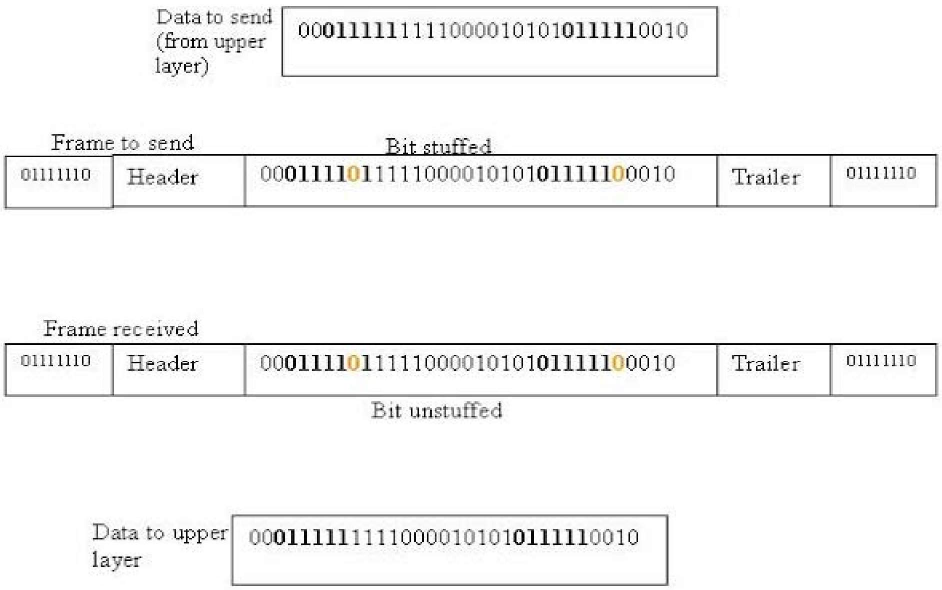
All stations are to accept the frame

Unnumbered frame

Specifies what kind of packet is contained in the payload, e.g., LCP, NCP, IP, OSI CLNP, IPX

3. Bit Stuffing

The third method allows data frames to contain an arbitrary number of bits and allows character codes with an arbitrary number of bits per character. At the start and end of each frame is a flag byte consisting of the special bit pattern 01111110. Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a zero bit into the outgoing bit stream. This technique is called bit stuffing. When the receiver sees five consecutive 1s in the incoming data stream, followed by a zero bit, it automatically destuffs the 0 bit. The boundary between two frames can be determined by locating the flag pattern.



Note: Bit stuffing is used in HDLC and its variants (except PPP).

- **Example#1:**

The following character encoding is used in a data link protocol:
A: 01000111; B: 11100011; FLAG: 01111110; ESC: 11100000
Show the bit sequence transmitted (in binary) for the four-character frame:
A B ESC FLAG when each of the following framing methods is used:
(a) Character count
(b) Flag bytes with byte stuffing.
(c) Starting and ending flag bytes, with bit stuffing.

- **Solution:**

a) 00000100 01000111 11100011 11100000 01111110
b) 01111110 01000111 11100011 11100000 11100000 11100000 01111110 01111110
c) 01111110 01000111 110100011 111000000 011111010 01111110

- **Example#2:**

The following character encoding is used in a data link protocol:
A: 11010101; B: 10101001; FLAG: 01111110; ESC: 10100011
Show the bit sequence transmitted (in binary) for the five-character frame:
A ESC B ESC FLAG when each of the following framing methods is used:
(a) Flag bytes with byte stuffing.
(b) Starting and ending flag bytes, with bit stuffing.

- **Solution:**

a) 01111110 11010101 10100011 10100011 10101001 10100011 10100011 10100011 01111110 01111110
b) 01111110 11010101 10100011 10101001 10100011 011111010 01111110

- **Example#3:**

Given the output after byte-stuffing: FLAG A B ESC ESC C ESC ESC ESC FLAG ESC FLAG D F FLAG. What is the original data?

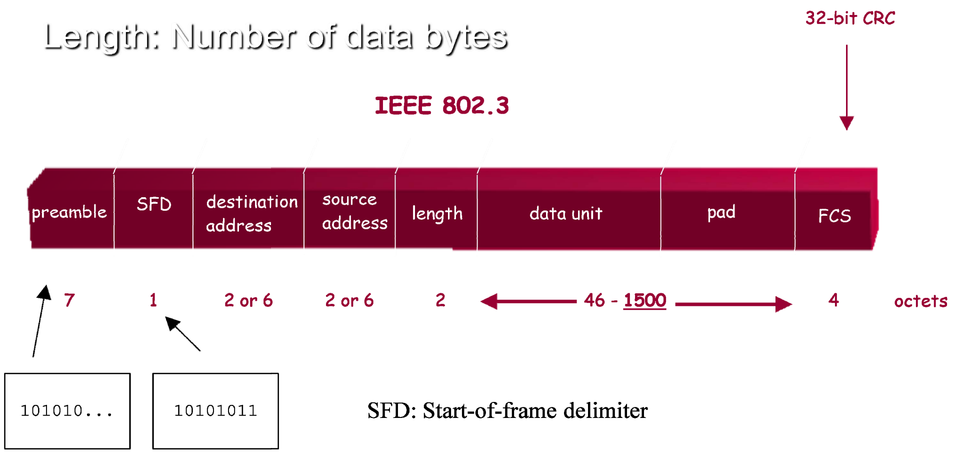
- **Solution:**

A B ESC C ESC FLAG FLAG D F

4. Physical Layer Coding Violations (using redundancy in physical layer encoding, used in IEEE 802.3)

- Framing in Ethernet / IEEE 802.3:
  - ➢ In Ethernet, biphase (or Manchester) coding is used for transmission (i.e., synchronous transmission):
    - ✓ A 0 and a 1 are coded with two elementary pulses with different levels, with a transition in the middle
  - ➢ The protocol used in the original Ethernet is based on listening to the activity on a shared medium (*carrier sense*)
    - ✓ A station detects that there is no transmission in course when no carrier is present (no transitions detected)
  - ➢ When a station acquires the medium it starts to transmit a frame and at the end of the frame it simply ceases to transmit coded bits.
  - ➢ At a receiving station the start of a frame is identified by a known pattern that consists in:
    - ✓ A *Preamble* that is made up of seven bytes with the pattern 10101010 and is used for bit synchronization
    - ✓ A *Start of Frame Delimiter* (SFD) with the pattern10101011 that precedes the remaining fields of the frame
  - ➢ At a receiving station, the end of a frame is detected by the absence of transitions in the coded signal (no carrier present)

IEEE802.3 Frame Format



SFD: Start-of-frame delimiter

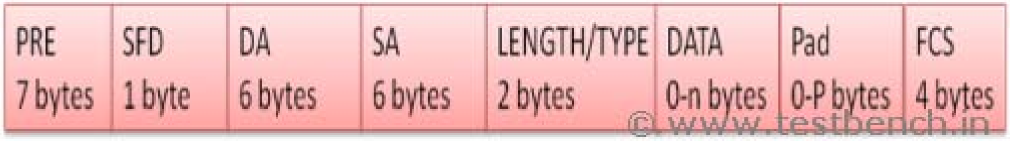
used to synchronize receiver, sender clocks

Notes:

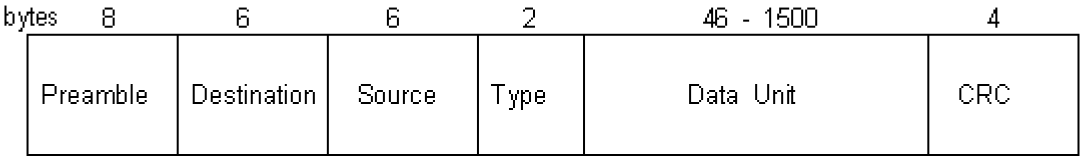
- There are also framing in *IEEE 802. 5 Token Ring* and framing in *FDDI Token Ring*

- **Further Information about IEEE 802.3 Frame Format:**

  - **MAC FRAME FORMAT** (Notice that there is no LLC over here)

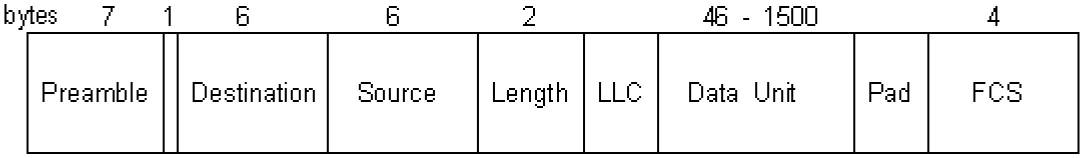


PRE	SFD	DA	SA	LENGTH/TYPE	DATA	Pad	FCS
7 bytes	1 byte	6 bytes	6 bytes	2 bytes	0-n bytes	0-P bytes	4 bytes



bytes 8	6	6	2	46 - 1500	4
Preamble	Destination	Source	Type	Data Unit	CRC

DIX Ethernet Packet



bytes 7	1	6	6	2	46 - 1500			4
Preamble		Destination	Source	Length	LLC	Data Unit	Pad	FCS

IEEE 802.3 Frame

- **Necessary discussions:**

PREAMBLE:

1. Need for Synchronization: In LAN implementation, most of the physical layer components are allowed to provide valid output only after some number of bit times prior to the valid input signals. So this condition necessitates using a Preamble which is to be sent before the start of the data. This allows the **Physical Layer Signaling** (PLS) circuitry (of the receiver) to reach its steady state with the received frame's timing. So Preamble is used for physical medium stabilization and synchronization followed by SFD.

2. Preamble is not used by the MAC layer, so the minimum amount of preamble required for a device to function properly depends up on which physical layer is implemented and not up on the MAC layer.

3. The preamble bits are transmitted in order from left to right and it should be noted that PRE ends with a '0' (alternating 1's and 0's).

4. IEEE Standard does not define the minimum PRE size, PRE size is handled depending up on the PHY as it is the function of physical medium. So min PRE size is considered as 1byte. Even though Standard defines as PRE as 7bytes, Mac should tolerate large amounts of Preamble.

5. The PRE is maintained in the fast Ethernet and gigabit systems to provide compatibility with the original Ethernet frame.

START FRAME DELIMITER [SFD]:

1. SFD field is the sequence 10101011 which immediately follows the PRE pattern and indicates the start of the frame.

2. The last two bits indicates the receiving interface that the end of the preamble and SFD has been reached and that the bits that follow are actual fields of the frame

3. Any successive bits following the transmission of SFD are recognized as data bits and are passed on to the LLC sub layer.

Computer Networks

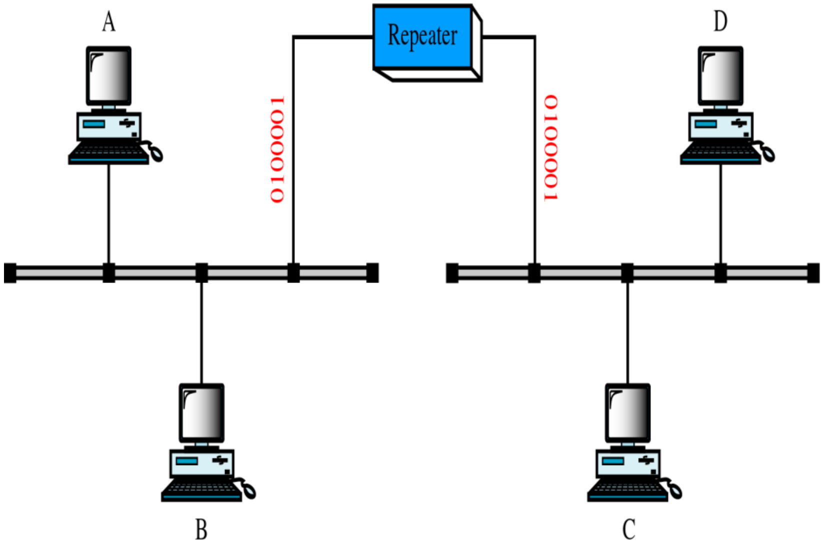
Internetworking Devices

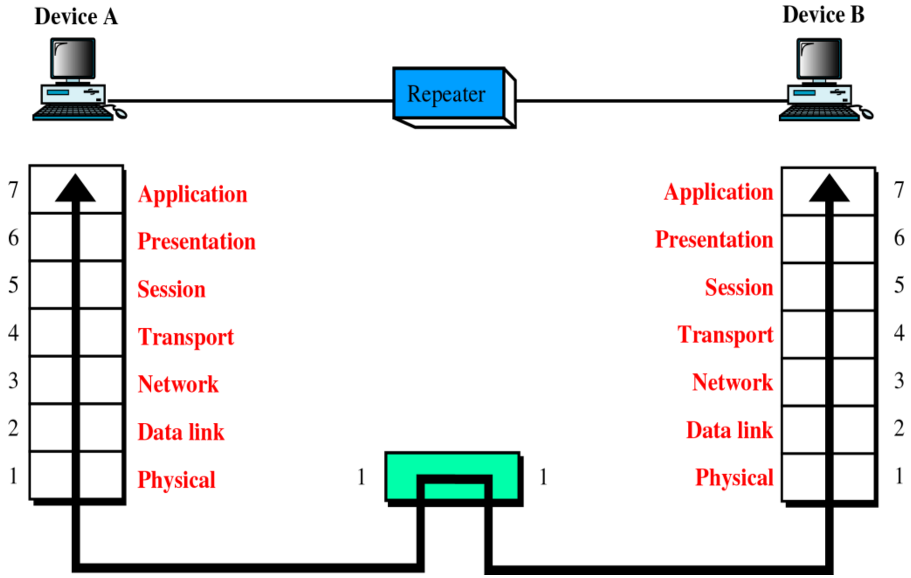
Internetworking Devices

- Repeaters
- Hubs
- Bridges
  - Learning algorithms
  - Problem of closed loops
- Switches
- Routers

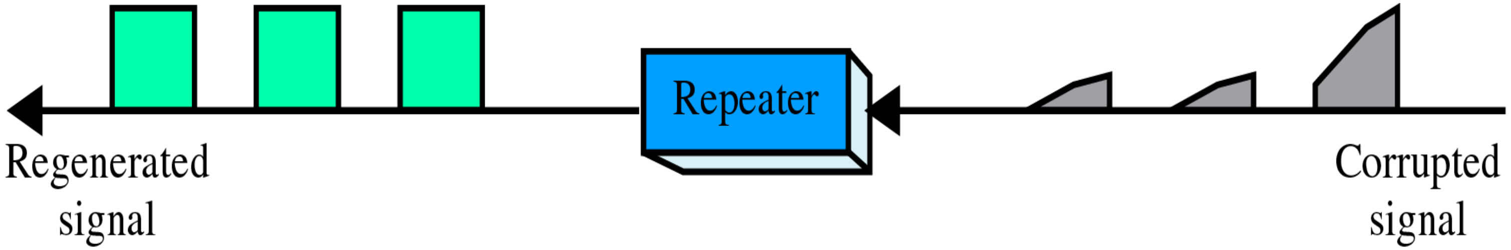
Repeaters

- Repeaters are purely physical layer devices
- Single collision domain

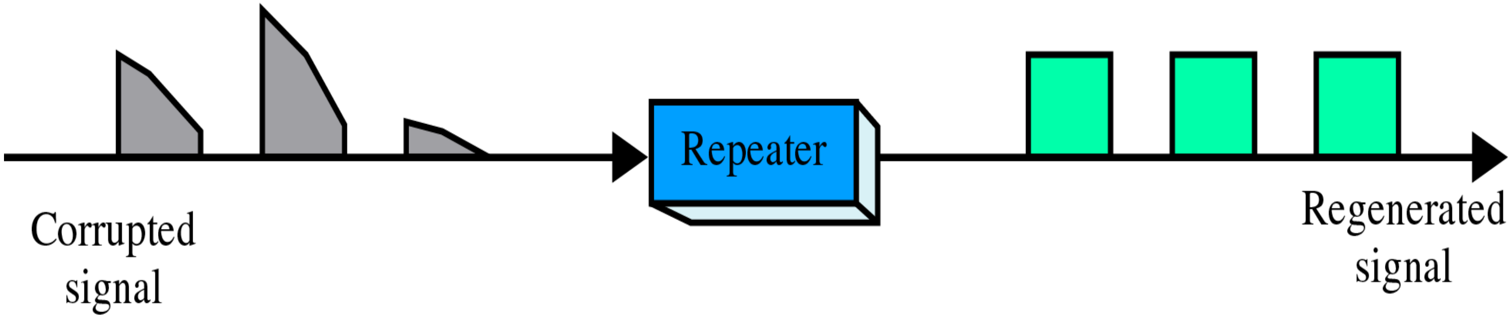




Functions of a Repeater



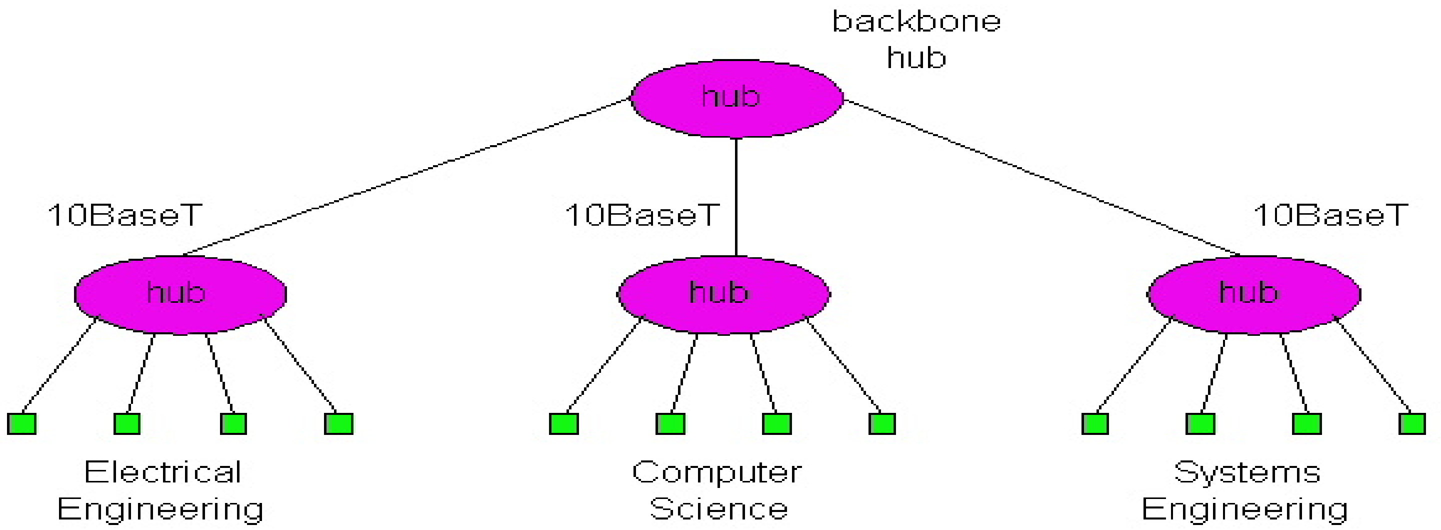
(a) Right-to-left transmission.



(b) Left-to-right transmission.

Shared Hubs

- Physical Layer devices: essentially repeaters operating at bit levels: repeat received bits on one interface to all other interfaces
- Hubs can be arranged in a *hierarchy* with backbone hub at its top



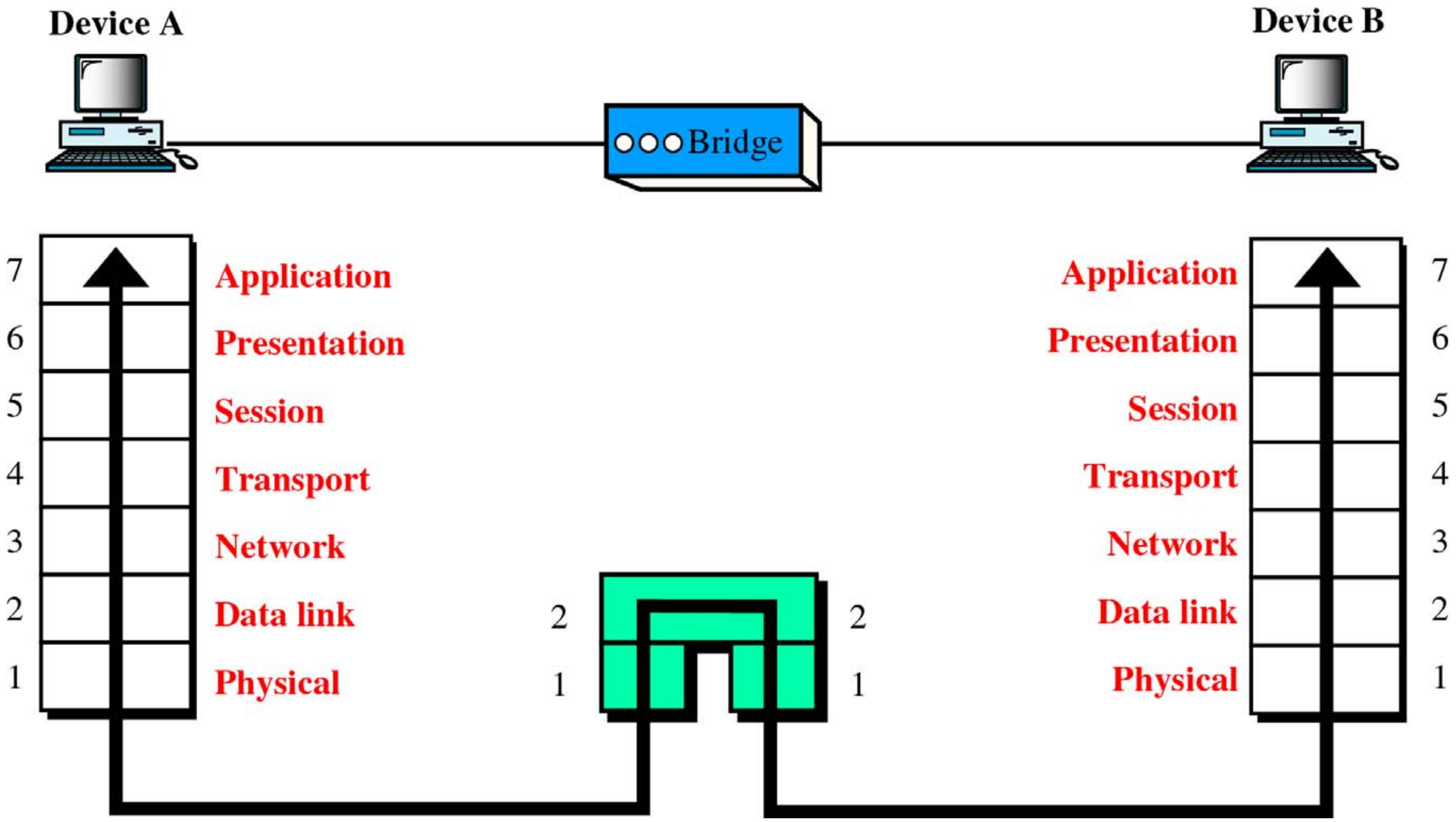
Shared Hubs Limitations

- Single collision domain results in no increase in maximum throughput
  - multi-tier throughput same as single segment throughput
- Cannot connect different Ethernet types (e.g., 10BaseT and 100baseT)

Bridges

- Bridges are MAC/Link layer devices operating on Ethernet frames, examining frame header and selectively forwarding frame based on its destination
- Bridge ***isolates collision*** domains since it buffers frames
- When frame is to be forwarded on segment, bridge uses CSMA/CD to access segment and transmit

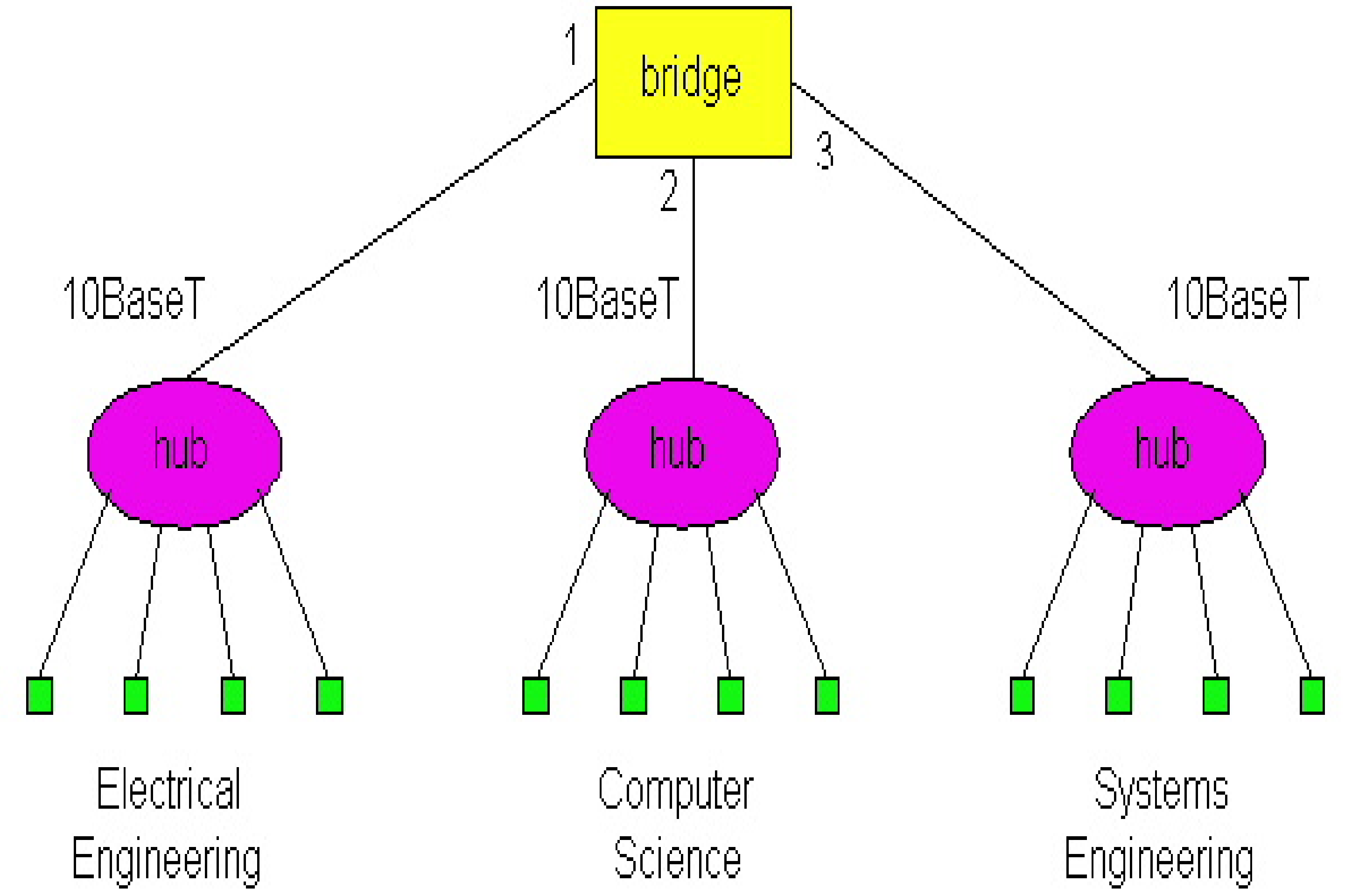
A Bridge in the OSI model



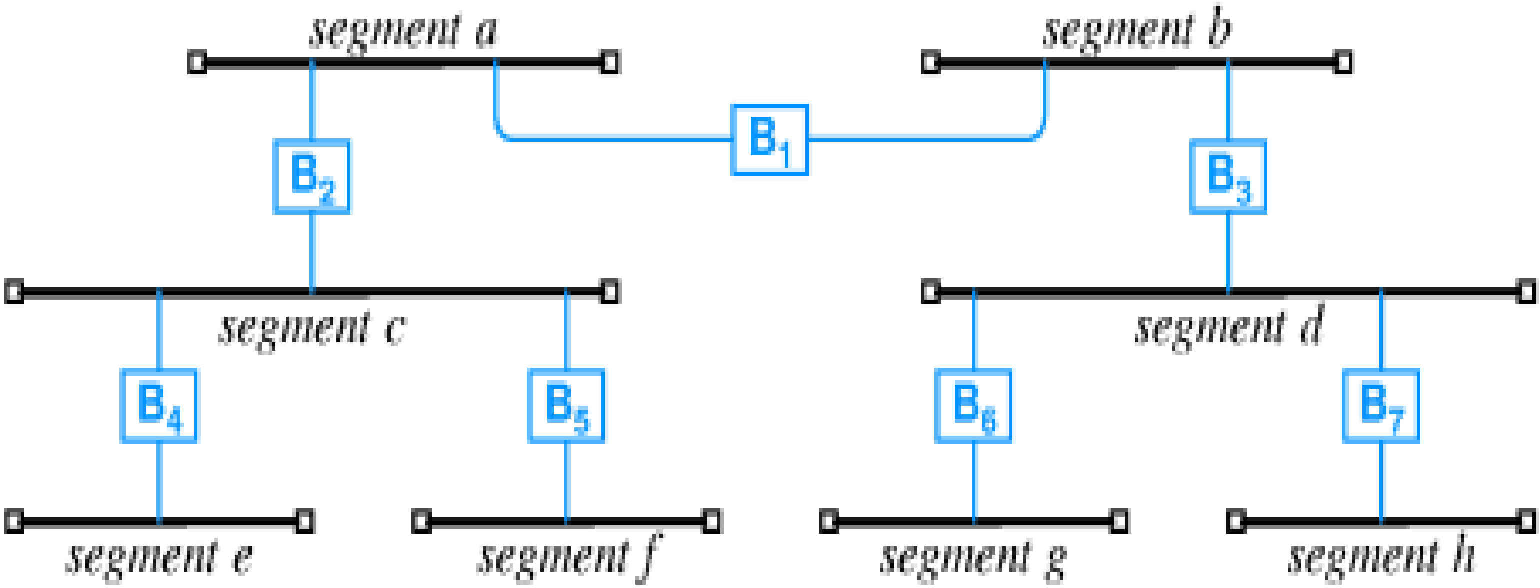
LAN Bridges

- Isolates collision domains resulting in higher total maximum throughput, and does not limit the number of nodes nor geographical coverage
- Can connect different type Ethernet since it is a store and forward device
- Transparent: no need for any change to hosts LAN adapters. Hosts do not communicate with bridges

Bridged LAN Configuration



Bridged LAN with Multiple Segments



Bridge Modes of Operation

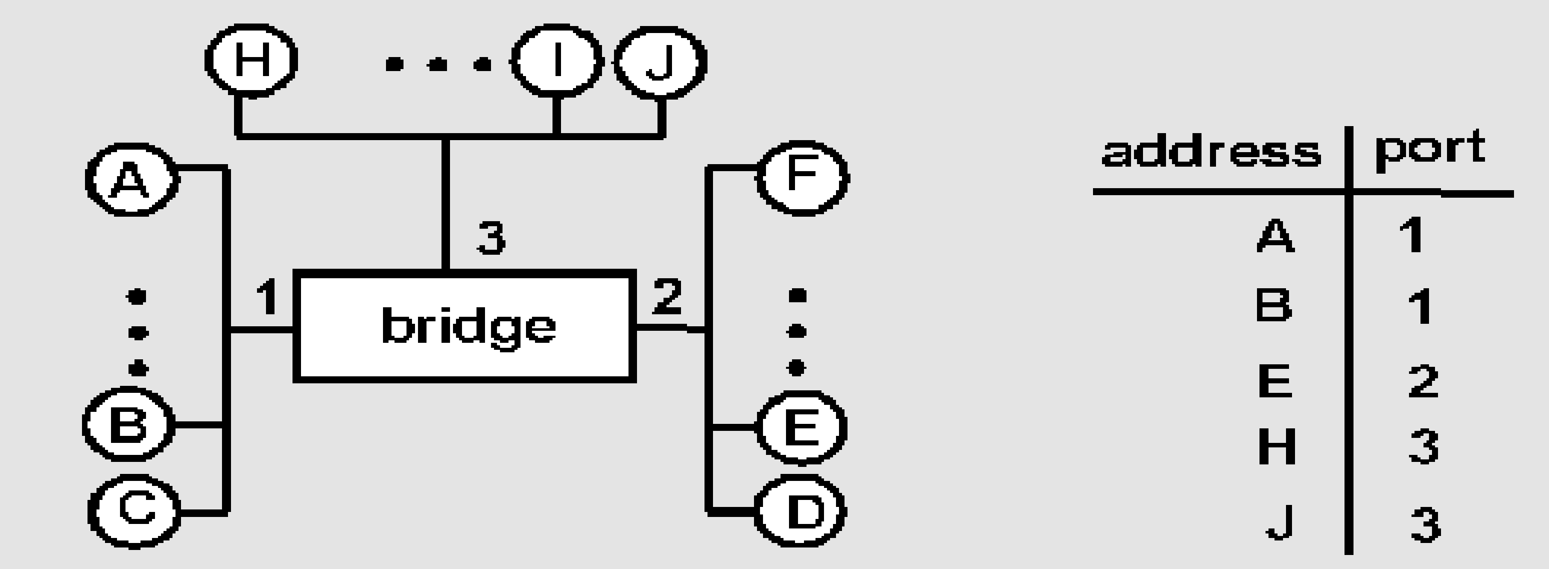
- ***Filtering***: Bridges filter frames if source and destination hosts are on the same segment! Other segments will not get such frames
- ***Forwarding:*** Bridges forward frames if source and destination hosts are on different segments ***and*** the bridge knows on which segment is the destination host connected to
- ***Flooding:*** Bridges flood frames to all interfaces (except the one it received the frame from) if it doesn’t know where the destination host is

Learning Bridges

- Bridges ***learn*** which hosts can be reached through which interfaces by maintain filtering tables
  - When a frame received, bridge “learns” location of sender: incoming LAN segment
  - Records sender location in filtering table
- Filtering table entries
  - Host MAC address, Bridge interface, Time stamp
  - Stale entries in filtering table dropped.

Example of Learning Bridges

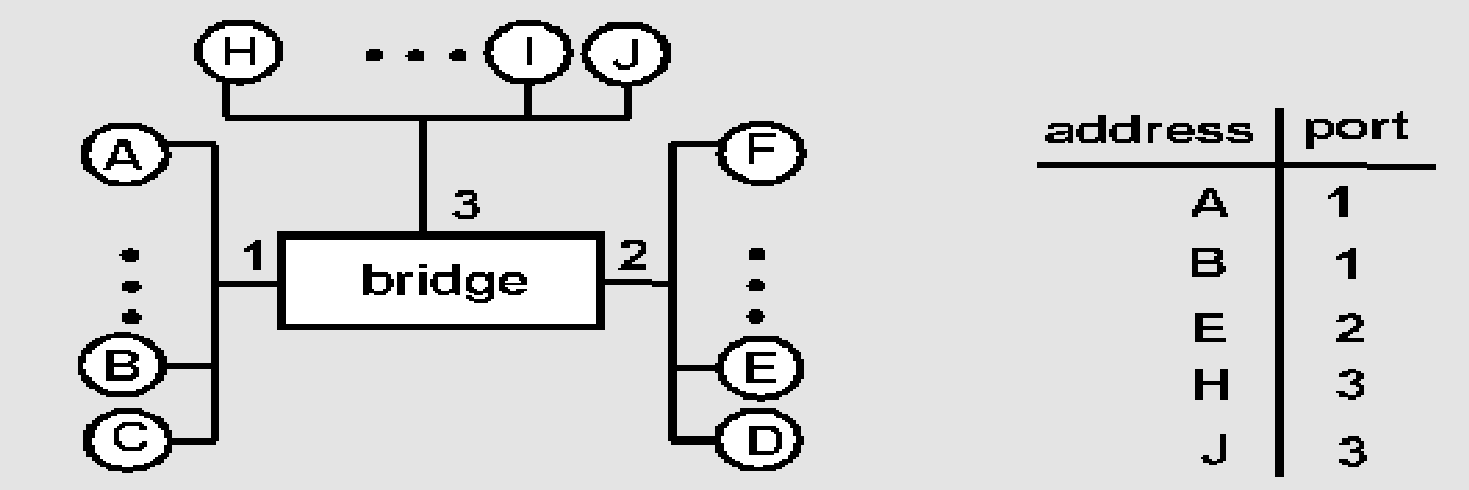
- Suppose C sends frame to D and D replies back with frame to C



address	port
A	1
B	1
E	2
H	3
J	3

- C sends frame, bridge has no info about D, so ***floods*** to both interfaces 2 & 3
  - – Bridge learns C is on port 1, add it to its table

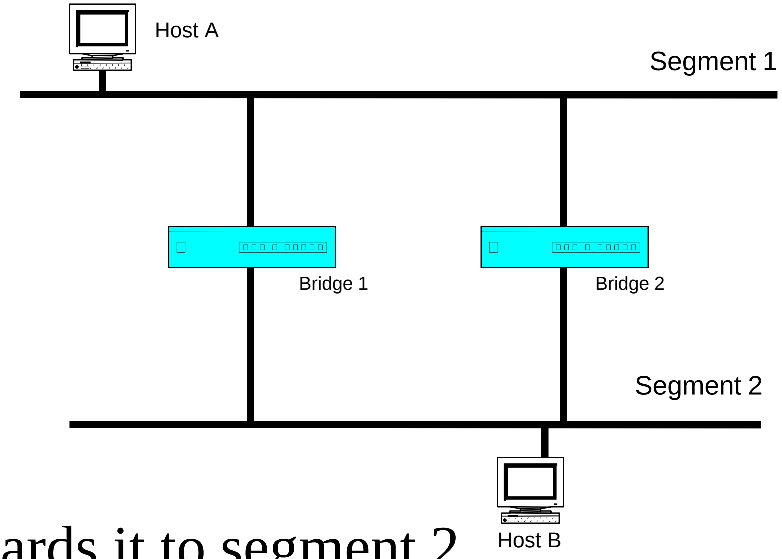
Example (Continued)



address	port
A	1
B	1
E	2
H	3
J	3

- D generates reply to C and sends it
  - – Bridge sees frame from D
  - – Bridge learns D is on interface 2, add to table
  - – Bridge knows C on interface 1, so it forwards frame out via interface 1 and filter it from interface 3

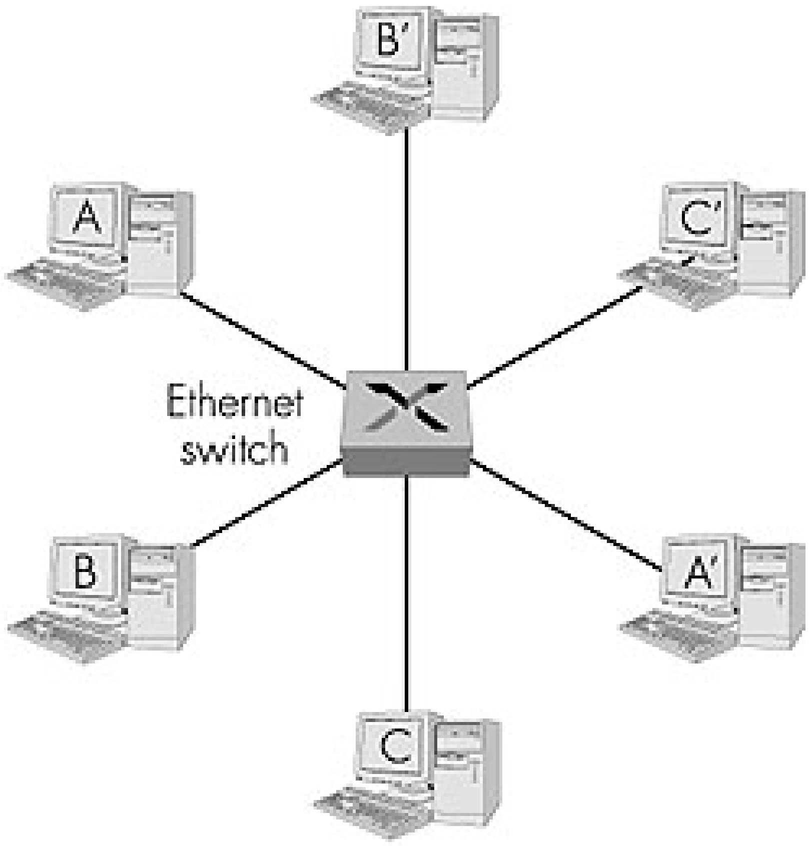
Closed Loops



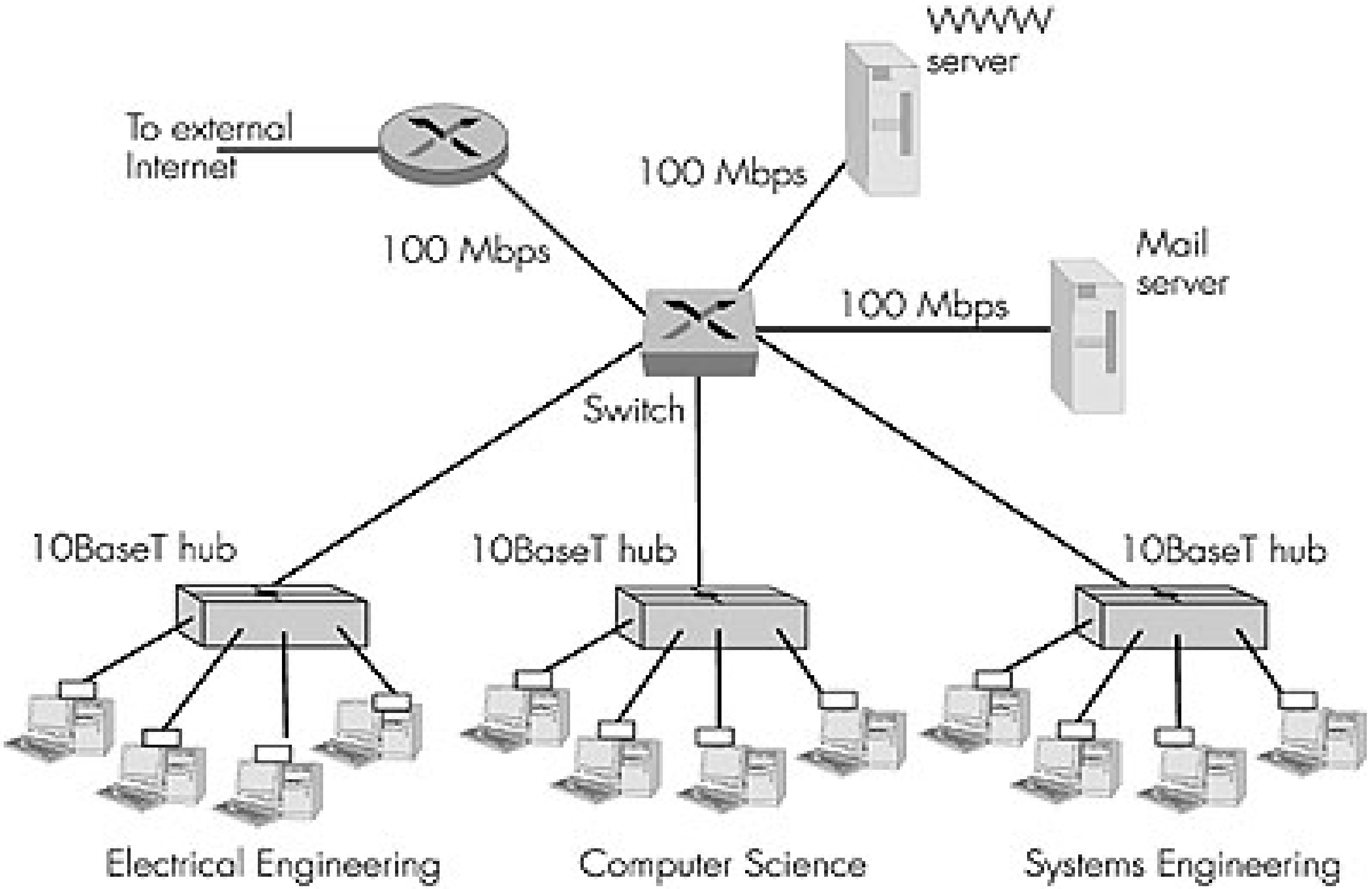
- Host A sends a frame to Host B
- Bridge 1 receives the frame
- Not knowing where host B is it forwards it to segment 2
- Frame goes to its destination B, but at the same time is picked up by Bridge 2
- Bridge 2 – Erroneously sees a frame on Segment 2 from Host A so he updates his tables to include Host A in Segment 2
- Because it does not know about Host B it forwards the frame to Segment 1
- The frame is then received by Bridge 1 again – and the cycle will repeat itself endlessly
- Solution: Spanning Tree Algorithm (insure that there would be ***one and only one path*** between any two hosts)

Ethernet “Layer 2” Switching

- layer 2 (frame) forwarding/ filtering/flooding based on MAC addresses
- Switching: A-to-B and A’-to-B’ simultaneously, no collisions
- Ethernet but no collisions
- *Store & Forward* v.s *Cut-through Switching*

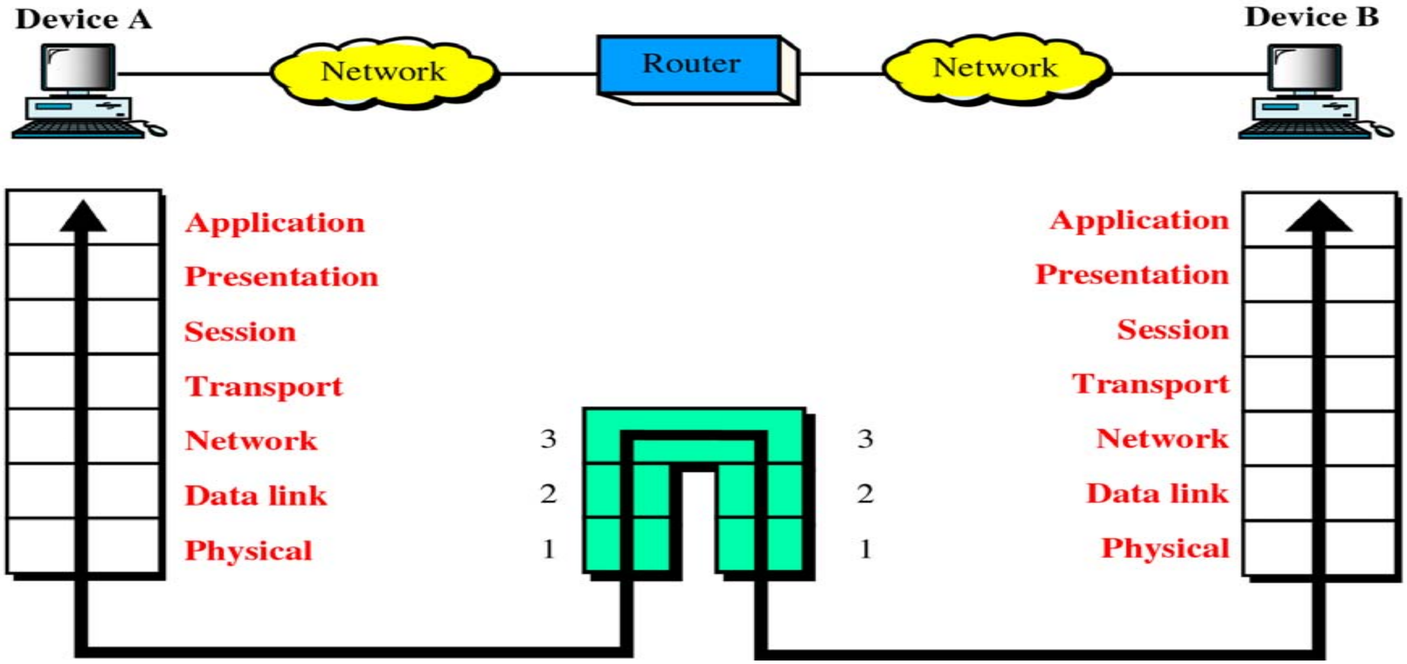


Example of Ethernet Switching



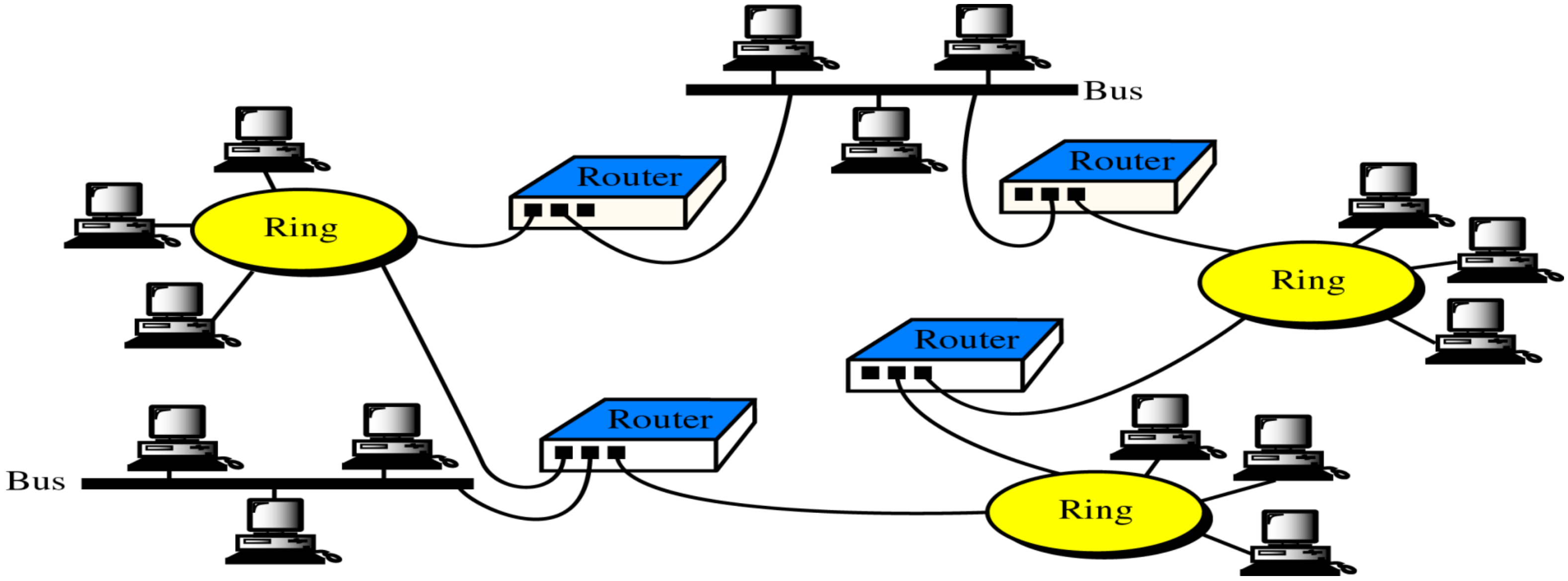
Routers

- Routers are ***network-layer*** devices
- Routers implement ***routing algorithms*** and maintain ***routing tables***



Example of an Internetwork

- Routers are used to interconnect ***arbitrary topologies***



Gateway

- A traditional "gateway" can mean a device that (sometimes) is able to route traffic, but whose primary goal is to *translate* from one protocol to another.
- For example- I would use a "gateway" if I wanted to send packets from my IPv4 network to/from a IPv6 network, or maybe an AppleTalk network to/from a Token Ring network.
- The primary goal of a "router" is to intelligently forward traffic from one interface to another... not necessarily to *translate* from one protocol to another.

Gateway (Cont.)

Summary:

- **What network device connects multiple networks that use the same protocols?**
  - The network device that connect multiple networks that use the same protocol is a ROUTER.
- **Which Device is used to connect dissimilar networks?**
  - Gateway

Ethernet (IEEE 802.3)

- Ethernet is a family of computer networking technologies for local area (LAN) and larger networks.
- It was commercially introduced in 1980 while it was first standardized in 1983 as IEEE 802.3, and has since been refined to support higher bit rates and longer link distances.
- Over time, Ethernet has largely replaced competing wired LAN technologies such as token ring, FDDI, and ARCNET.
- The Ethernet standards comprise several wiring and signaling variants of the OSI physical layer in use with Ethernet.
- The original 10BASE5 Ethernet used coaxial cable as a shared medium.
- Later the coaxial cables were replaced with twisted pair and fiber optic links in conjunction with hubs or switches.

  - ✓ Various standard defined for IEEE802.3 (Old Ethernet)
    - ▪ 10Base5 -- thickwire coaxial
    - ▪ 10Base2 -- thinwire coaxial or cheapernet
    - ▪ 10BaseT -- twisted pair
    - ▪ 10BaseF -- fiber optics
  - ✓ Fast Ethernet
    - ▪ 100BaseTX and 100BaseF

- Old Ethernet: CSMA/CD, Shared Media, and Half Duplex Links
- Fast Ethernet: No CSMA/CD, Dedicated Media, and Full Duplex Links

- Data rates have been incrementally increased from the original 10 megabits per second to 100 gigabits per second over its history.
- Systems communicating over Ethernet divide a stream of data into shorter pieces called frames. Each frame contains source and destination addresses and error-checking data so that damaged data can be detected and re-transmitted.
- As per the OSI model, Ethernet provides services up to and including the data link layer.
- Evolution:

  - ✓ Shared media
  - ✓ Repeaters and hubs
  - ✓ Bridging and switching
  - ✓ Advanced networking

- Varieties of Ethernet: Ethernet physical layer:

  - ✓ The Ethernet physical layer is the physical layer component of the Ethernet family of computer network standards.

- ✓ The Ethernet physical layer evolved over a considerable time span and encompasses coaxial, twisted pair and fiber optic physical media interfaces and speeds from 10 Mbit to 100 Gbit.
- ✓ The most common forms used are 10BASE-T, 100BASE-TX, and 1000BASE-T. They run at 10 Mbit/s, 100 Mbit/s, and 1 Gbit/s, respectively.
- ✓ Fiber optic variants of Ethernet offer high performance, electrical isolation and distance (tens of kilometers with some versions).

- Ethernet over twisted pair:
  - ✓ Ethernet over twisted pair technologies use twisted-pair cables for the physical layer of an Ethernet computer network.
  - ✓ Early Ethernet cabling had generally been based on various grades of coaxial cable, but in 1984, StarLAN showed the potential of simple *unshielded* twisted pair by using Cat3 cable—the same simple cable used for telephone systems. This led to the development of 10BASE-T and its successors 100BASE-TX and 1000BASE-T, supporting speeds of 10, 100 and 1000 Mbit/s respectively.

- **Communication Standards:**

Ethernet Standard	**Date**	**Description**
Experimental Ethernet	1973	2.94 Mbit/s (367 kB/s) over coaxial cable (coax) bus
Ethernet II (DIX v2.0)	1982	10 Mbit/s (1.25 MB/s) over thick coax. Frames have a Type field. This frame format is used on all forms of Ethernet by protocols in the Internet protocol suite.
IEEE 802.3 standard	1983	10BASE5 10 Mbit/s (1.25 MB/s) over thick coax. Same as Ethernet II (above) except Type field is replaced by Length, and an 802.2 LLC header follows the 802.3 header. Based on the CSMA/CD Process.
802.3a	1985	10BASE2 10 Mbit/s (1.25 MB/s) over thin Coax (a.k.a. thinnet or cheapernet)
802.3b	1985	10BROAD36
802.3c	1985	10 Mbit/s (1.25 MB/s) repeater specs
802.3d	1987	Fiber-optic inter-repeater link
802.3e	1987	1BASE5 or StarLAN
802.3i	1990	10BASE-T 10 Mbit/s (1.25 MB/s) over twisted pair
802.3j	1993	10BASE-F 10 Mbit/s (1.25 MB/s) over Fiber-Optic
802.3u	1995	100BASE-TX, 100BASE-T4, 100BASE-FX Fast Ethernet at 100 Mbit/s (12.5 MB/s) w/autonegotiation
802.3x	1997	Full Duplex and flow control; also incorporates DIX framing, so there's no longer a DIX/802.3 split
802.3y	1998	100BASE-T2 100 Mbit/s (12.5 MB/s) over low quality twisted pair
802.3z	1998	1000BASE-X Gbit/s Ethernet over Fiber-Optic at 1 Gbit/s (125 MB/s)
802.3-1998	1998	A revision of base standard incorporating the above amendments and errata
802.3ab	1999	1000BASE-T Gbit/s Ethernet over twisted pair at 1 Gbit/s (125 MB/s)
802.3ac	1998	Max frame size extended to 1522 bytes (to allow "Q-tag") The Q-tag includes

		802.1Q VLAN information and 802.1p priority information.
802.3ad	2000	Link aggregation for parallel links, since moved to IEEE 802.1AX
802.3-2002	2002	A revision of base standard incorporating the three prior amendments and errata
802.3ae	2002	10-gigabit Ethernet over fiber; 10GBASE-SR, 10GBASE-LR, 10GBASE-ER, 10GBASE-SW, 10GBASE-LW, 10GBASE-EW
802.3af	2003	Power over Ethernet (15.4 W)
802.3ah	2004	Ethernet in the First Mile
802.3ak	2004	10GBASE-CX4 10 Gbit/s (1,250 MB/s) Ethernet over twinaxial cables
802.3-2005	2005	A revision of base standard incorporating the four prior amendments and errata.
802.3an	2006	10GBASE-T 10 Gbit/s (1,250 MB/s) Ethernet over unshielded twisted pair (UTP)
802.3ap	2007	Backplane Ethernet (1 and 10 Gbit/s (125 and 1,250 MB/s) over printed circuit boards)
802.3aq	2006	10GBASE-LRM 10 Gbit/s (1,250 MB/s) Ethernet over multimode fiber
P802.3ar	Cancelled	Congestion management (withdrawn)
802.3as	2006	Frame expansion
802.3at	2009	Power over Ethernet enhancements (25.5 W)
802.3au	2006	Isolation requirements for Power over Ethernet (802.3-2005/Cor 1)
802.3av	2009	10 Gbit/s EPON
802.3aw	2007	Fixed an equation in the publication of 10GBASE-T (released as 802.3-2005/Cor 2)
802.3-2008	2008	A revision of base standard incorporating the 802.3an/ap/aq/as amendments, two corrigenda and errata. Link aggregation was moved to 802.1AX.
802.3az	2010	Energy Efficient Ethernet
802.3ba	2010	40 Gbit/s and 100 Gbit/s Ethernet. 40 Gbit/s over 1m backplane, 10 m Cu cable assembly (4x25 Gbit or 10x10 Gbit lanes) and 100 m of MMF and 100 Gbit/s up to 10 m of Cu cable assembly, 100 m of MMF or 40 km of SMF respectively
802.3-2008/Cor 1	2009	Increase Pause Reaction Delay timings which are insufficient for 10 Gbit/s (workgroup name was 802.3bb)
802.3bc	2009	Move and update Ethernet related TLVs (type, length, values), previously specified in Annex F of IEEE 802.1AB (LLDP) to 802.3.
802.3bd	2010	Priority-based Flow Control. An amendment by the IEEE 802.1 Data Center Bridging Task Group (802.1Qbb) to develop an amendment to IEEE Std 802.3 to add a MAC Control Frame to support IEEE 802.1Qbb Priority-based Flow Control.
802.3.1	2011	MIB definitions for Ethernet. It consolidates the Ethernet related MIBs present in Annex 30A&B, various IETF RFCs, and 802.1AB annex F into one master document with a machine readable extract. (workgroup name was P802.3be)
802.3bf	2011	Provide an accurate indication of the transmission and reception initiation times of certain packets as required to support IEEE P802.1AS.
802.3bg	2011	Provide a 40 Gbit/s PMD which is optically compatible with existing carrier SMF 40 Gbit/s client interfaces (OTU3/STM-256/OC-768/40G POS).

802.3-2012	2012	A revision of base standard incorporating the 802.3at/av/az/ba/bc/bd/bf/bg amendments, a corrigenda and errata.
802.3bj	~Mar 2014	Define a 4-lane 100 Gbit/s backplane PHY for operation over links consistent with copper traces on “improved FR-4” (as defined by IEEE P802.3ap or better materials to be defined by the Task Force) with lengths up to at least 1m and a 4-lane 100 Gbit/s PHY for operation over links consistent with copper twinaxial cables with lengths up to at least 5m.
802.3bk	2013	This amendment to IEEE Std 802.3 defines the physical layer specifications and management parameters for EPON operation on point-to-multipoint passive optical networks supporting extended power budget classes of PX30, PX40, PRX40, and PR40 PMDs.
802.3bm	2014	100G/40G Ethernet for optical fiber
802.3bp	2014	1000BASE-T1 - Gigabit Ethernet over a single twisted pair, automotive & industrial environments
802.3bq	~Feb 2016	40GBASE-T for 4-pair balanced twisted-pair cabling with 2 connectors over 30 m distances
802.3bs	~ 2017	400 Gb/s Ethernet over optical fiber using multiple 25G/50G lanes
802.3bw		100BASE-T1 - 100 Mbit/s Ethernet over a single twisted pair for automotive applications
802.3-2015	2015	802.3bx - a new consolidated revision of the 802.3 standard including amendments 802.2bk/bj/bm

- **Ethernet Media Standards and Distances: Details**

✓ There are single-mode optical fiber (SMF), laser optimized multi-mode optical fiber (MMF)

✓ Multi-mode fibers are described using a system of classification determined by the ISO 11801 standard — OM1, OM2, and OM3. OM4 (defined in TIA-492-AAAD) was finalized in August 2009, and was published by the end of 2009 by the Telecommunications Industry Association (TIA). The letters "OM" stand for *optical multi-mode*.

✓ The main difference between multi-mode and single-mode optical fiber is that the former has much larger core diameter, typically 50–100 micrometers.

✓ Multi-mode fibers are described by their core and cladding diameters. Thus, 62.5/125 µm multi-mode fiber has a core size of 62.5 micrometres (μm) and a cladding diameter of 125 µm.

✓ 62.5/125 µm (OM1) and conventional 50/125 µm multi-mode fiber (OM2) were widely deployed in premises applications. These fibers easily support applications ranging from Ethernet (10 Mbit/s) to gigabit Ethernet (1 Gbit/s).

✓ Optical fiber manufacturers have greatly refined their manufacturing process to offer higher bandwidths (data rates). Hence, newer deployments often use laser-optimized 50/125 µm multi-mode fiber (OM3) and 50/125 µm multi-mode fiber (OM4, higher bandwidth). These fibers easily support applications ranging from 10 gigabit Ethernet to 100 gigabit Ethernet .

- ✓ **100-gigabit Ethernet** (**100GbE**) and **40-gigabit Ethernet** (**40GbE**) are groups of computer networking technologies for transmitting Ethernet frames at rates of 100 and 40 gigabits per second (100 to 40 Gbit/s), respectively. The technology was first defined by the IEEE 802.3ba-2010 standard.
- ✓ The industry has widely adopted the terms short reach, long reach and extended reach to match 10GBASE-SR, 10GBASE-LR, and 10GBASE-ER respectively. From an IEEE standards perspective, the “S” represents the 850 nm Short wavelength, the “L” stands for the 1310 nm Long wavelength, and the “E” stands for the 1550 nm Extra long wavelength.

100G Port Types:

- ❖ **100GBASE-CR10:** 100GBASE-CR10 ("copper", 10m) is a port type for twin-ax copper cable. It uses ten lanes of twin-ax cable delivering serialized data at a rate of 10.3125 Gbit/s per lane.
- ❖ **100GBASE-CR4:** 100GBASE-CR4 ("copper", 10m) is a port type for twin-ax copper cable. It uses four lanes of twin-ax cable delivering serialized data at a rate of 25.78125 Gbit/s per lane.
- ❖ **100GBASE-SR10:** 100GBASE-SR10 ("short range", OM3 MMF cable: 100m, OM4 MMF cable: 150m) is a port type for multi-mode fiber and uses 850 nm lasers. It uses ten lanes of multi-mode fiber delivering serialized data at a rate of 10.3125 Gbit/s per lane.
- ❖ **100GBASE-SR4:** 100GBASE-SR4 ("short range", OM3 MMF cable: 100m, OM4 MMF cable: 150m) is a port type for multi-mode fiber and uses 850 nm lasers. It uses four lanes of multi-mode fiber delivering serialized data at a rate of 25.78125 Gbit/s per lane.
- ❖ **100GBASE-LR4:** 100GBASE-LR4 ("long range", 10 km) is a port type for single-mode fiber and uses four lasers using four wavelengths around 1310 nm. Each wavelength carries data at a rate of 25.78125 Gbit/s.
- ❖ **100GBASE-ER4:** 100GBASE-ER4 ("extended range", 40km) is a port type for single-mode fiber and uses four lasers using four wavelengths around 1550 nm. Each wavelength carries data at a rate of 25.78125 Gbit/s.

40G Port Types

- ❖ **40GBASE-CR4:** 40GBASE-CR4 ("copper", 10m) is a port type for twin-ax copper cable. It uses four lanes of twin-ax cable delivering serialized data at a rate of 10.3125 Gbit/s per lane.
- ❖ **40GBASE-SR4:** 40GBASE-SR4 ("short range", OM3 MMF cable: 100m, OM4 MMF cable: 150m) is a port type for multi-mode fiber and uses 850 nm lasers. It uses four lanes of multi-mode fiber delivering serialized data at a rate of 10.3125 Gbit/s per lane.

- **40GBASE-LR4:** 40GBASE-LR4 ("long range", 10 km) is a port type for single-mode fiber and uses 1310 nm lasers. It uses four wavelengths delivering serialized data at a rate of 10.3125 Gbit/s per wavelength.
- **40GBASE-ER4:** 40GBASE-ER4 ("extended range", 40km) is a port type for single-mode fiber being defined in P802.3bm and uses 1550 nm lasers. It uses four wavelengths delivering serialized data at a rate of 10.3125 Gbit/s per wavelength.
- **40GBASE-FR:** 40GBASE-FR is a port type for single-mode fiber (10km). It uses 1550 nm optics, has a reach of 10 km and is capable of receiving 1550 nm and 1310 nm wavelengths of light. 1550 nm was chosen as the wavelength transmission to make it compatible with existing test equipment and infrastructure.
- **40GBASE-T**: 40GBASE-T (100m) is a port type for 4-pair balanced twisted-pair Cat.8 copper cabling being defined in P802.3bq.

Summary:

- **10/100/1000 on Copper**

Standard	**Media**	**distance**
10BASE-T	Cat-3 (2-pair)	100m
100BASE-T	Cat-5 (2-pair)	100m
1000BASE-T	Cat-5 (4-pair)	100m

- **100 Mbit Fiber**

Standard	**Media**	**distance**	**wavelength**
100BASE-FX	62.5µm MMF	2km	1300nm

- **Gigabit Fiber**

Standard	**Media**	**distance**	**wavelength**
1000BASE-SX	62.5µm MMF	220m	850nm
1000BASE-SX	50µm MMF	500m	850nm
1000BASE-LX	62.5µm MMF	550m	1310nm
1000BASE-LX	50µm MMF	550m	1310nm
1000BASE-LX/LH	SMF	10km	1310nm

1000BASE-ZX	SMF	70km	1550nm

- **10 Gigabit Fiber**

Standard	**Media**	**distance**	**wavelength**
10GBASE-SR	62.5µm MMF	26m-82m	850nm
10GBASE-LRM	62.5µm MMF	220m	1310nm
10GBASE-LX4	62.5µm MMF	300m	1300nm
10GBASE-SR	50µm OM3	300m	850nm
10GBASE-LRM	50µm OM3	260m	1310nm
10GBASE-LX4	50µm OM3	300m	1300nm
10GBASE-LR	SMF	10km	1310nm
10GBASE-LX4	SMF	10km	1300nm
10GBASE-ER	SMF	40km	1550nm
10GBASE-ZR	SMF	80-120km	1550nm

- **10 Gigabit Copper**

Standard	**Media**	**distance**
10GBASE-CX4	Cat-5e	15m
10GBASE-T	Cat-6 Unshielded	55m
10GBASE-T	Cat-6 Shielded	100m
10GBASE-T	Cat-6a	100m
10GBASE-T	Cat-7	100m

- **40 Gigabit**

40GBASE-CR4	Cat-7	10m	802.3ba
40GBASE-SR4	OM3 MMF	100m	850nm, 4 strands
40GBASE-SR4	OM4 MMF	150m	850nm, 4 strands
40GBASE-LR4	SMF	10km	(4) 10G ~1310nm waves
40GBASE-FR	SMF	10km	(1) 40G 1550nm wave

- **100 Gigabit**

Standard	**Media**	**distance**
100GBASE-CR10	Cat-7	10m
100GBASE-SR10	OM3 MMF	100m
100GBASE-SR10	OM4 MMF	150m
100GBASE-LR4	SMF	10km
100GBASE-ER4	SMF	40km

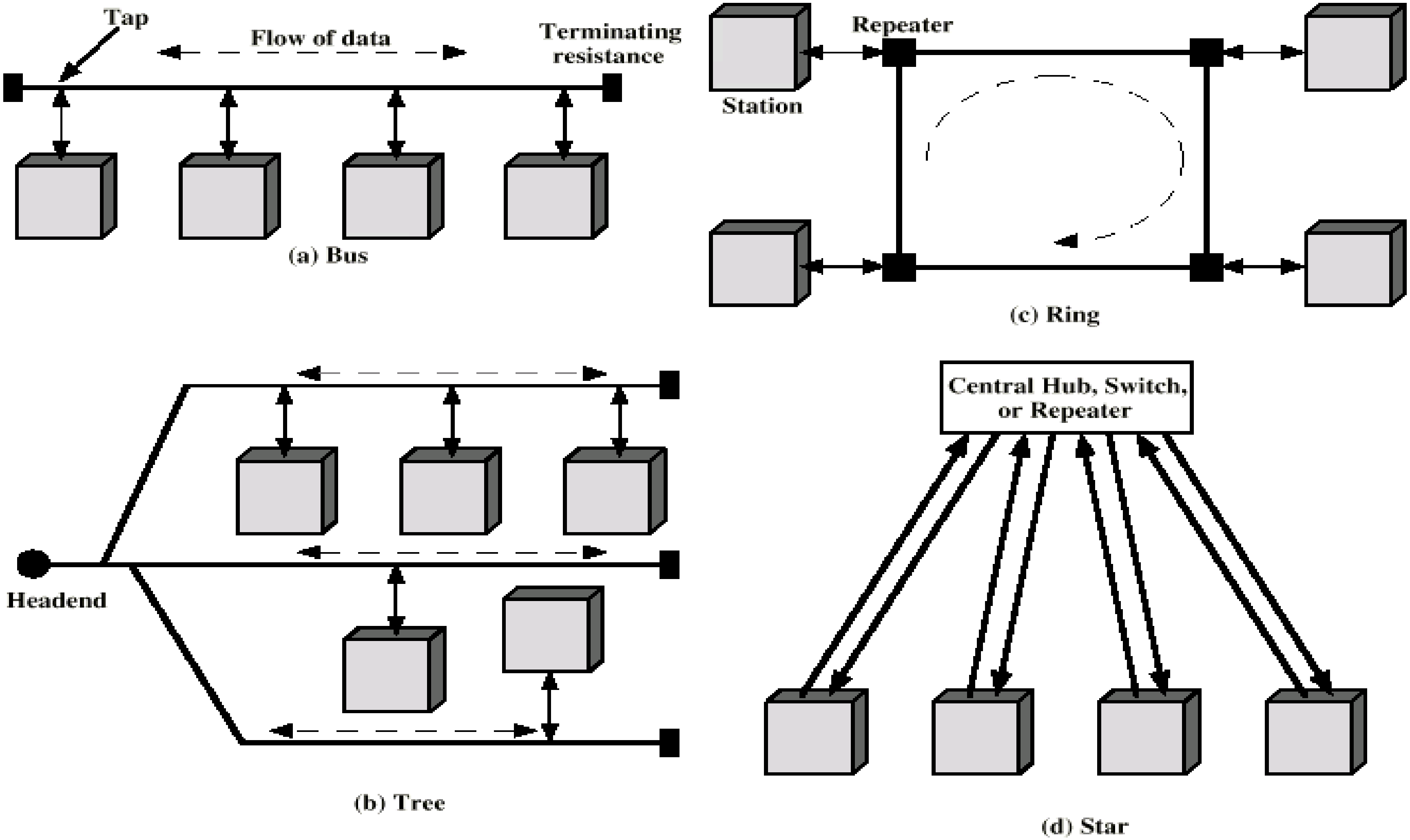
Computer Networks

Local Area Networks

Local Area Networks

- A LAN is a network:
  - provides Connectivity of computers, mainframes, storage devices, etc.
  - spans limited geographical area (up to ~ 2.5 km)
- LAN topologies
- Medium Access Control

LAN Topologies



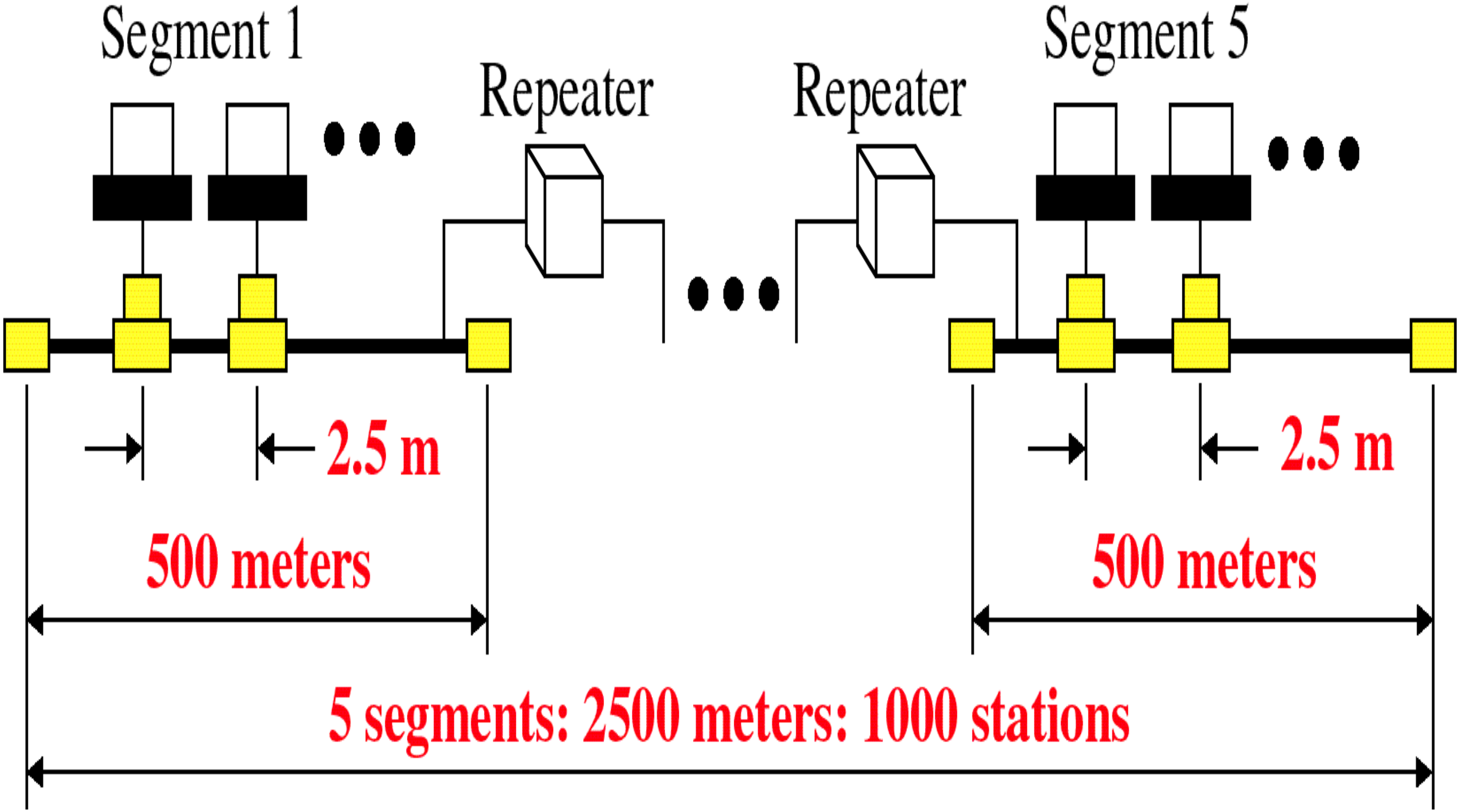
(a) Bus

(c) Ring

(b) Tree

(d) Star

Ethernet Segments

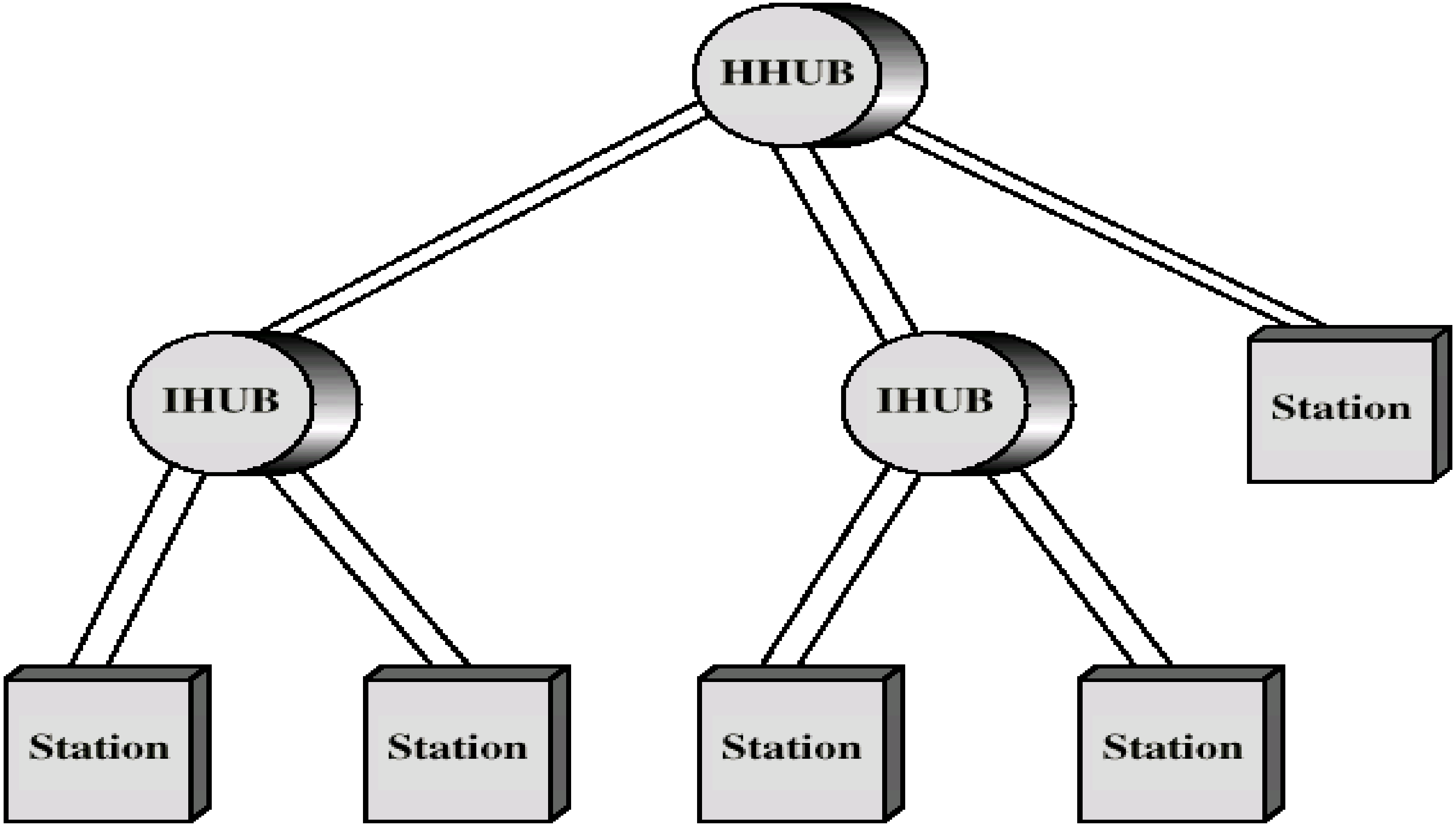


Segment and Backbone

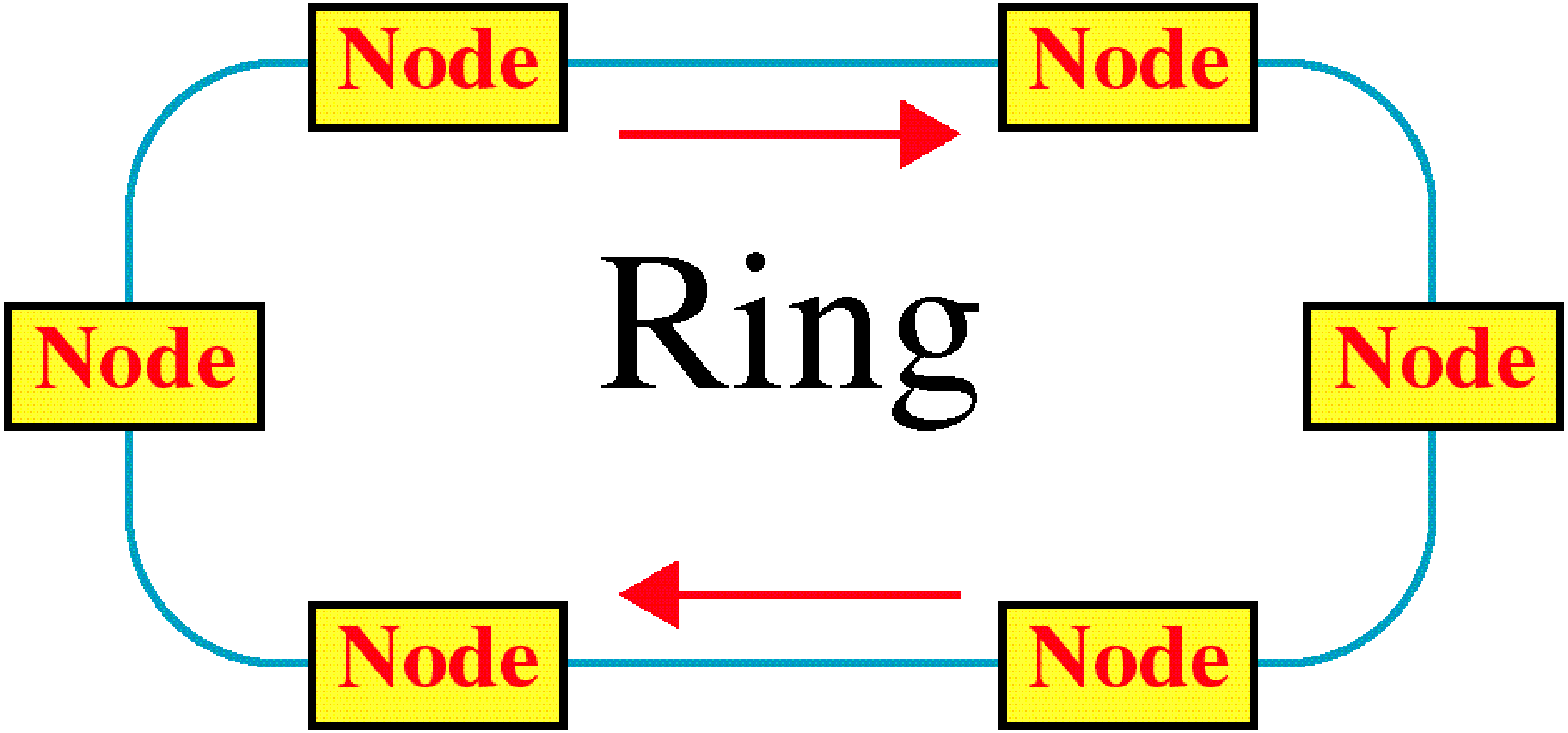
Segment - A segment is any portion of a network that is separated, by a switch, bridge or router, from other parts of the network.

Backbone - The backbone is the main cabling of a network that all of the segments connect to. Typically, the backbone is capable of carrying more information than the individual segments. For example, each segment may have a transfer rate of 10 Mbps (megabits per second), while the backbone may operate at 100 Mbps.

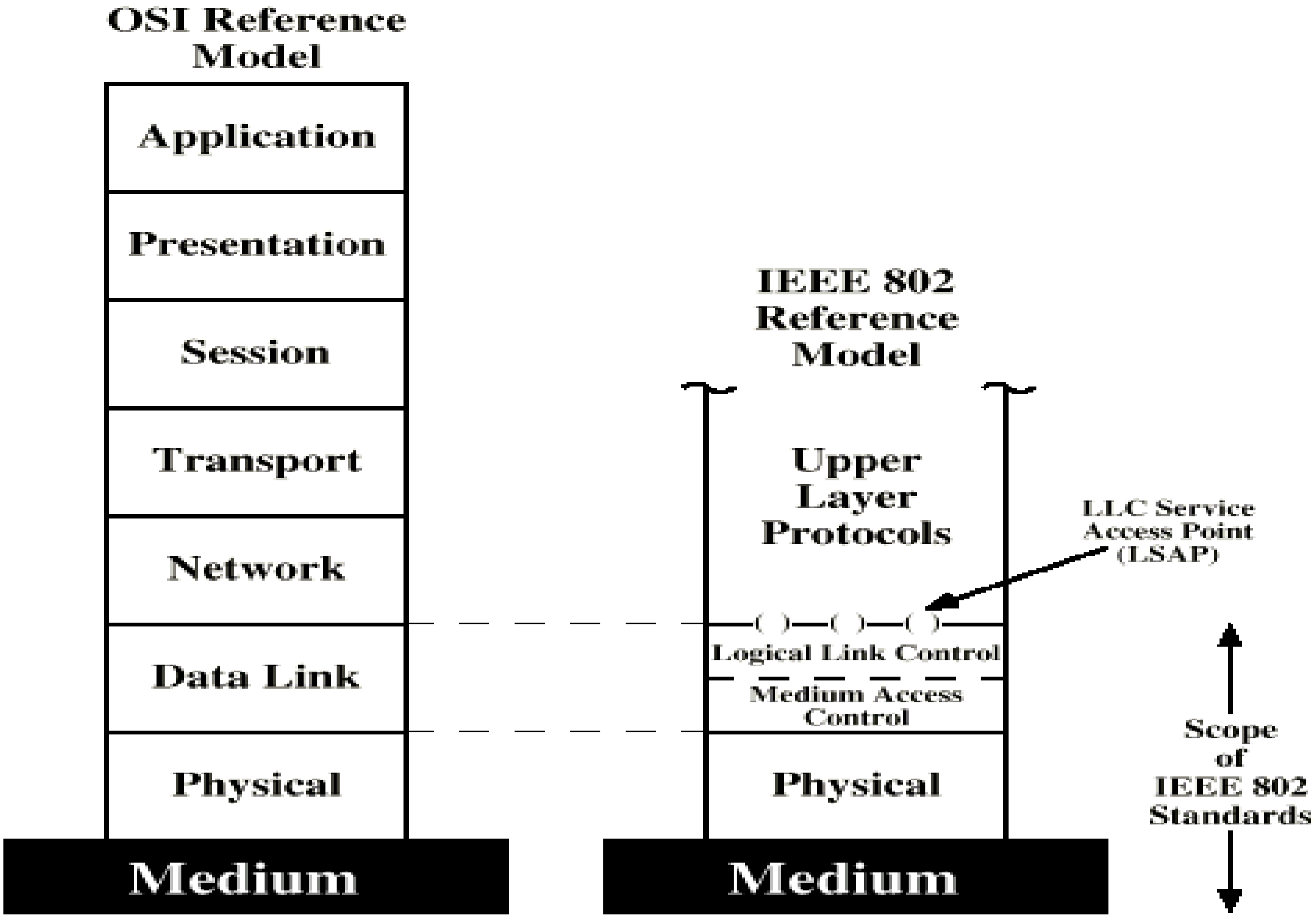
Two-Level Hub Topology



Ring Topology



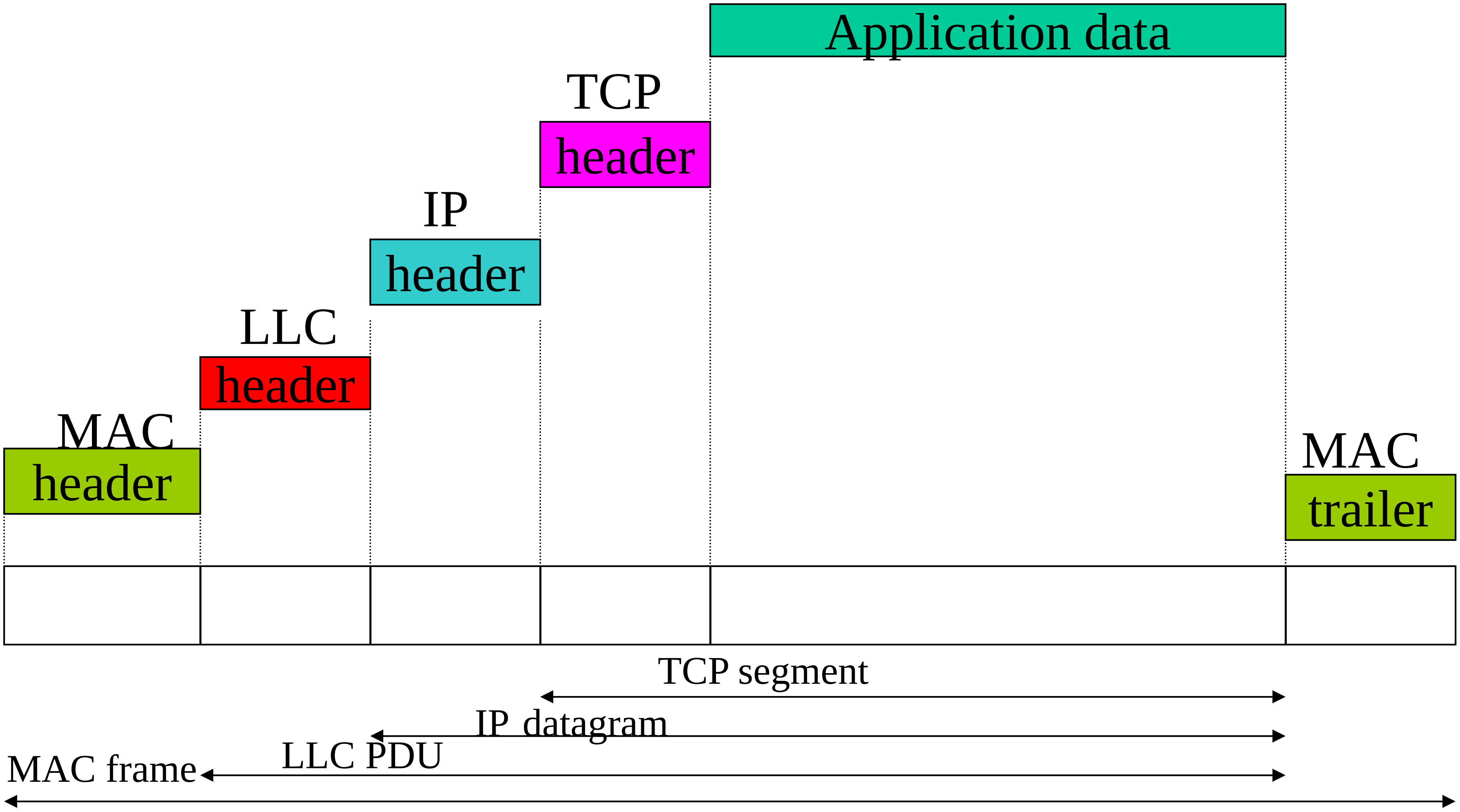
IEEE802 Protocol Suite v.s. OSI



802 standards

- MAC + physical layers
  - 802.3
    - Bus/tree/star topologies
    - CSMA/CD
  - 802.5
    - Token Ring
  - 802.8
    - FDDI
  - 802.11
    - Wireless

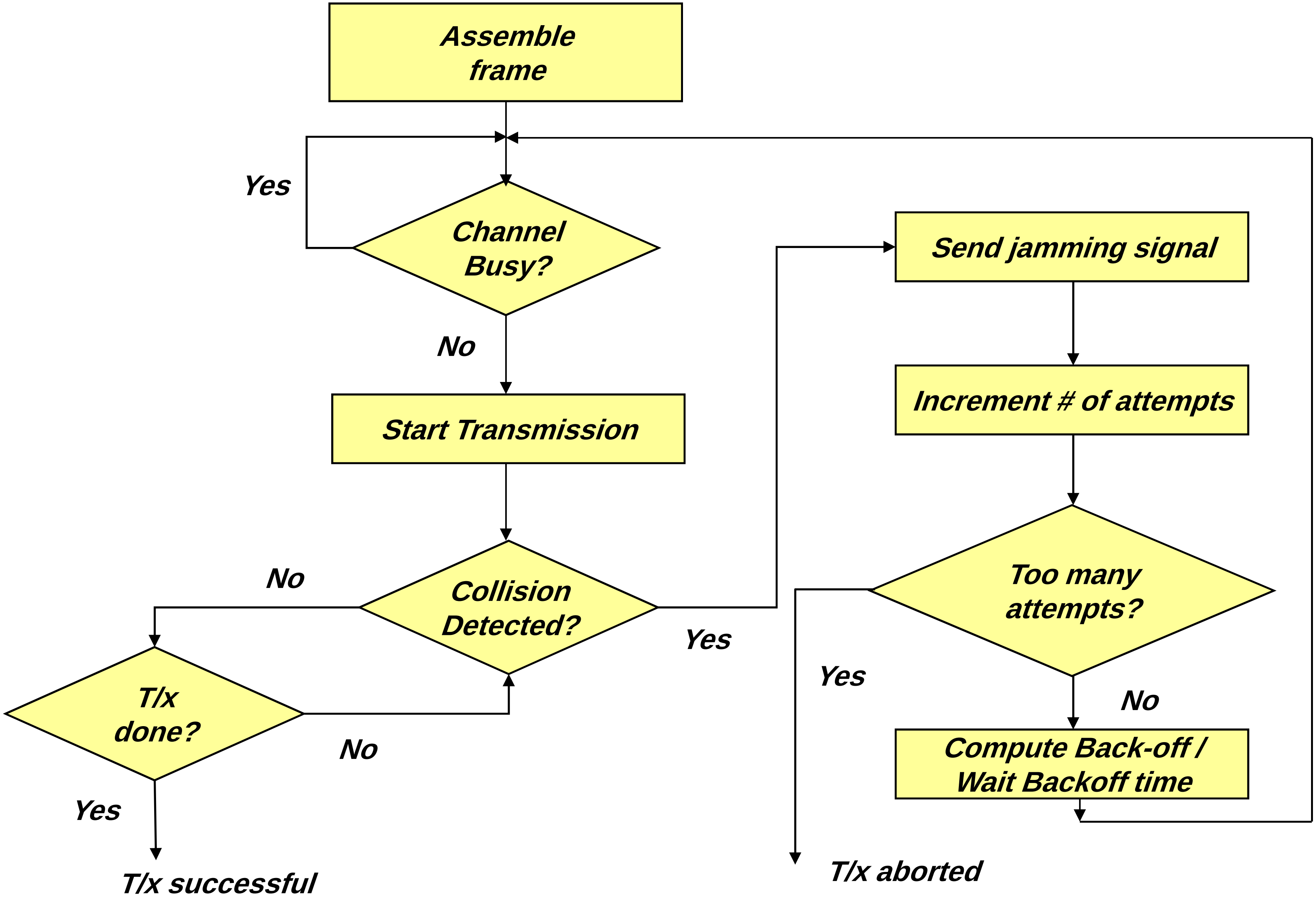
Encapsulation



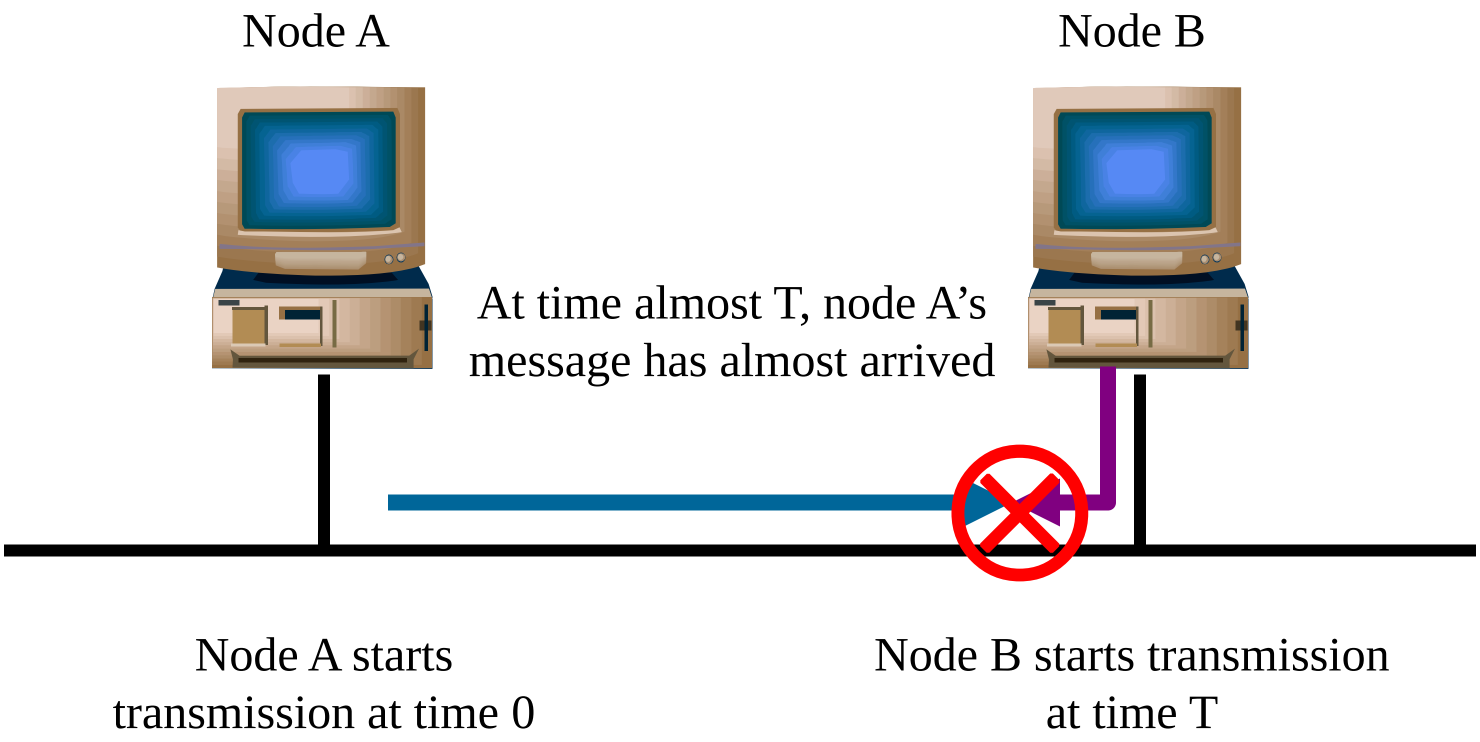
IEEE802.3: CSMA/CD

- Ethernet uses CSMA/CD – listens to line before/during sending
- CSMA/CD: Carrier sense, multiple access with collision detection
  - collisions *detected* within short time
  - colliding transmissions aborted
  - Persistent or non-persistent retransmission
- Collision Detection:
  - On baseband bus, collision produces much higher signal voltage than transmitted signal
  - For Hub-topology activity on more than one port

Transmit Process in IEEE802.3



Collision Detection



Node A

Node B

At time almost T, node A’s message has almost arrived

Node A starts transmission at time 0

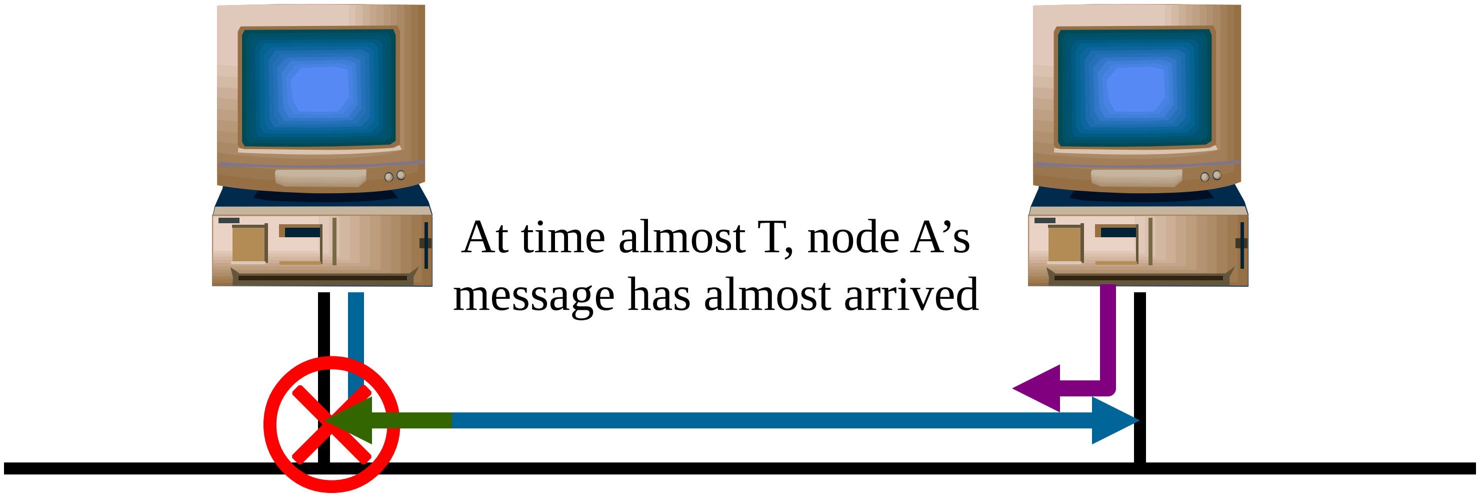
Node B starts transmission at time T

How can we ensure that A knows about the collision?

Collision Detection contd.

Node A

Node B



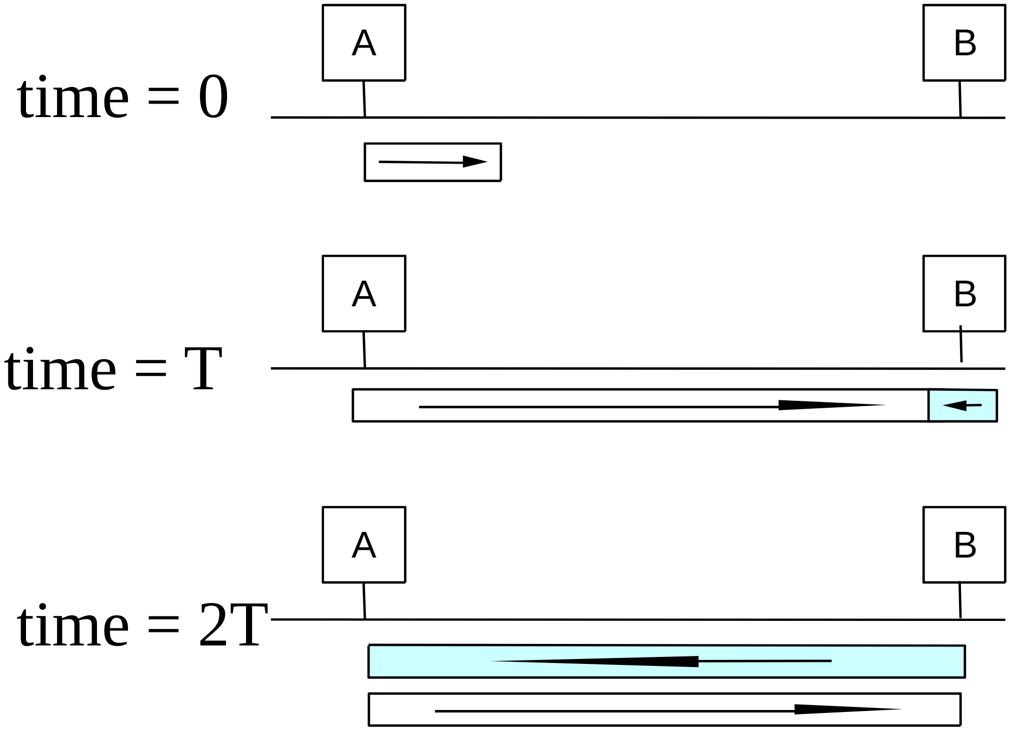
At time almost T, node A’s message has almost arrived

Node A starts transmission at time 0

Node B starts transmission at time T

At time 2T, A is still transmitting and notices a collision

Collision Detection contd.



time = 0

time = T

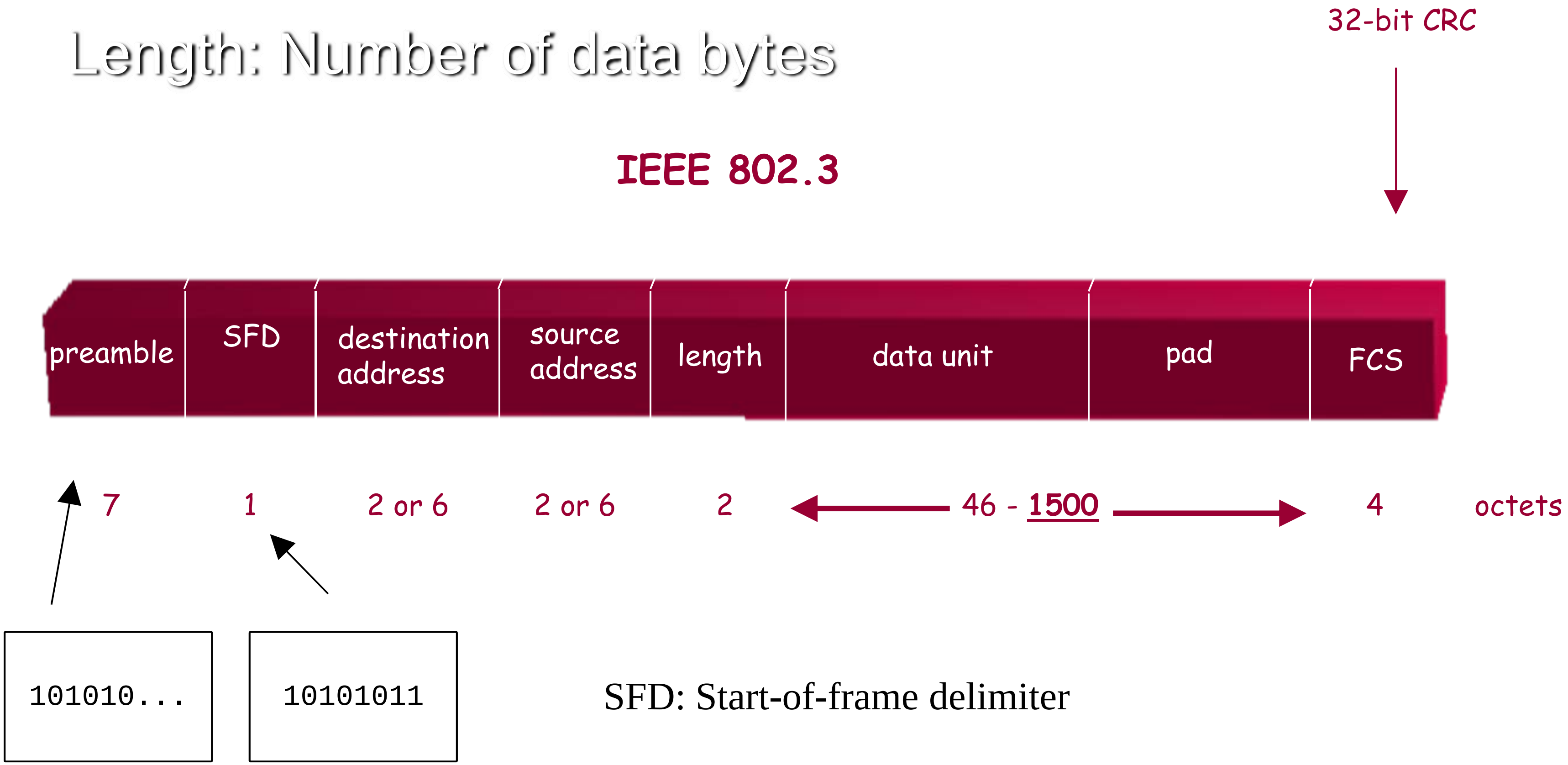
time = 2T

Collision Detection contd.

- How can A know that a collision has taken place?
  - There must be a mechanism to insure retransmission on collision
  - A’s message reaches B at time T
  - B’s message reaches A at time 2T
  - So, A must still be transmitting at 2T
- IEEE 802.3 specifies max value of 2T to be 51.2 micro.sec.
  - This relates to maximum distance of 2500m between hosts
  - At 10Mbps it takes 0.1 micro.sec. to transmit one bit so 512 bits (64B) take 51.2 micro.sec. to send
  - So, Ethernet frames must be at least 64Bytes long
    - Padding is used if data is less than 64Bytes
- Send jamming signal after collision is detected to insure all hosts see collision

IEEE802.3 Frame Format

Length: Number of data bytes



32-bit CRC

IEEE 802.3

preamble	SFD	destination address	source address	length	data unit	pad	FCS	
7	1	2 or 6	2 or 6	2	46 - 1500		4	octets

101010...

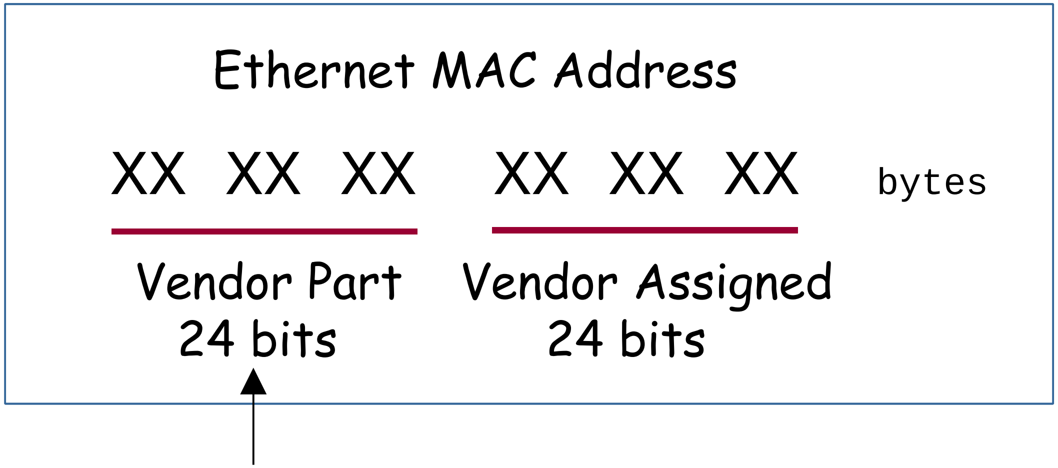
10101011

SFD: Start-of-frame delimiter

used to synchronize receiver, sender clocks

IEEE802.3 MAC Addresses

- Source and destination MAC addresses. These are the ***hardware*** addresses or ***physical*** addresses. They are 6 Octets each



Ethernet MAC Address

XX XX XX XX XX XX bytes

Vendor Part
24 bits

Vendor Assigned
24 bits

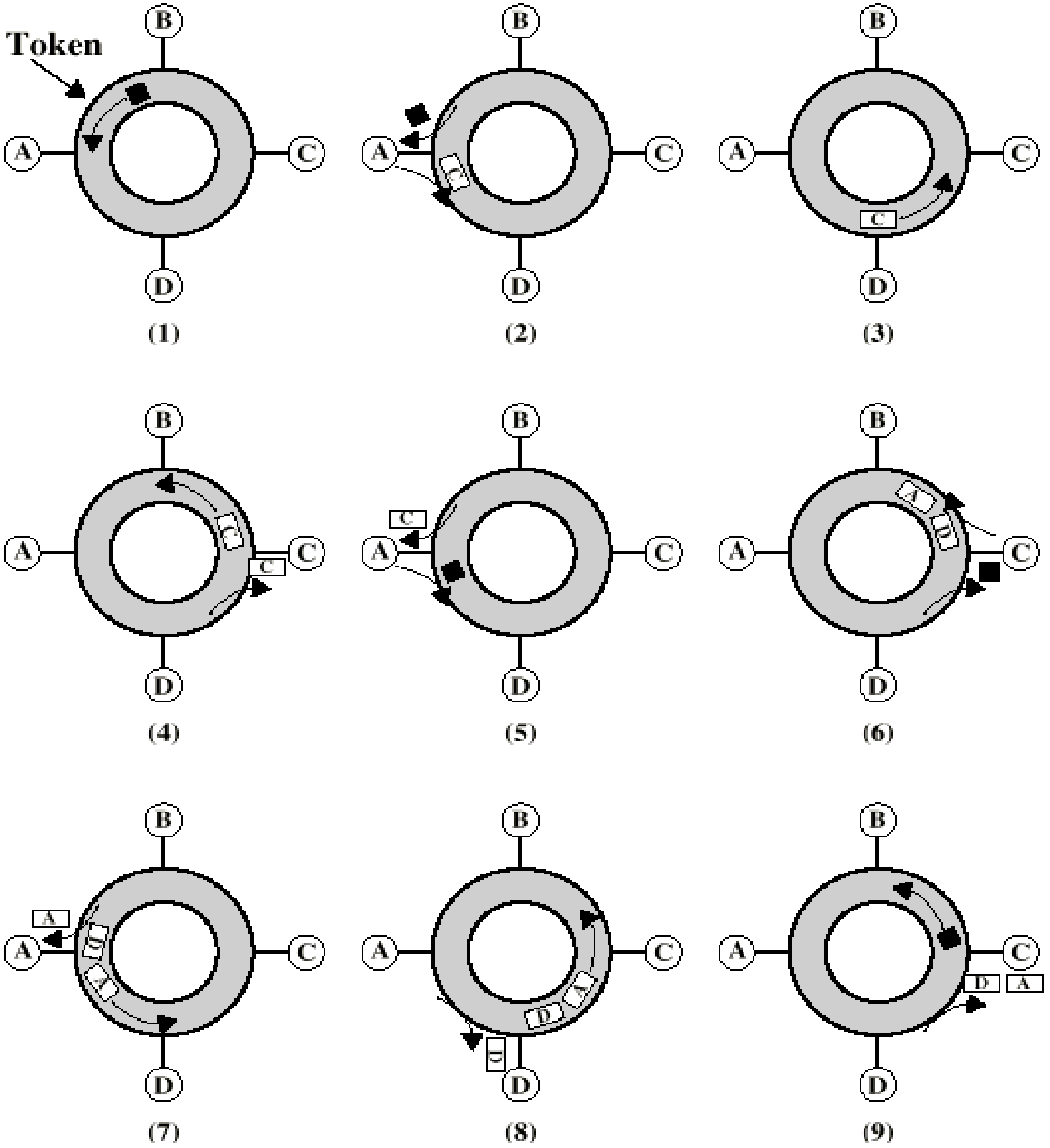
IEEE Organizationally Unique Identifier (OUI)
- allows vendor to build hardware with unique addresses

Windows 2000 > ipconfig /all

IEEE802.5 : Token Ring

- Frames flow in one direction
- Special bit pattern (token) rotates around ring. The token is 24-bit long
- Node having a frame to transmit must capture token first
- Node must release token after done transmitting
- Node remove frame when it comes back around
- Round-robin service

Token Ring Operation

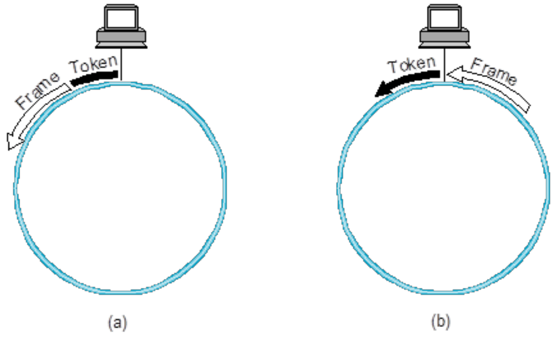


IEEE802.5:Token Ring (16Mbps)

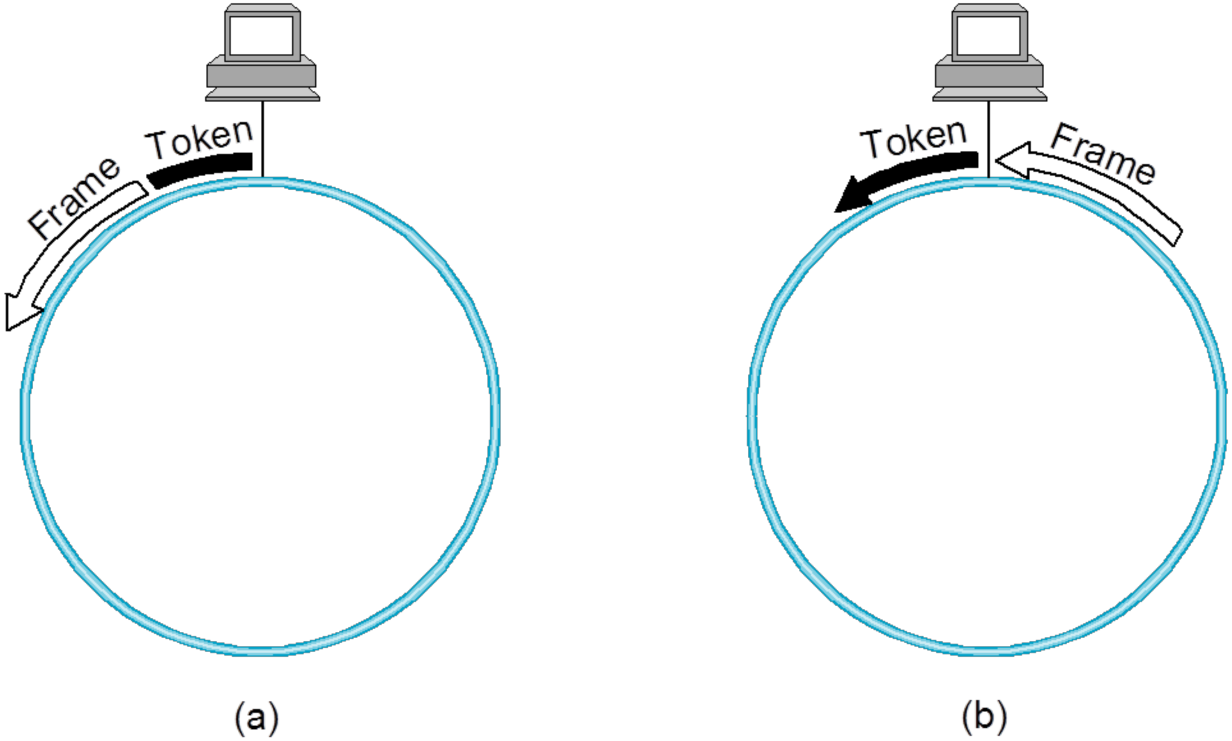
- Token Holding Time (THT)
- Token Rotation Time
  - TRT <= # of Active Nodes * THT + Ring Delay
  - Active nodes denotes the number of nodes that have data to transmit.
  - Ring latency denotes how long it takes the token to circulate around the ring when no one has data to send.
- Monitor Station
  - Introduce additional delay
  - Detect missing token

    # of Nodes * THT + Ring Delay
  - Ring delay- token circulation time

Token Release



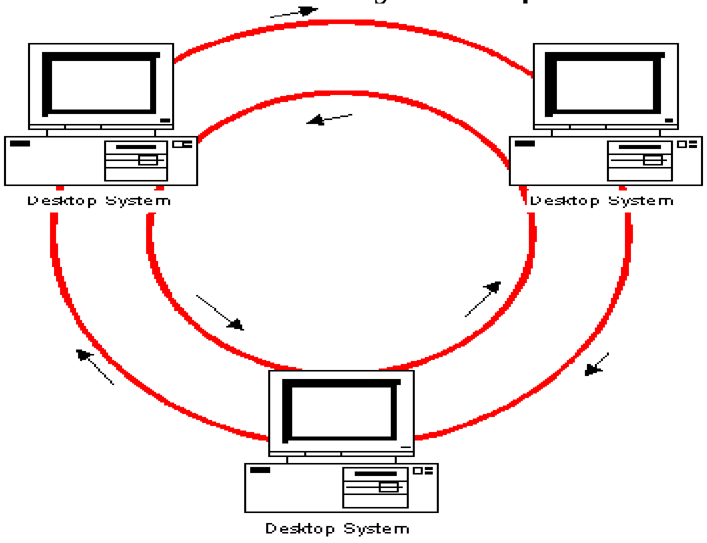
- When the sending node <u>releases</u> the token
  - early release
    - the sender inserts the token back onto the ring immediately following its frame
    - better bandwidth utilization, especially on large rings
  - delayed release
    - after the frame it transmits has gone all the way around the ring and been removed
  - 802.5 originally used delayed token release, but support for early release was subsequently added



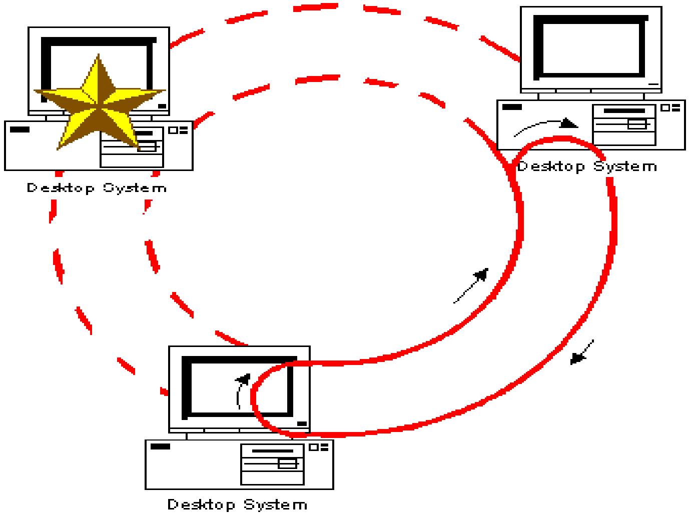
Token release: (a) early versus (b) delayed

Fiber Distributed Data Interface protocol
(IEEE 802.8 100Mbps)

- As opposed to Token Ring's single ring, FDDI, uses two to achieve better results and less chance of failure.
- In a basic Token Ring network, at any instant there is a single active ring monitor which supplies the master clock for the ring, whereas in FDDI this approach isn't ideal because of the high data rates. Instead, each ring interface has its own local clock, and outgoing data is transmitted using this clock.
- Due to its dual ring architecture, FDDI has the ability to recover from link and station failures. If a station goes down, the signals are routed around it by a loop formed from the rings



FDDI - all stations functioning



FDDI - one station is down

Overview of Leading Wireless Technologies

	Bluetooth 802.15.1	Wi-Fi 802.11	WiMAX 802.16	3G Cellular
Typical link length	10m	100m	10km	Tens of km
Typical bandwidth	2.1 Mbps (shared)	54 Mbps (shared)	70 Mbps (shared)	384+ Kbps (per connection)
Typical use	Link a peripheral to a notebook computer	Link a notebook computer to a wired base	Link a building to a wired tower	Link a cell phone to a wired tower
Wired technology analogy	USB	Ethernet	Coaxial cable	DSL

Data Link Layer:
Flow and Error Control

Link Control Mechanisms

3 techniques at link level:

- Stop-and-wait
- Go-back-N
- Selective-reject

Latter 2 are special cases of sliding-window

Assume 2 end systems connected by direct link

Flow Control

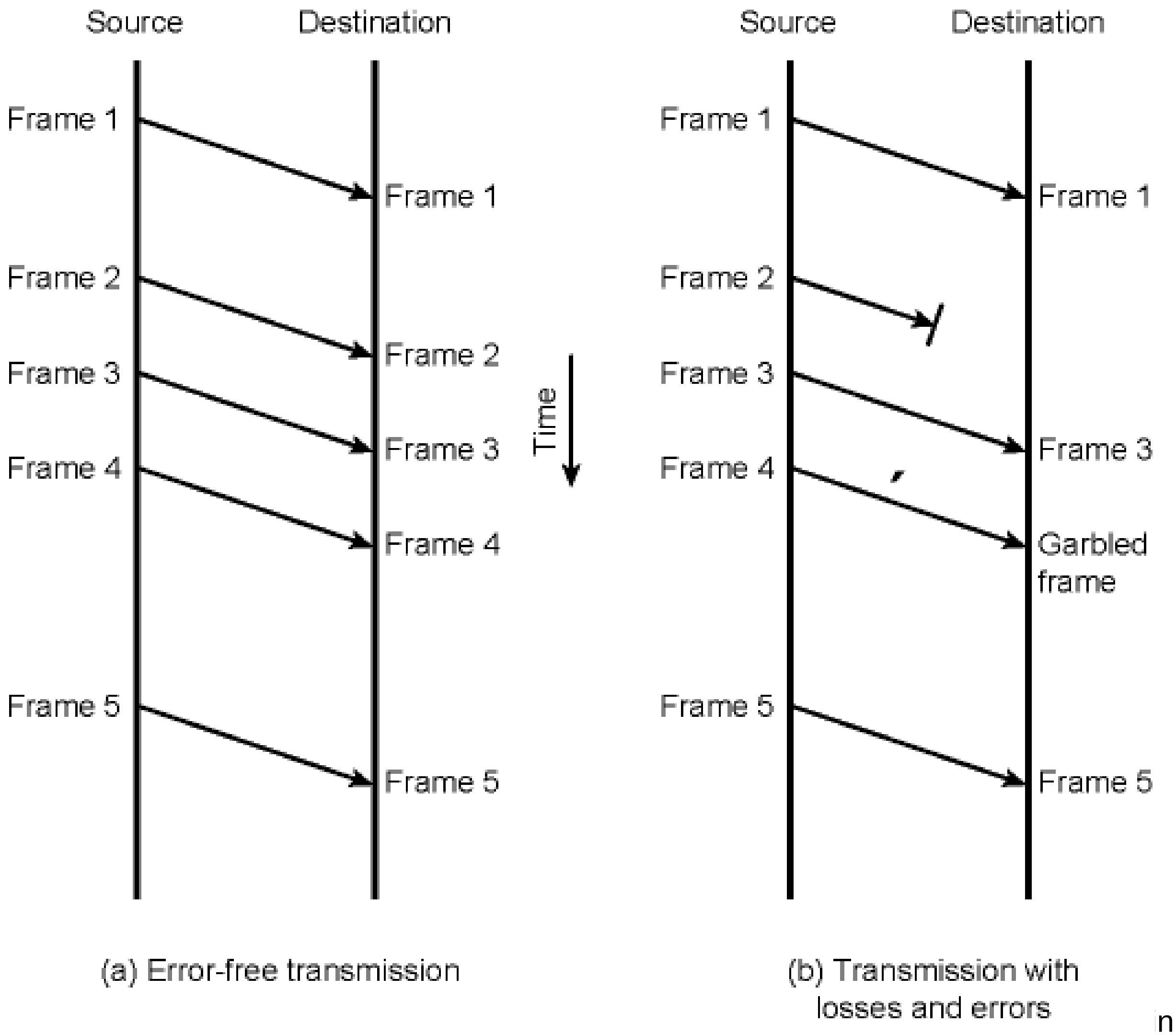
- Ensuring the sending entity does not overwhelm the receiving entity
- Limits the amount or rate of data that is sent
- Preventing buffer overflow
  - Source may send PDUs faster than destination can process headers
  - Higher-level protocol user at destination may be slow in retrieving data
  - Destination may need to limit incoming flow to match outgoing flow for retransmission

Sequence of Frames

Source breaks up message into sequence of frames

- Buffer size of receiver may be limited
- Longer transmission are more likely to have an error

Model of Frame Transmission



Automatic Repeat Request (ARQ)

- Uses:
  - Error detection
  - Timers
  - Acknowledgements
  - Retransmissions
- Algorithms
  - Stop and wait
  - Go back N
  - Selective reject (selective retransmission)

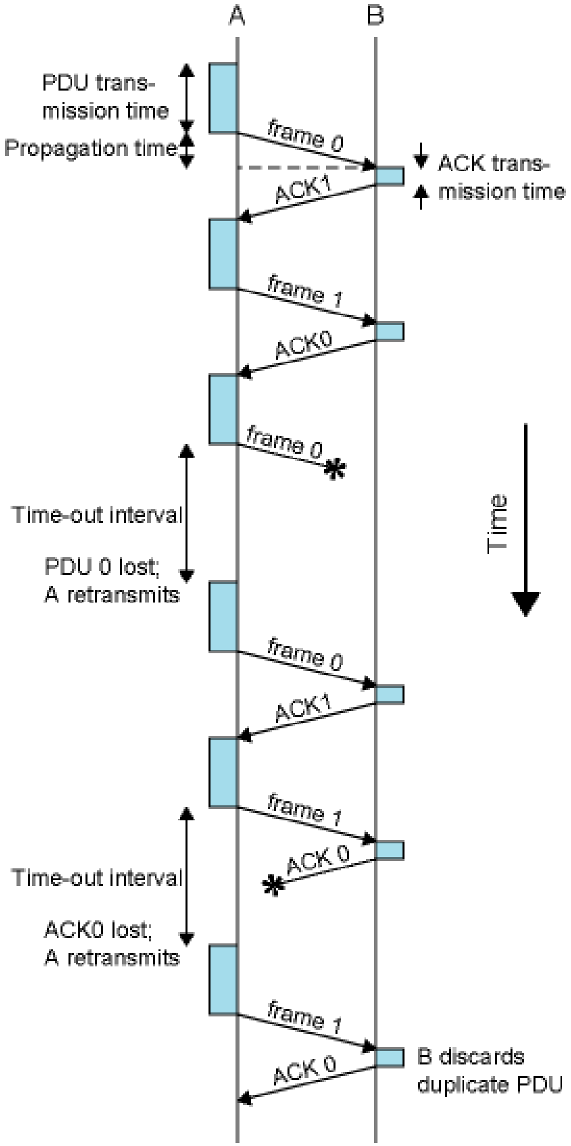
Stop and Wait

- Source transmits frame
- Destination receives frame and replies with acknowledgement
- Source waits for ACK before sending next frame
- Destination can stop flow by not sending ACK
- Works well for a few large frames

Stop and Wait

- 2 kinds of errors:
  - Damaged frame at destination
  - Damaged acknowledgement at source
- If received frame damaged, discard it
  - Transmitter has timeout
  - If no ACK within timeout, retransmit
- If ACK damaged,transmitter will not recognize it
  - Transmitter will retransmit
  - Receive gets two copies of frame
  - Use ACK0 and ACK1

Stop-and-Wait ARQ



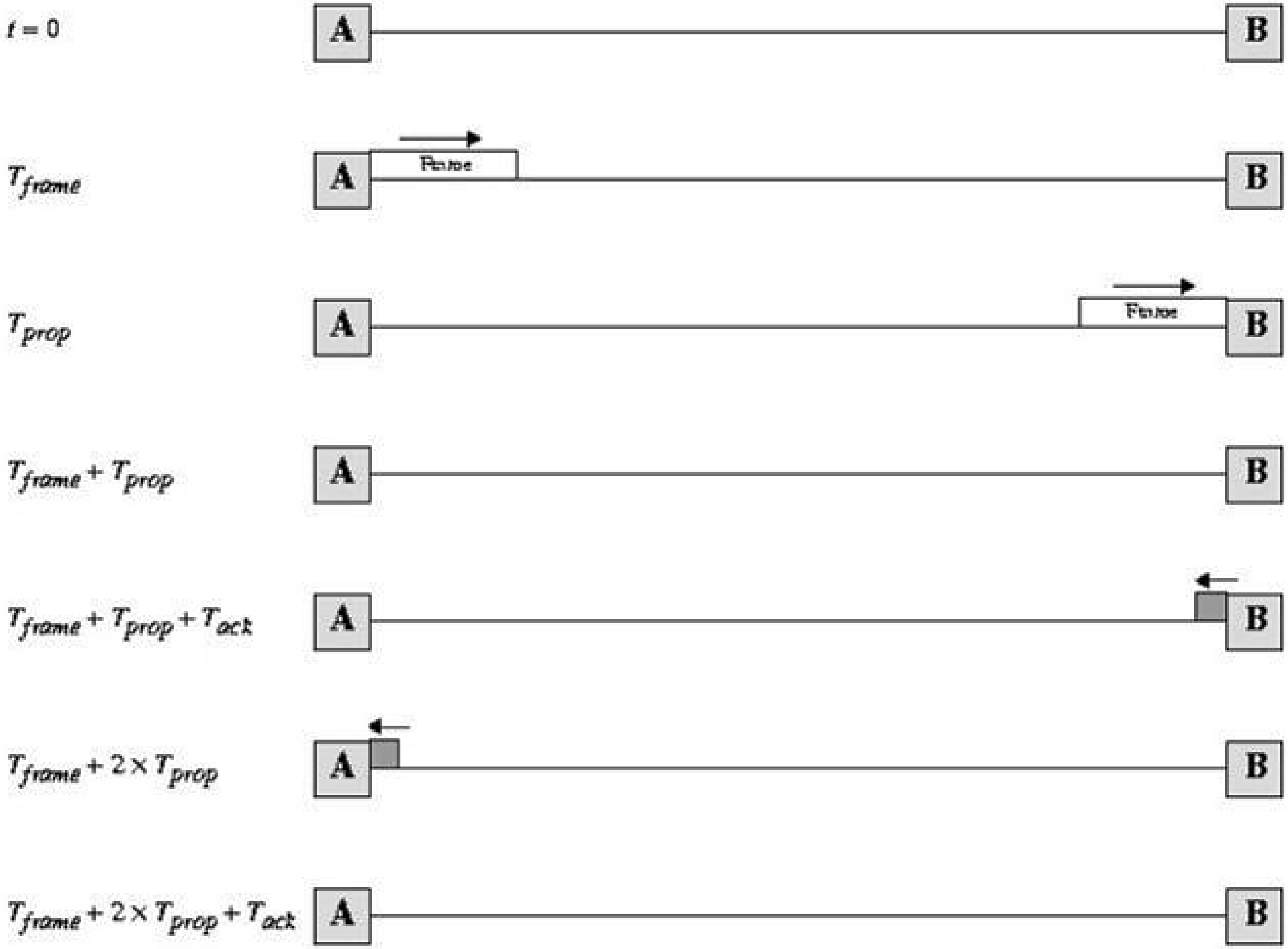

$t = 0$
A
B
T_{frame}
Frame
T_{prop}
Frame
$T_{frame} + T_{prop}$
$T_{frame} + T_{prop} + T_{ack}$
$T_{frame} + 2 \times T_{prop}$
$T_{frame} + 2 \times T_{prop} + T_{ack}$

Figure 11.4 Stop-and-Wait Link Utilization

Stop-and-Wait Link Utilization

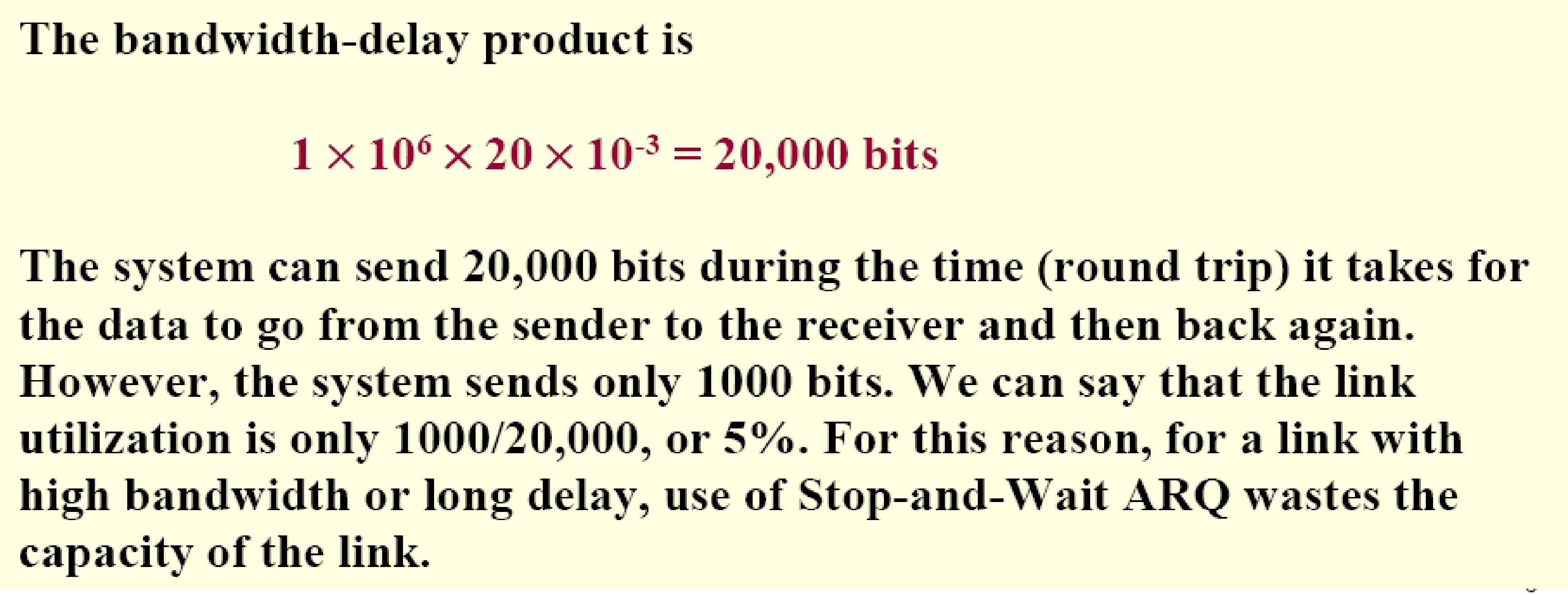
- If T_{prop} large relative to T_{frame} then throughput reduced
- If propagation delay is long relative to transmission time, line is mostly idle
- Problem is only one frame in transit at a time
- Stop-and-Wait rarely used because of inefficiency

Example

In a Stop-and-Wait ARQ system, the bandwidth of the line is 1 Mbps, and 1 bit takes 20 ms to make a round trip. What is the bandwidth-delay product? If the system data frames are 1000 bits in length, what is the utilization percentage of the link?

Solution

Efficiency will be 50% when the time to transmit the frame equals the round-trip propagation delay.
This argument can be easily proven.



The bandwidth-delay product is

$$1 \times 10^6 \times 20 \times 10^{-3} = 20,000 \text{ bits}$$

The system can send 20,000 bits during the time (round trip) it takes for the data to go from the sender to the receiver and then back again. However, the system sends only 1000 bits. We can say that the link utilization is only 1000/20,000, or 5%. For this reason, for a link with high bandwidth or long delay, use of Stop-and-Wait ARQ wastes the capacity of the link.

This solution is not accurate 100%.

- T_{trans}= 1000bits/(1 Mbps)=1ms
- T_{prop}= 10ms

- 1 bit → 10 ms to arrive
- The whole frame arrives at (10+1)=11ms
- The second frame will be send at time = 11ms+ 10ms (ACK propagation)= 21ms.

- Now link efficiency = T_{trans}/Overall time= 1ms/21ms=0.048

- A channel has a bit rate of 4 Kbps and a propagation delay of 20 msec. For what range of frame sizes does stop-and-wait give an efficiency of at least 50 percent?

- Efficiency will be 50% when the time to transmit the frame equals the round-trip propagation delay. At a transmission rate of 4 bits/ms, 160 bits takes 40 ms. For frame sizes above 160 bits, stop-and-wait is reasonably efficient.
- Efficiency increases as the transmission time increases and propagation delay decreases.

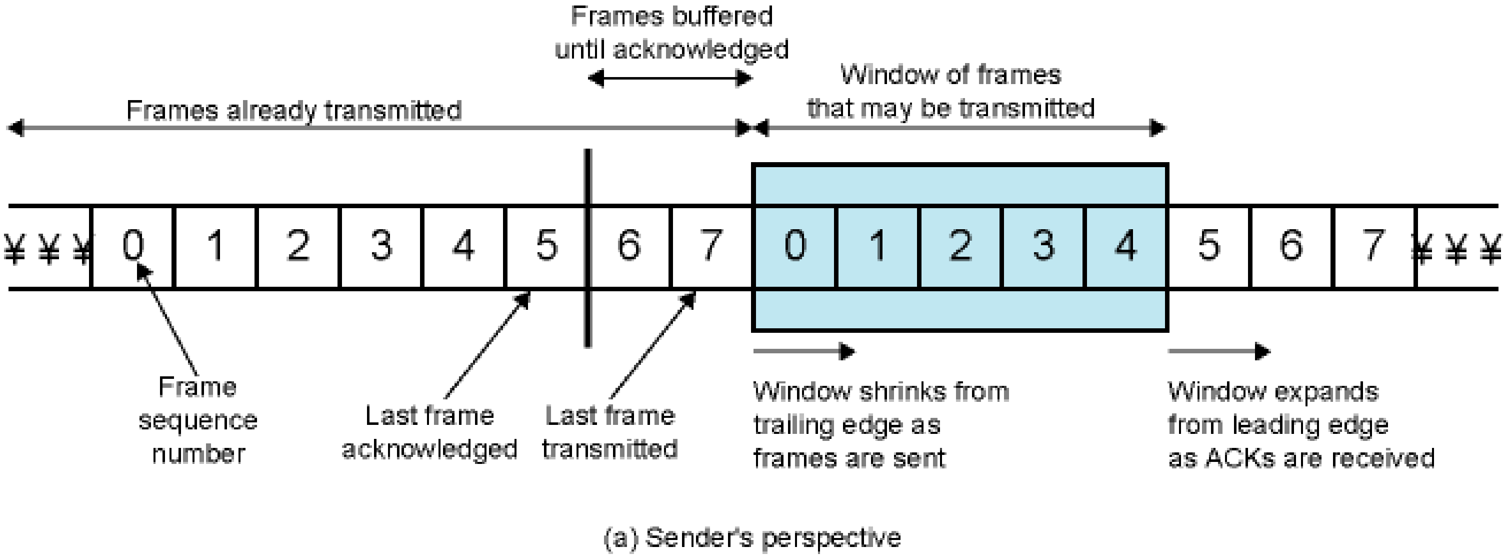
Fragmentation

- Large block of data may be split into small frames
  - Limited buffer size
  - Errors detected sooner
  - On error, retransmission of smaller frames is needed
  - Prevents one station occupying medium for long periods
- Stop and wait becomes inadequate

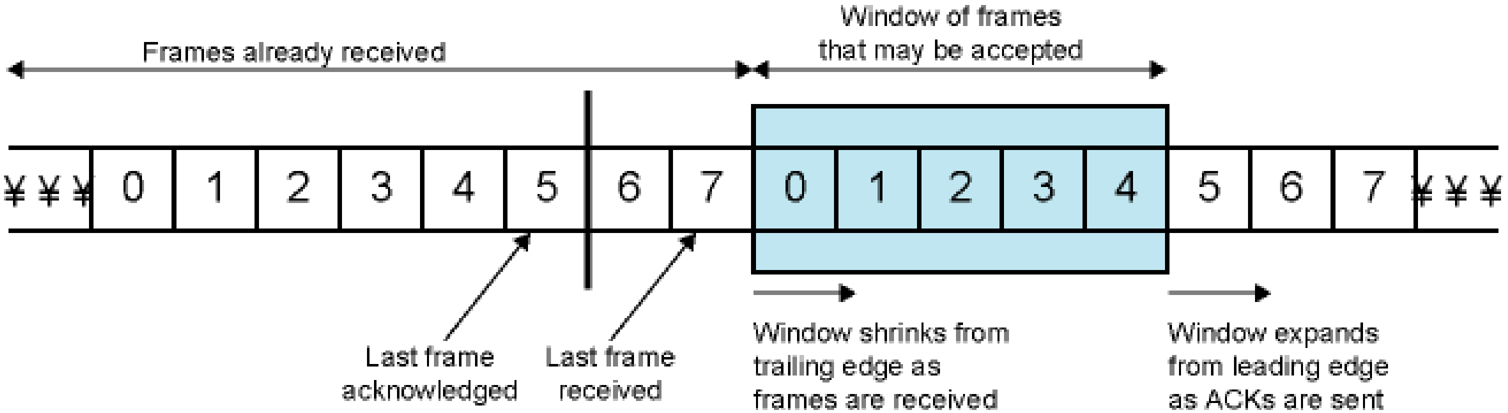
Sliding Windows Flow Control

- Allow multiple frames to be in transit
- Receiver has buffer W long
- Transmitter can send up to W frames without ACK
- Each frame is numbered
- ACK includes number of next frame expected
- Sequence number bounded by size of field (k)
  - Frames are numbered modulo 2^k

Sliding Window

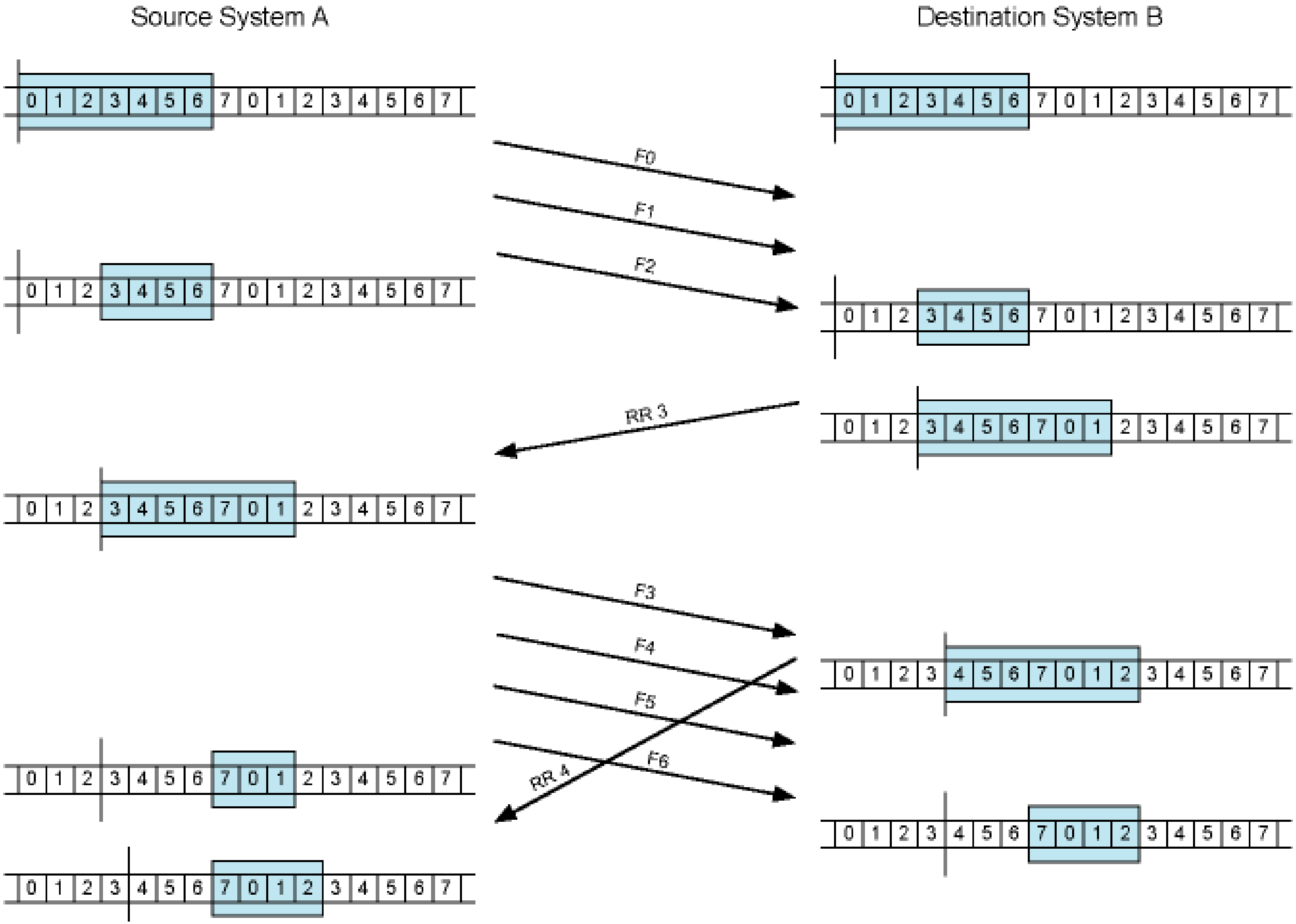

Frames buffered until acknowledged
Window of frames that may be transmitted
Frames already transmitted
¥ ¥ ¥ 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 ¥ ¥ ¥
Frame sequence number
Last frame acknowledged
Last frame transmitted
Window shrinks from trailing edge as frames are sent
Window expands from leading edge as ACKs are received

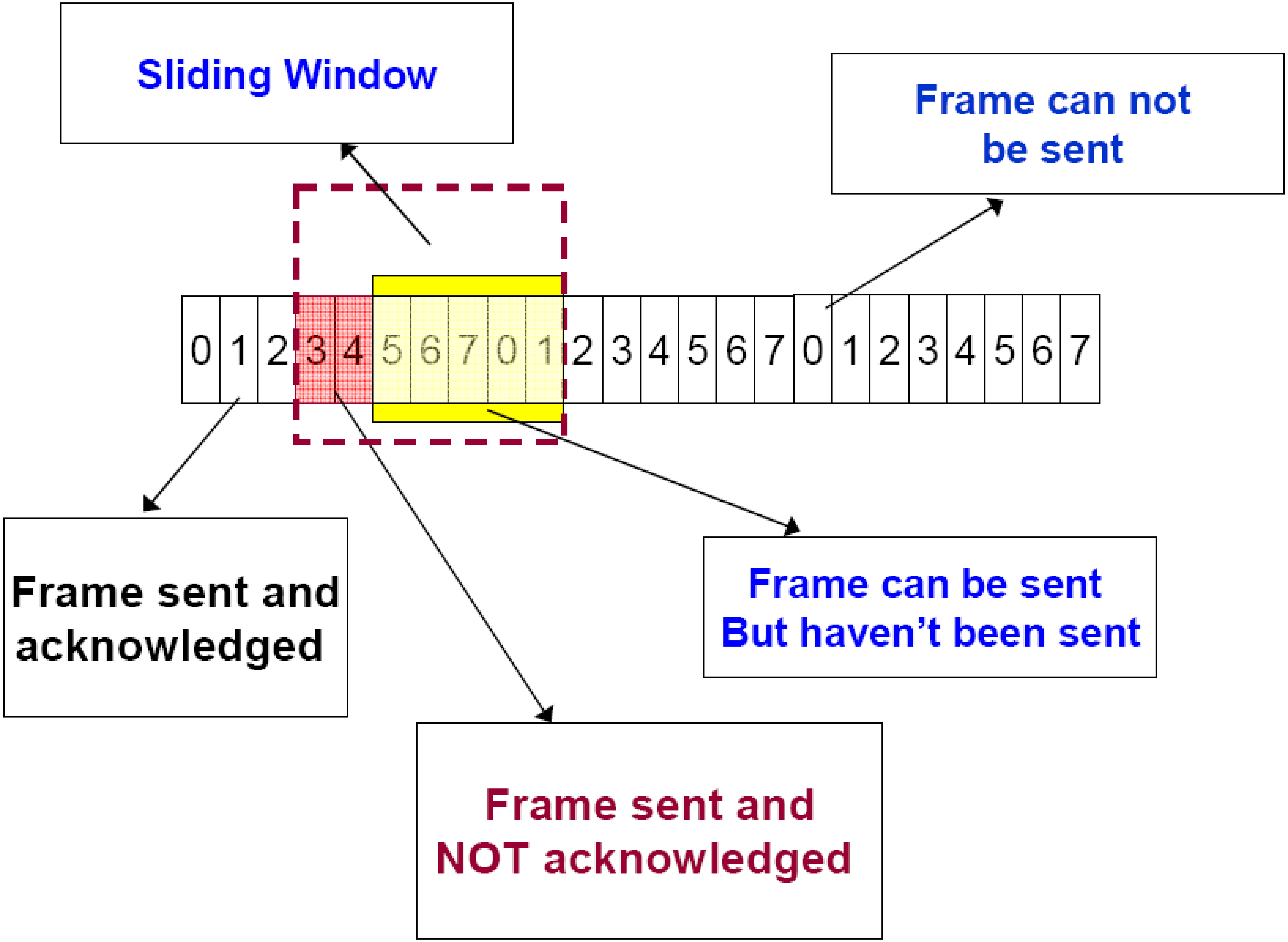
(a) Sender's perspective


Window of frames that may be accepted
Frames already received
¥ ¥ ¥ 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 ¥ ¥ ¥
Last frame acknowledged
Last frame received
Window shrinks from trailing edge as frames are received
Window expands from leading edge as ACKs are sent

(b) Receiver's perspective

Example of a Sliding Window Protocol





Sliding Window

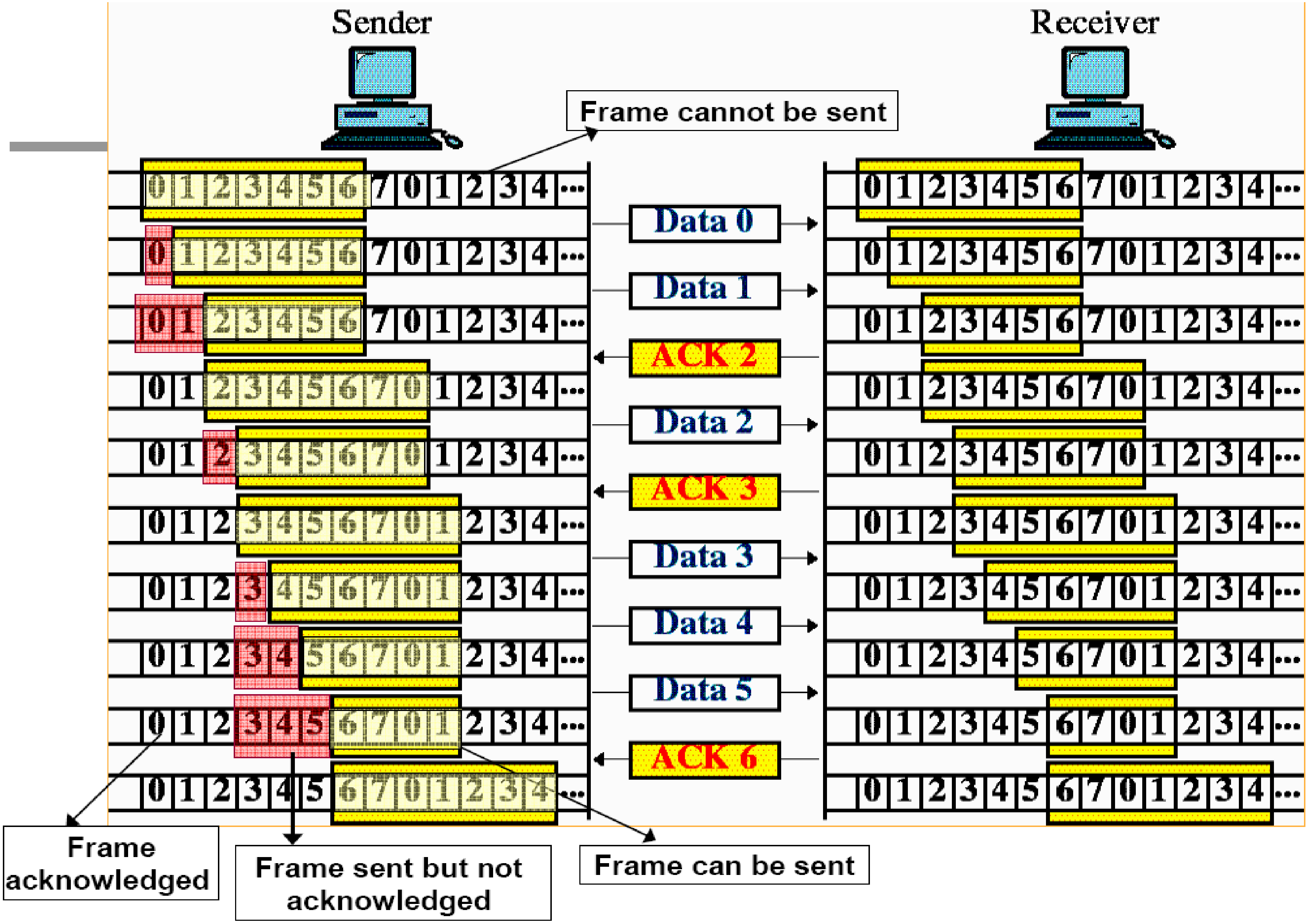
Frame can not be sent

0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7

Frame sent and acknowledged

Frame can be sent But haven’t been sent

Frame sent and NOT acknowledged



Sender

Receiver

Frame cannot be sent

Data 0

Data 1

ACK 2

Data 2

ACK 3

Data 3

Data 4

Data 5

ACK 6

Frame acknowledged

Frame sent but not acknowledged

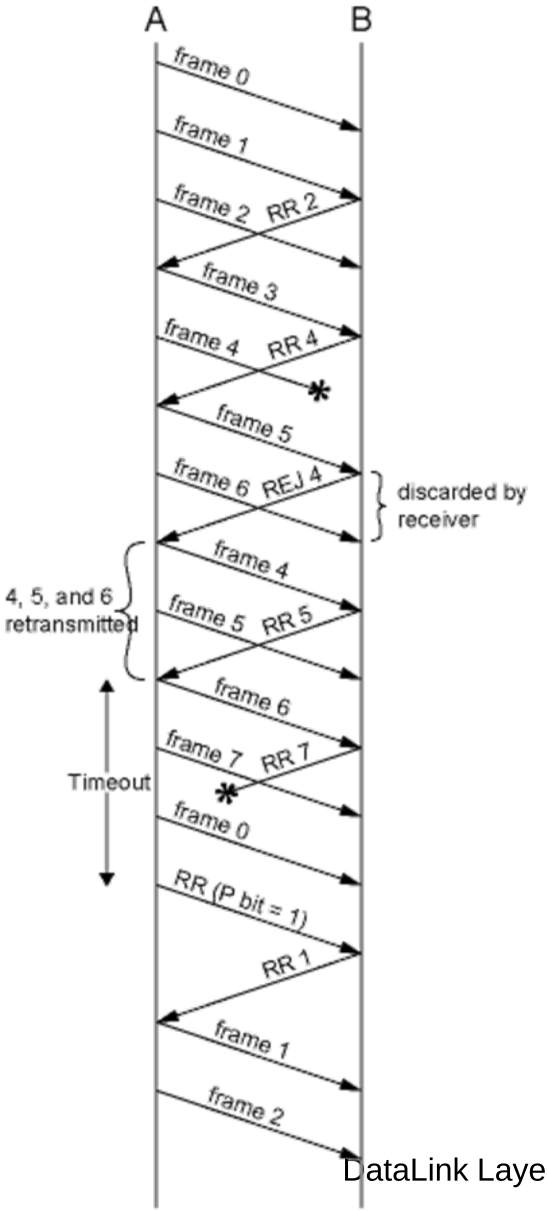
Frame can be sent

Go-Back-N ARQ

- Based on sliding window
- If no error, ACK as usual with next frame expected
- Use window to control number of outstanding frames
- If error, reply with rejection
  - Discard that frame and all future frames until error frame received correctly
  - Transmitter must go back and retransmit that frame and all subsequent frames

Go-Back-N ARQ Protocol

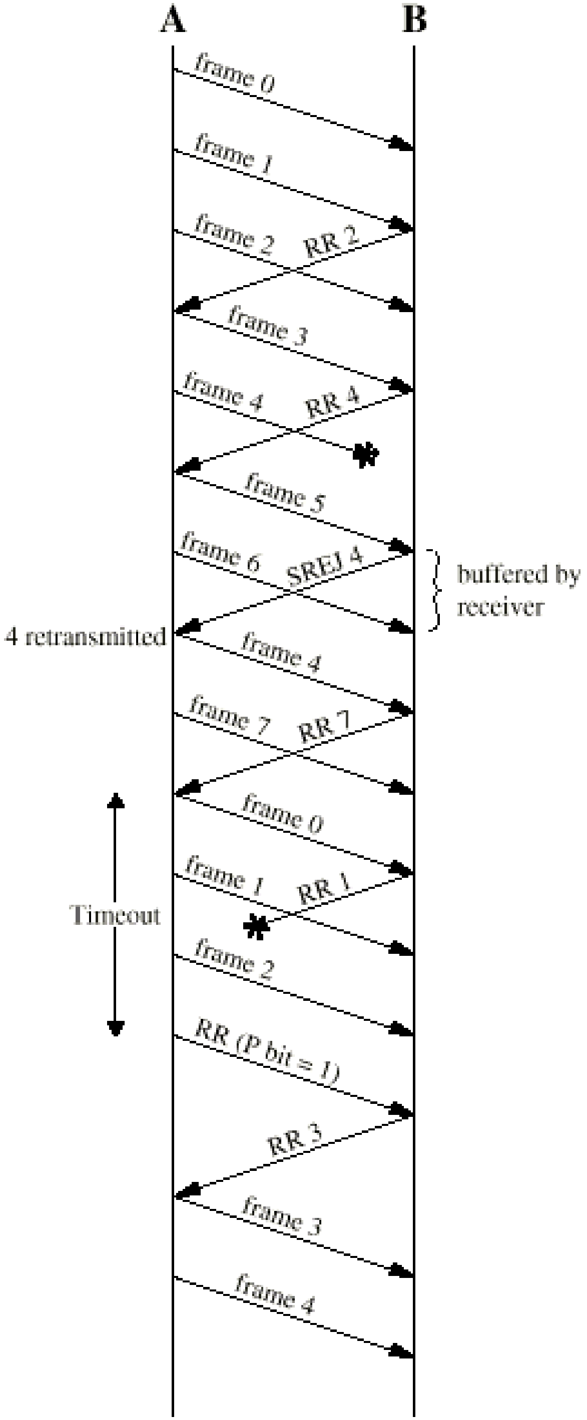
When A's timer expires, it transmits an RR frame that includes a bit known as the P bit, which is set to 1. B interprets the RR frame with a P bit as a **command** that must be acknowledge by sending an RR indicating the next frame that it expects.

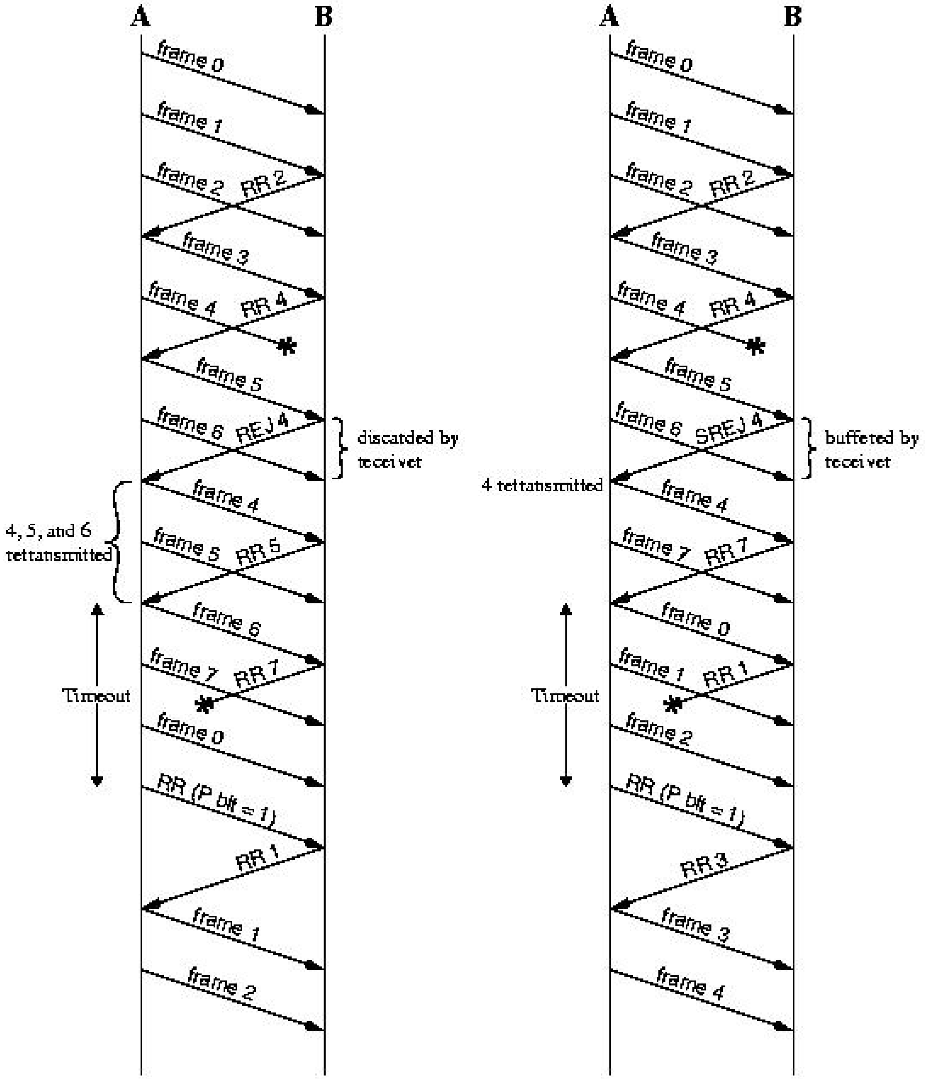


Selective Reject

- Also called selective retransmission
- Only rejected frames are retransmitted
- Subsequent frames are accepted by the receiver and buffered
- Selective reject would appear to be more efficient than go-back-N, because it minimizes the amount of retransmission. On the other hand, the receiver must maintain a buffer large enough to save post-SREJ frames until the frame in error is retransmitted and must maintain logic for reinserting that frame in the proper sequence.
- The transmitter, too, requires more complex logic to be able to send a frame out of sequence.
- Because of such complications, select-reject ARQ is much less widely used than go-back-N ARQ.

Selective Reject - Diagram





(a) Go-back-N ARQ

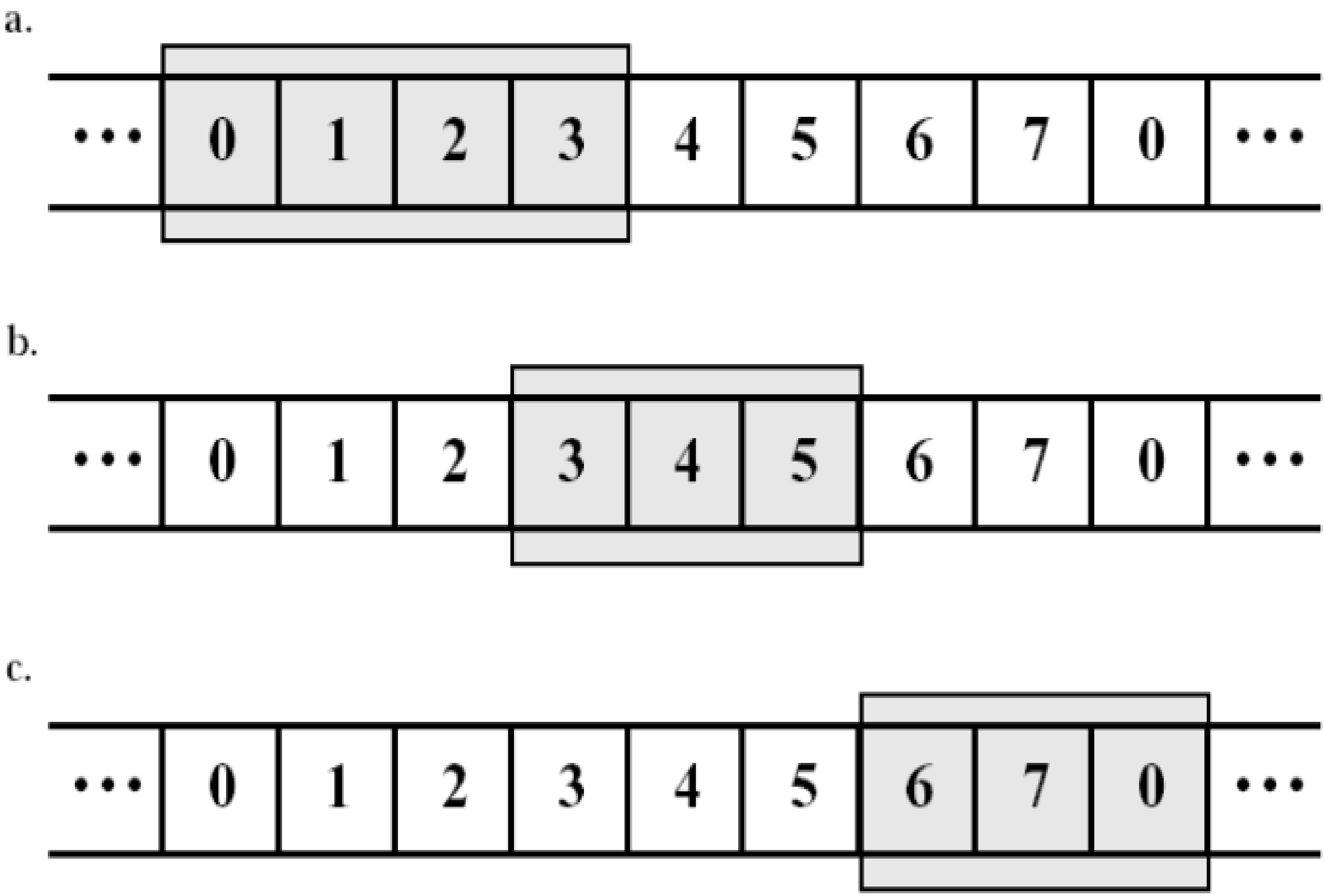
(b) Selective-reject ARQ

Figure 11.7 Sliding-Window ARQ Protocols

Example

- Two neighboring nodes (A and B) use a sliding-window protocol with a 3-bit sequence number. As the ARQ mechanism, Go-back-N is used with a window size of 4. assuming that A is transmitting and B is receiving. Show the window positions for the following succession of events:
- **Before A sends any frames**
- **After A sends frames 0,1,2 and receives acknowledgment from B for 0 and 1**
- **After A sends frames 3, 4, and 5 and B acknowledges 4 and the ACK is received by A.**

Answer:



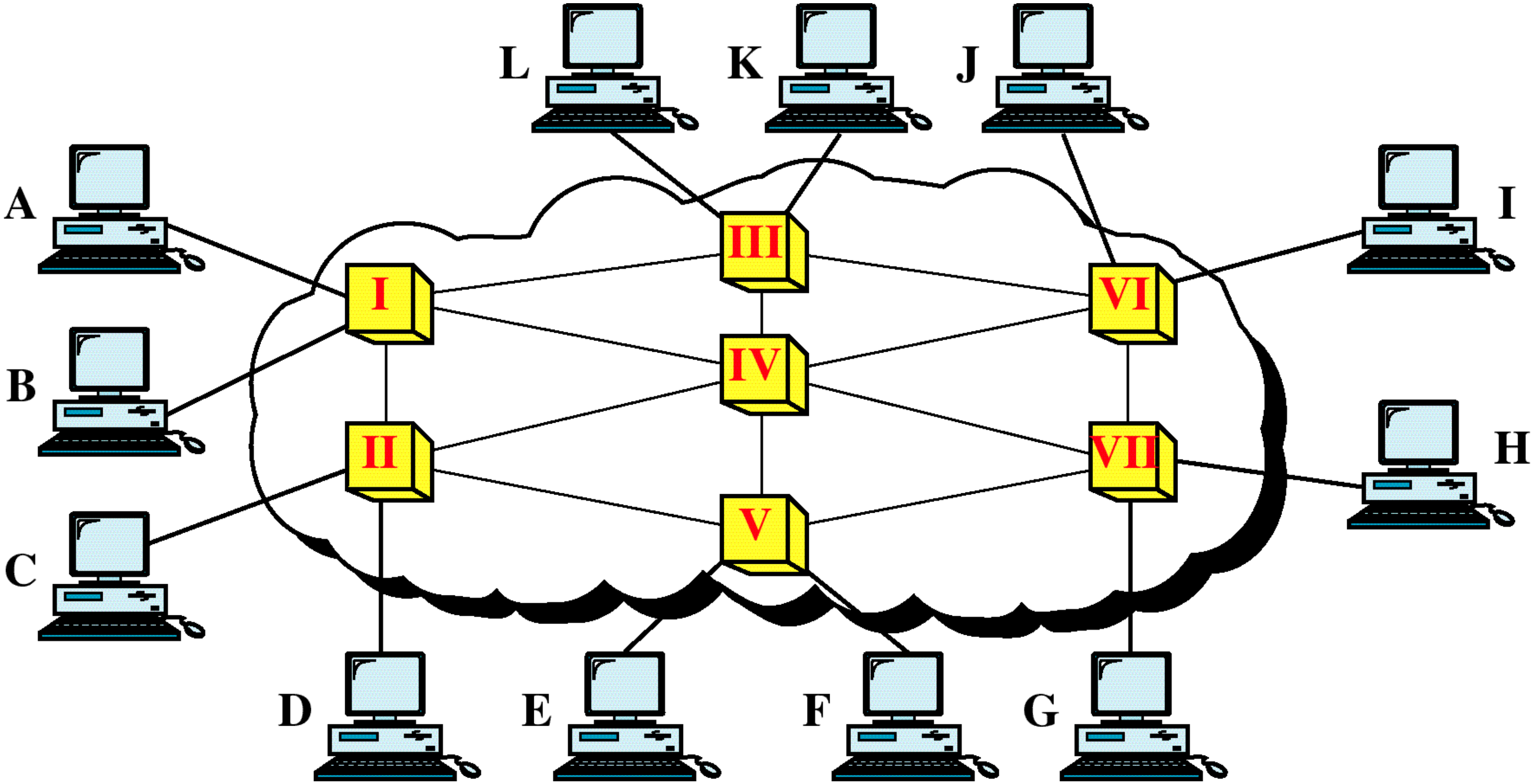
Computer Networks

Switching Technologies

Switched Network

- Long distance transmission typically done over a network of switched nodes
- End devices are stations
  - – Computer, terminal, phone, etc.
- A collection of nodes and connections is a communications network
- Data routed by being switched from node to node

A Switched Network



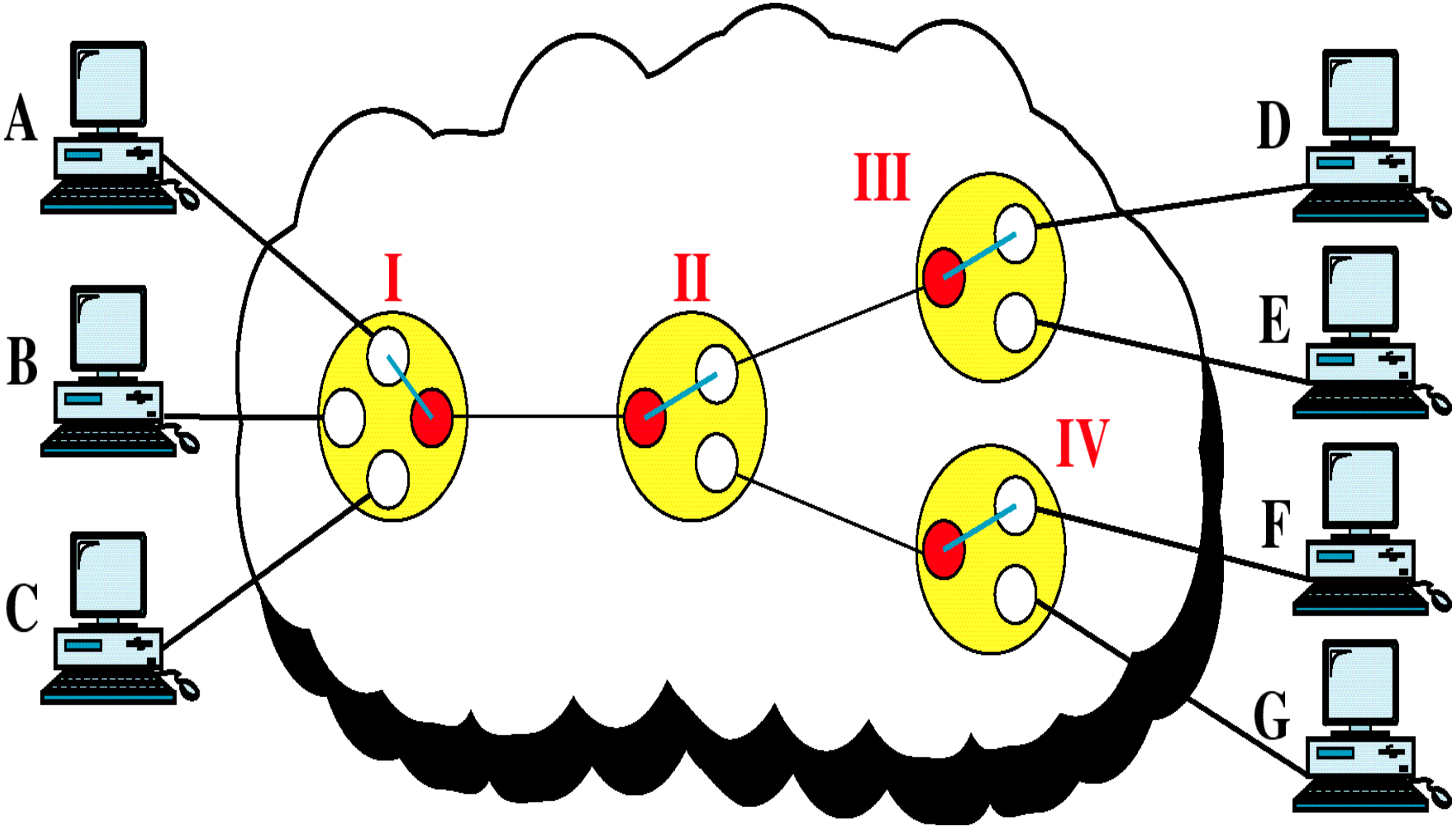
Switching Nodes

- Node to node links usually multiplexed
- Two different switching technologies
  - Circuit switching
  - Packet switching
    - Datagram
    - Virtual Circuits

Circuit Switching

- A Circuit is a dedicated (for the duration of the call) communication path between two stations.
- Circuit switching requires three phases namely ***circuit establishment***, ***data transfer*** and ***circuit termination***
- Must have intelligence for routing
- Once connected, transfer is transparent
- Developed for voice traffic (Telephony)

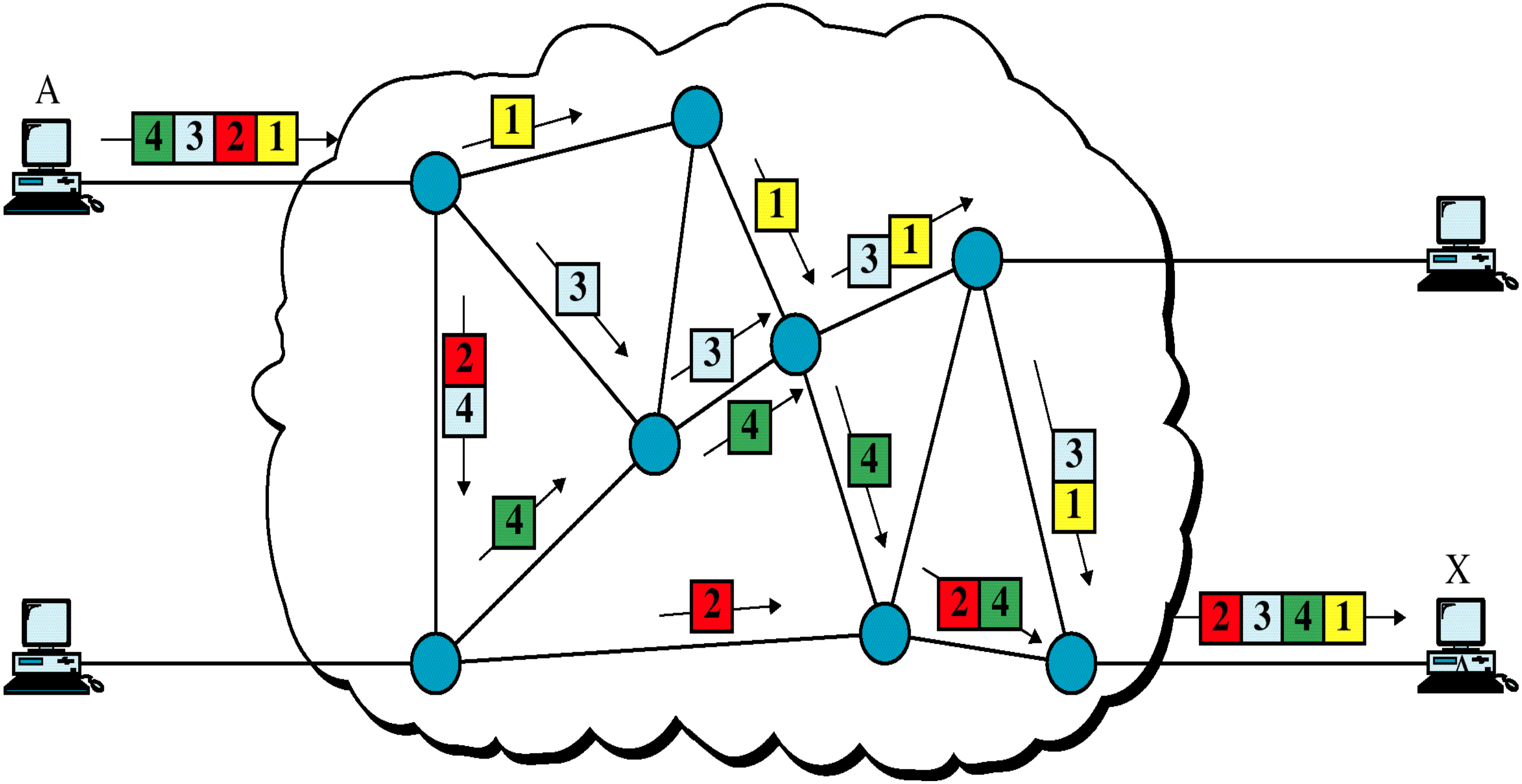
Circuit Switching



Connection-less (Datagram) Packet Switching

- Each packet treated independently
- Packets can take any practical route
- Packets may arrive out of order
- Packets may be lost (dropped)
- Up to receiver to re-order packets

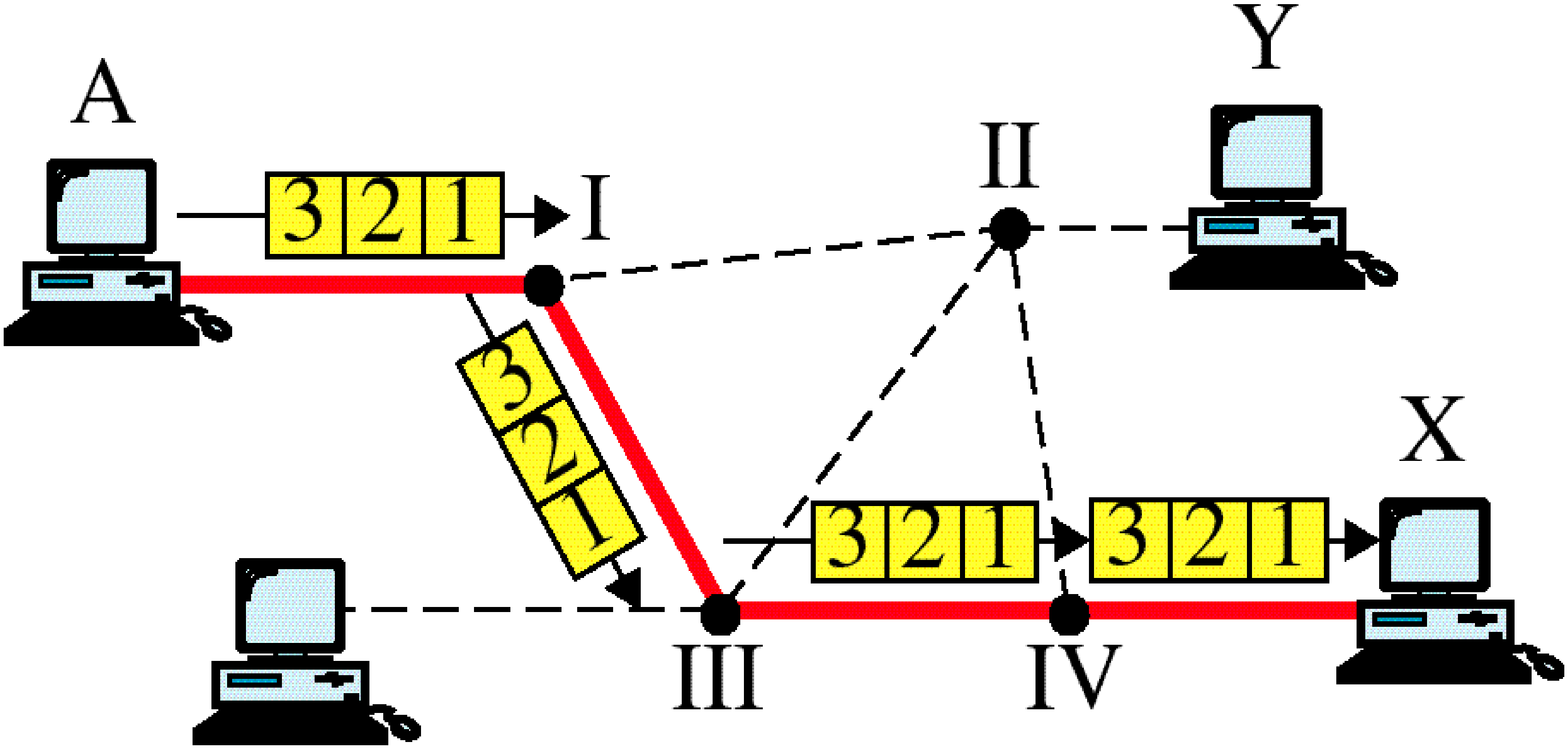
Connection-Less Packet Switching



Virtual Circuit Switching

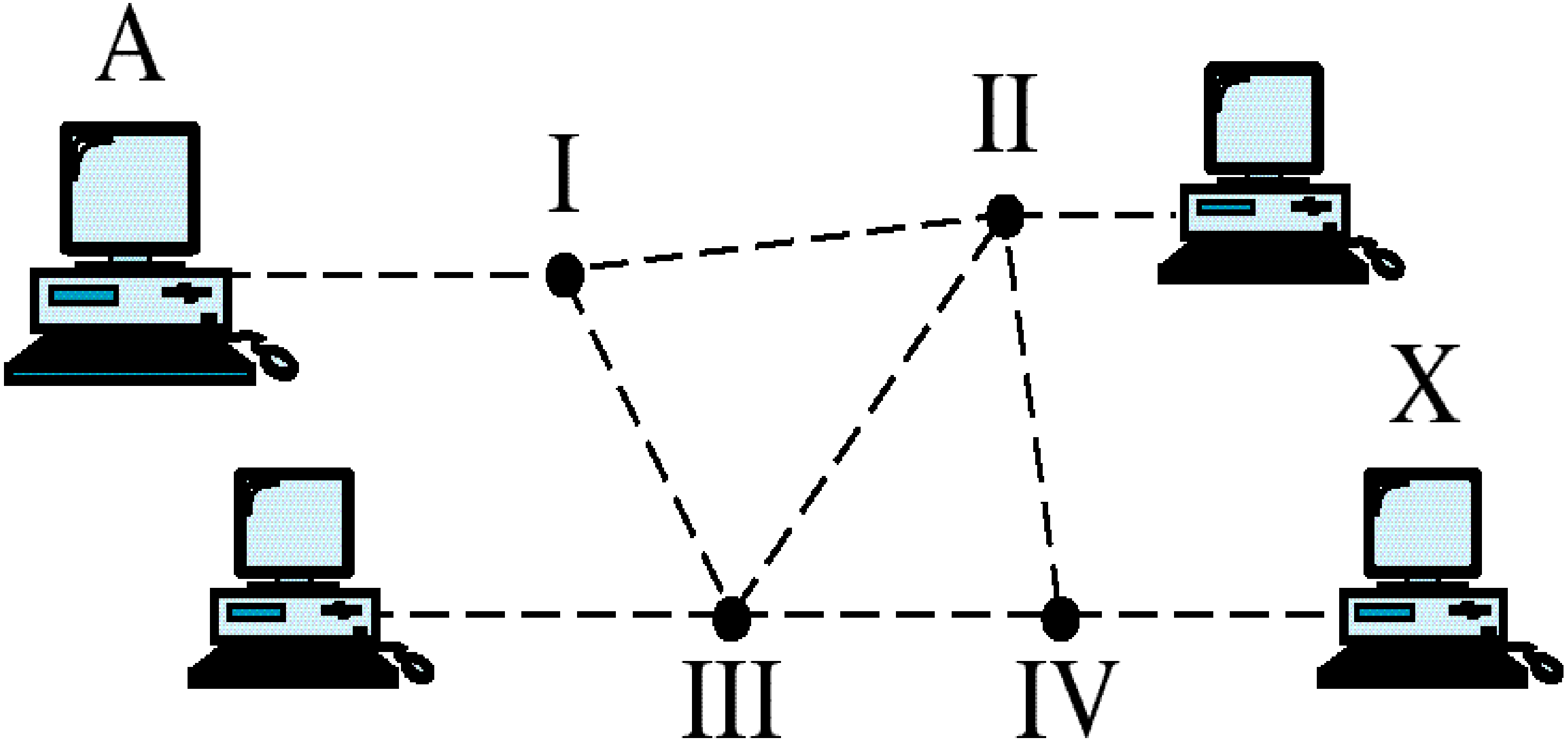
- A virtual connection (not a dedicated path) is established before any packets are sent
- Call request and call accept packets establish connection (handshake)
- Each packet contains a virtual circuit identifier instead of destination address
- No routing decisions required for each packet
- Clear request to drop virtual connection

Switched Virtual Circuit (Cont.)



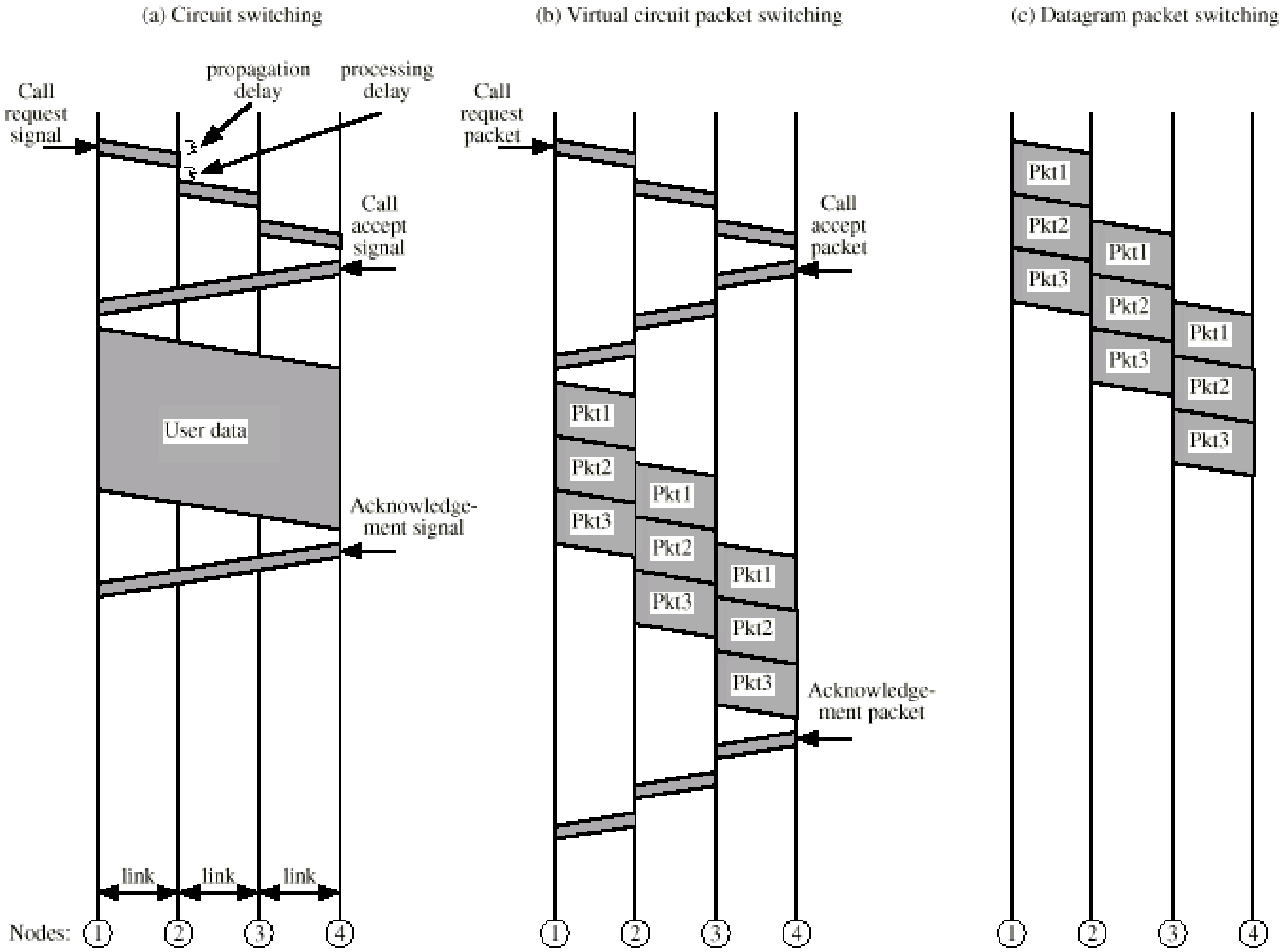
Data transfer

Switched Virtual Circuit (Cont.)



Connection release

Timing Diagrams



Virtual Circuit v.s. Datagram

- Virtual circuits
  - Packets are forwarded more quickly
    - No routing decisions to make
  - Less reliable
    - Loss of a node looses all circuits through that node
- Datagram
  - No call setup phase
    - Better if few packets
  - More flexible
    - Routing can be used to avoid congested parts of the network

Circuit v.s. Packet Switching

Item	*Circuit Switching*	*Packet Switching*
• Dedicated Path	Yes	No
• Bandwidth	Fixed	Dynamic
• Call Setup	Yes	No
• Store & Forward	No	Yes
• Congestion	@ set-up	anytime
• Potentially wasted BW	Yes	No
• Packets follow same route	Yes	Not necessarily

Computer Networks

IP: The Internet Protocol

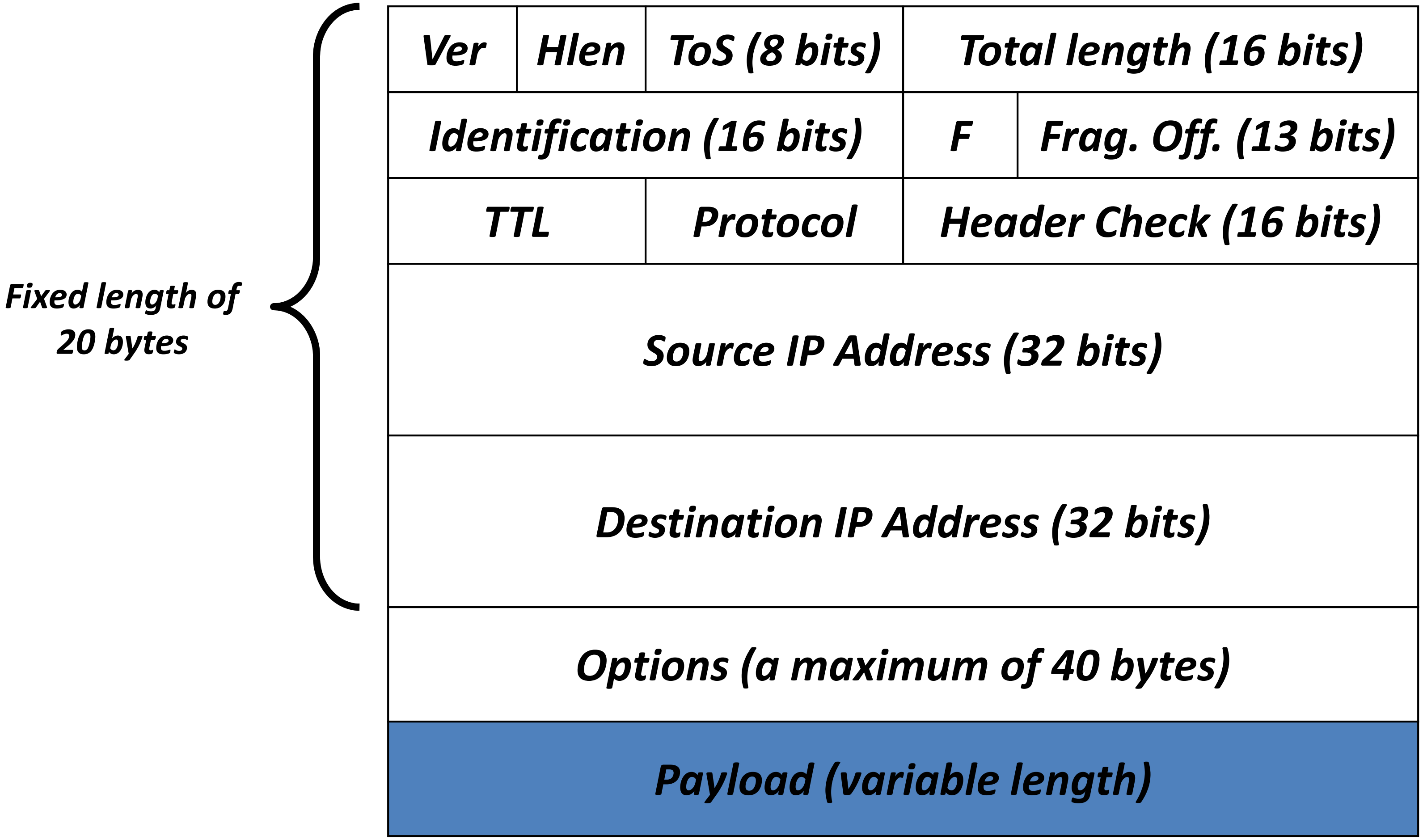
IP: The Internet Protocol

- IP is a ***connection-less***, ***unreliable*** network layer protocol

- IP provides ***best effort*** services in the sense
  - There is no guarantee of delivery of error-free packets
  - There is no guarantee of ordered delivery of packets
  - There is no guarantee of delivery of packets

- IP relies on upper layer transport protocols (TCP) to take care of these problems.

IP Datagram Format



Fixed length of 20 bytes

Ver | *Hlen* | *ToS (8 bits)* | *Total length (16 bits)*

Identification (16 bits) | *F* | *Frag. Off. (13 bits)*

TTL | *Protocol* | *Header Check (16 bits)*

Source IP Address (32 bits)

Destination IP Address (32 bits)

Options (a maximum of 40 bytes)

Payload (variable length)

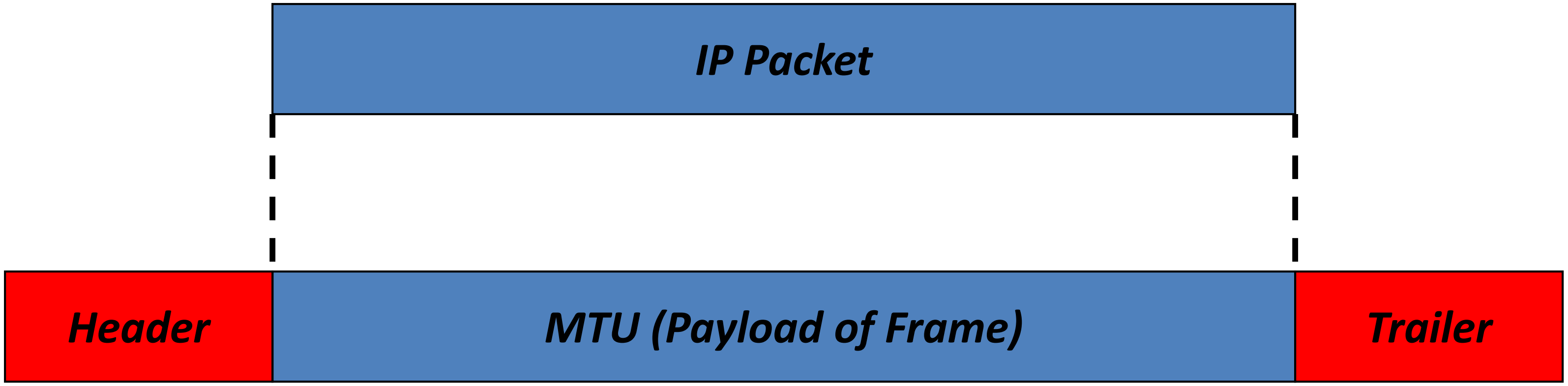
Protocol Field

- The protocol field (8-bits) defines the protocol that is using the services of IP
- It defines the final destination protocol the packet should be delivered to. This is important since several protocols could be multiplexed over IP
  - TCP: 6, UDP: 17

Fragmentation

- IP packet may travel over different networks (LANs and WANs)
- A router de-capsulate an IP packet from the frame it receives, process it, and encapsulate it in another frame
- Frame size and format varies depend on the data link protocol used by the physical network through which the frame is traveling
- MTU (***Maximum Transmission Unit***) is the maximum size of the data field (payload) in the frame
- If Packet size > MTU, Need for Fragmentation

MTU



Protocol	***MTU (Octets)***
Ethernet	***1500***
Token Ring (4 Mbps)	***4464***
Token Ring (16 Mbps)	***17914***

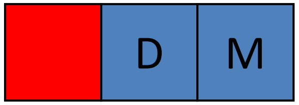
Fragmentation (Cont.)

- Each fragment has its own header (most of fields are copied, some will change, including the total length, the Flags and the fragmentation offset fields)
- A fragmented datagram may itself be fragmented if it encounters a network with smaller MTU.
- A packet can be fragmented by a source host or by any router in the path. Re-assembly of the packet must be done at the destination host.

Fields related to Fragmentation

- ***Identification*** (16 bits): All fragments of a packet has the same ID number which is the same as that of the original packet. The R/x knows that all fragments having the same ID should be assembled into one packet

- ***Flags*** (3 bits):



D: Do not Fragment

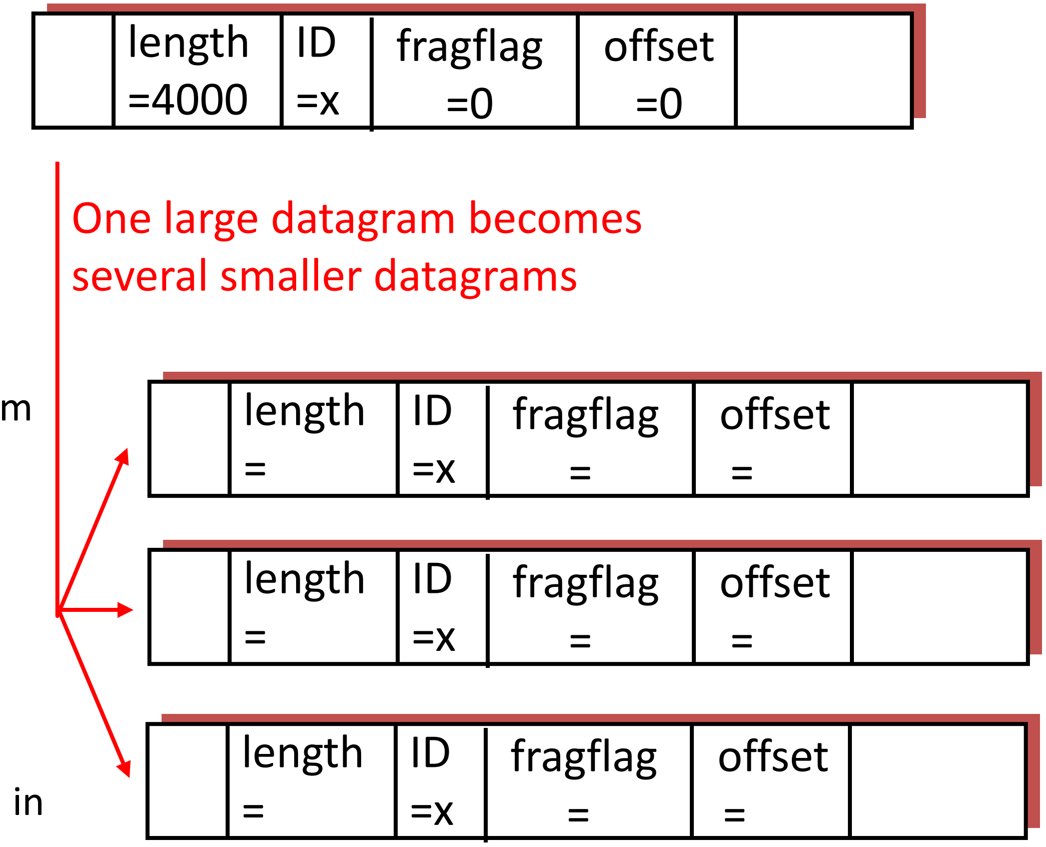
M: More Fragment

- ***Fragmentation Offset*** (13 bits): Relative position of the fragment to the whole packet measured in units of 8 octets

IP Fragmentation: Reassembly

Example

- 4000 byte datagram
  - 3980 bytes of data
  - 20 bytes of IP header
- MTU = 1500 bytes
- Length: length of data in datagram (data + IP header)
- ID: unique identifier, used for reassembly
- fragflag:
  - 0 - last fragment
  - 1 – more fragments to follow
- Offset: Offset relative to location in initial datagram.
  - given in 8-byte chunks.



length =4000 | ID =x | fragflag =0 | offset =0

One large datagram becomes several smaller datagrams

length = | ID =x | fragflag = | offset =

length = | ID =x | fragflag = | offset =

length = | ID =x | fragflag = | offset =

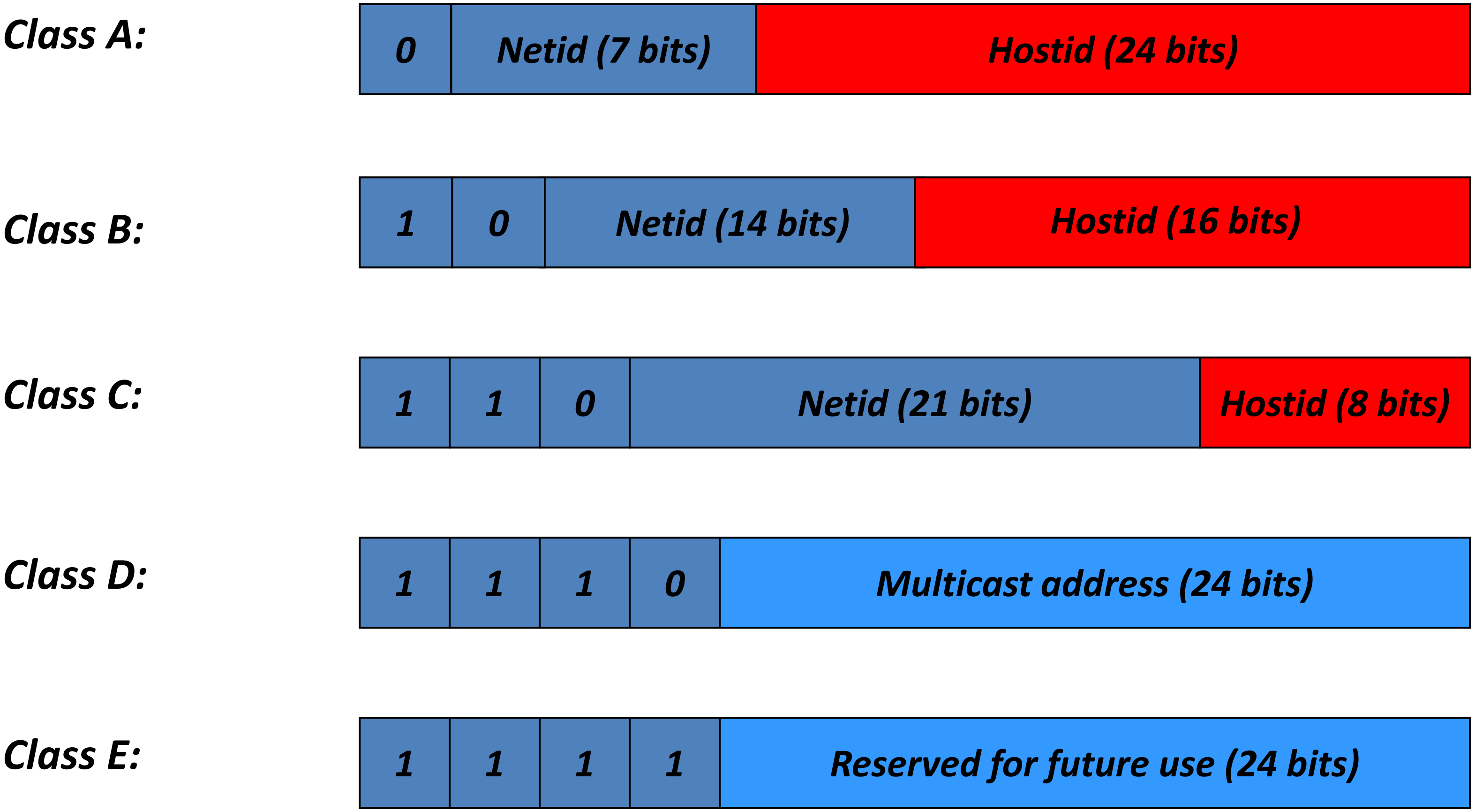
IPv4 Addressing

- The Internet is made of combination of LANs and WANs connected via routers
- A host needs to be able to communicate with another host without worrying about which physical network must be passed through
- ***Hosts*** must therefore be identified ***uniquely*** and ***globally*** at the network layer
- For efficient and optimum routing, ***routers*** must also be identified ***uniquely*** and ***globally*** at the network layer

IP Addressing (Continued)

- IPv4 address is a 32-bit address, implemented in software, is used to uniquely and globally identify a host or a router on the Internet
- A device can have more than one IP address if it is connected to more than one network (***multi-homed***)
- An IP address have two parts, the ***netid*** and the ***hostid***. They have variable lengths depending on the class of the address
- All devices on the same network have the same netid

Classful IP Addressing



Class A: *0* *Netid (7 bits)* *Hostid (24 bits)*

Class B: *1* *0* *Netid (14 bits)* *Hostid (16 bits)*

Class C: *1* *1* *0* *Netid (21 bits)* *Hostid (8 bits)*

Class D: *1* *1* *1* *0* *Multicast address (24 bits)*

Class E: *1* *1* *1* *1* *Reserved for future use (24 bits)*

Decimal to Binary Conversion

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

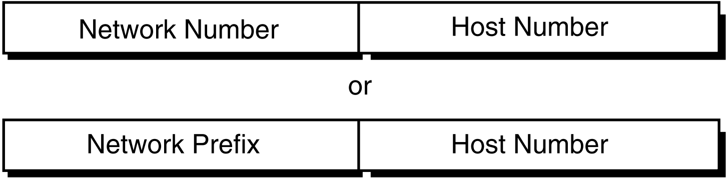
ex: 205 decimal = 1100 1101 binary

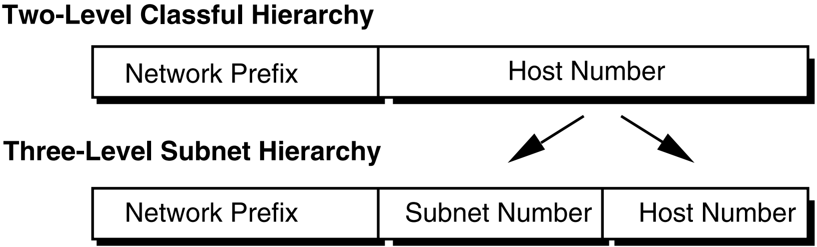
205-128 = 77	-> 1 × 2^7
77 - 64 = 13	-> 1 × 2^6
13 - 32 < 0	-> 0 × 2^5
13 - 16 < 0	-> 0 × 2^4
13 - 8 = 5	-> 1 × 2^3
5 - 4 = 1	-> 1 × 2^2
1 - 2 < 0	-> 0 × 2^1
1 - 1 = 0	-> 1 × 2^0

Classful Addressing (Cont.)

	From	***To***
Class A	***0.0.0.0***	***127.255.255.255***
Class B	***128.0.0.0***	***191.255.255.255***
Class C	***192.0.0.0***	***223.255.255.255***
Class D	***224.0.0.0***	***239.255.255.255***
Class E	***240.0.0.0***	***255.255.255.255***

FIGURE 3. Two-Level Internet Address Structure

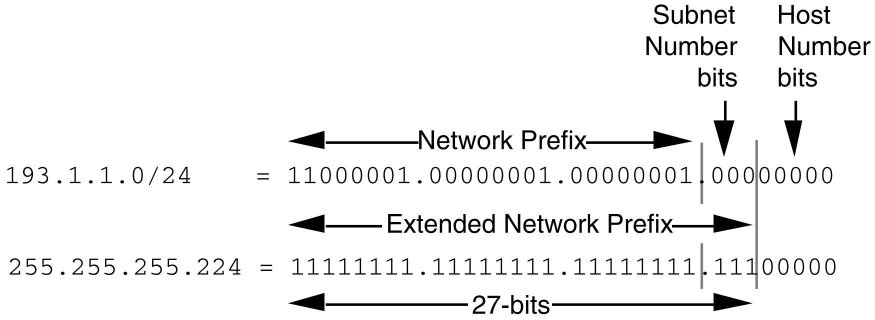



Two-Level Classful Hierarchy
Network Prefix
Host Number
Three-Level Subnet Hierarchy
Network Prefix
Subnet Number
Host Number

Subnet Example #1

Given

An organization is assigned the network number 193.1.1.0/24 and it needs to define six subnets. The largest subnet is required to support 25 hosts.


Subnet Number bits
Host Number bits
Network Prefix
193.1.1.0/24 = 11000001.00000001.00000001.00000000
Extended Network Prefix
255.255.255.224 = 11111111.11111111.11111111.11100000
27-bits

Base Net: 11000001.00000001.00000001 .00000000 = 193.1.1.0/24

Subnet #0: 11000001.00000001.00000001.000 00000 = 193.1.1.0/27

Subnet #1: 11000001.00000001.00000001.001 00000 = 193.1.1.32/27

Subnet #2: 11000001.00000001.00000001.010 00000 = 193.1.1.64/27

Subnet #3: 11000001.00000001.00000001.011 00000 = 193.1.1.96/27

Subnet #4: 11000001.00000001.00000001.100 00000 = 193.1.1.128/27

Subnet #5: 11000001.00000001.00000001.101 00000 = 193.1.1.160/27

Subnet #6: 11000001.00000001.00000001.110 00000 = 193.1.1.192/27

Subnet #7: 11000001.00000001.00000001.111 00000 = 193.1.1.224/27

The valid host addresses for Subnet #2 in this example are listed in the following sample code. The underlined portion of each address identifies the extended network prefix, while the bold digits identify the 5-bit host number field:

```
Subnet #2: 11000001.00000001.00000001.010 00000 = 193.1.1.64/27
Host #1: 11000001.00000001.00000001.010 00001 = 193.1.1.65/27
Host #2: 11000001.00000001.00000001.010 00010 = 193.1.1.66/27
Host #3: 11000001.00000001.00000001.010 00011 = 193.1.1.67/27
Host #4: 11000001.00000001.00000001.010 00100 = 193.1.1.68/27
Host #5: 11000001.00000001.00000001.010 00101 = 193.1.1.69/27
.
.
Host #15: 11000001.00000001.00000001.010 01111 = 193.1.1.79/27
Host #16: 11000001.00000001.00000001.010 10000 = 193.1.1.80/27
.
.
Host #27: 11000001.00000001.00000001.010 11011 = 193.1.1.91/27
Host #28: 11000001.00000001.00000001.010 11100 = 193.1.1.92/27
Host #29: 11000001.00000001.00000001.010 11101 = 193.1.1.93/27
Host #30: 11000001.00000001.00000001.010 11110 = 193.1.1.94/27
```

The valid host addresses for Subnet #6 are listed in the following sample code. The underlined portion of each address identifies the extended network prefix, while the bold digits identify the 5-bit host number field:

```
Subnet #6: 11000001.00000001.00000001.110 00000 = 193.1.1.192/27
Host #1: 11000001.00000001.00000001.110 00001 = 193.1.1.193/27
Host #2: 11000001.00000001.00000001.110 00010 = 193.1.1.194/27
Host #3: 11000001.00000001.00000001.110 00011 = 193.1.1.195/27
Host #4: 11000001.00000001.00000001.110 00100 = 193.1.1.196/27
Host #5: 11000001.00000001.00000001.110 00101 = 193.1.1.197/27
.
.
Host #15: 11000001.00000001.00000001.110 01111 = 193.1.1.207/27
Host #16: 11000001.00000001.00000001.110 10000 = 193.1.1.208/27
.
.
Host #27: 11000001.00000001.00000001.110 11011 = 193.1.1.219/27
Host #28: 11000001.00000001.00000001.110 11100 = 193.1.1.220/27
Host #29: 11000001.00000001.00000001.110 11101 = 193.1.1.221/27
Host #30: 11000001.00000001.00000001.110 11110 = 193.1.1.222/27
```

Defining the Broadcast Address for Each Subnet

The broadcast address for Subnet #2 is the all-1s host address or:

```
11000001.00000001.00000001.010 11111 = 193.1.1.95
```

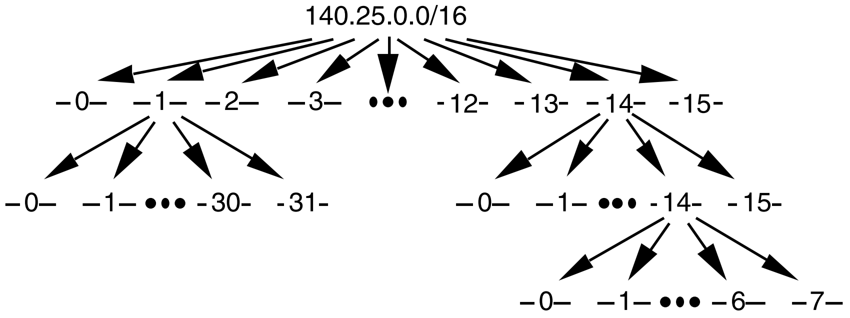
Note that the broadcast address for Subnet #2 is exactly one less than the base address for Subnet #3 (193.1.1.96). This is always the case-the broadcast address for Subnet #n is one less than the base address for Subnet #(n+1).

VLSM Example

Given

An organization has been assigned the network number 140.25.0.0/16 and it plans to deploy VLSM. Figure 21 provides a graphic display of the VLSM design for the organization.

FIGURE 21. Address Strategy for VLSM Example

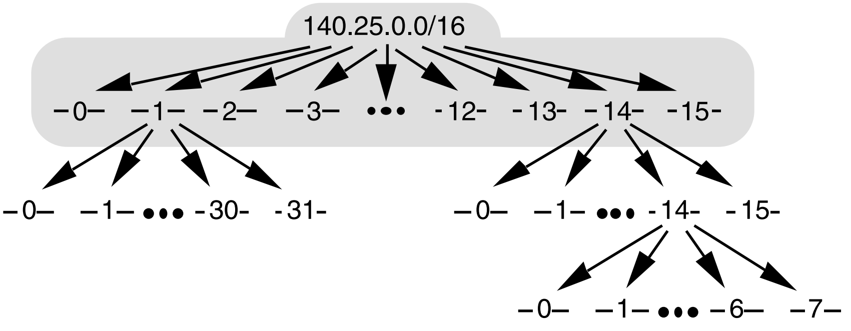


The first step of the subnetting process divides the base network address into 16 equally sized address blocks. Then Subnet #1 is divided into 32 equally sized address blocks and Subnet #14 is divided into 16 equally sized address blocks. Finally, Subnet #14-14 is divided into eight equally sized address blocks.

Define the 16 Subnets of 140.25.0.0/16

The first step in the subnetting process divides the base network address into 16 equally sized address blocks, as illustrated in Figure 22.

FIGURE 22. Sixteen Subnets for 140.25.0.0/16



Since 16 = 24, four bits are required to identify each of the 16 subnets. This means that the organization needs four more bits, or a /20, in the extended network prefix to define the 16 subnets of 140.25.0.0/16. Each of these subnets represents a contiguous block of 212 (or 4,096) network addresses.

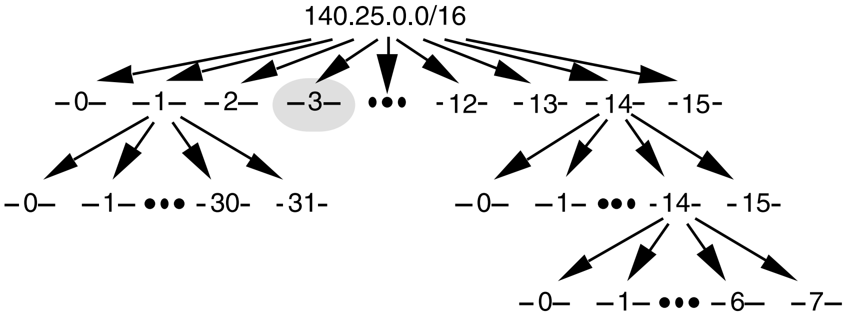
The 16 subnets of the 140.25.0.0/16 address block are listed in the following code sample. The subnets are numbered 0 through 15. The underlined portion of each address identifies the extended network prefix, while the bold digits identify the 4 bits representing the subnet number field:

```
Base Network: 10001100.00011001 .00000000.00000000 = 140.25.0.0/16
Subnet #0: 10001100.00011001.0000 0000.00000000 = 140.25.0.0/20
Subnet #1: 10001100.00011001.0001 0000.00000000 = 140.25.16.0/20
Subnet #2: 10001100.00011001.0010 0000.00000000 = 140.25.32.0/20
Subnet #3: 10001100.00011001.0011 0000.00000000 = 140.25.48.0/20
Subnet #4: 10001100.00011001.0100 0000.00000000 = 140.25.64.0/20
:
:
Subnet #13: 10001100.00011001.1101 0000.00000000 = 140.25.208.0/20
Subnet #14: 10001100.00011001.1110 0000.00000000 = 140.25.224.0/20
Subnet #15: 10001100.00011001.1111 0000.00000000 = 140.25.240.0/20
```

Define the Host Addresses for Subnet #3 (140.25.48.0/20)

Figure 23 shows the host addresses that can be assigned to Subnet #3 (140.25.48.0/20).

FIGURE 23. Host Address for Subnet #3 (140.25.48.0/20)



Since the host number field of Subnet #3 contains 12 bits, there are 4,094 valid host addresses (212 -2) in the address block. The hosts are numbered 1 through 4,094. The valid host addresses for Subnet #3 are listed in the following sample code. The underlined portion of each address identifies the extended network prefix, while the bold digits identify the 12-bit host number field:

```
Subnet #3: 10001100.00011001.0011 0000.00000000 = 140.25.48.0/20
Host #1: 10001100.00011001.0011 0000.00000001 = 140.25.48.1/20
Host #2: 10001100.00011001.0011 0000.00000010 = 140.25.48.2/20
Host #3: 10001100.00011001.0011 0000.00000011 = 140.25.48.3/20
:
:
Host #4093: 10001100.00011001.0011 1111.11111101 = 140.25.63.253/20
Host #4094: 10001100.00011001.0011 1111.11111110 = 140.25.63.254/20
```

The broadcast address for Subnet #3 is the all-1s host address or:

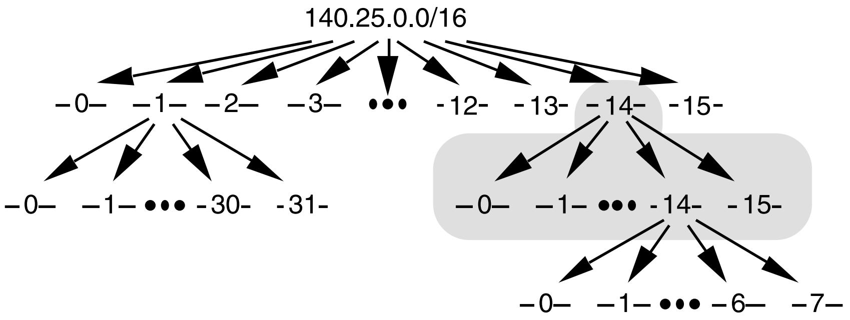
10001100.00011001.0011 1111.11111111 = 140.25.63.255

The broadcast address for Subnet #3 is exactly one less than the base address for Subnet #4 (140.25.64.0).

Define the Sub-Subnets for Subnet #14 (140.25.224.0/20)

After the base network address is divided into 16 subnets, Subnet #14 is subdivided into 16 equally sized address blocks. This division is illustrated in Figure 24.

FIGURE 24. Sub-Subnets for Subnet #14 (140.25.224.0/20)



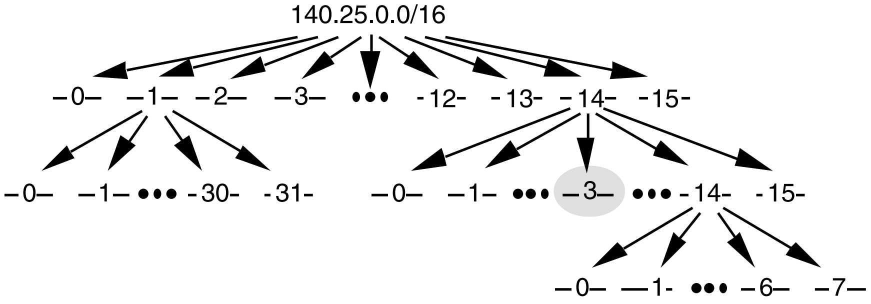
Since 16 = 24, four more bits are required to identify each of the 16 subnets. This means that the organization will need to use a /24 as the extended network prefix length. The 16 subnets of the 140.25.224.0/20 address block are listed in the following sample code. The subnets are numbered 0 through 15. The underlined portion of each sub-subnet address identifies the extended network prefix, while the bold digits identify the 4 bits representing the sub-subnet number field:

Subnet #14: 10001100.00011001.1110 0000.00000000 = 140.25.224.0/20
Subnet #14-0: 10001100.00011001.1110 0000 .00000000 = 140.25.224.0/24
Subnet #14-1: 10001100.00011001.1110 0001 .00000000 = 140.25.225.0/24
Subnet #14-2: 10001100.00011001.1110 0010 .00000000 = 140.25.226.0/24
Subnet #14-3: 10001100.00011001.1110 0011 .00000000 = 140.25.227.0/24
Subnet #14-4: 10001100.00011001.1110 0100 .00000000 = 140.25.228.0/24
.
.
Subnet #14-14: 10001100.00011001.1110 1110 .00000000 = 140.25.238.0/24
Subnet #14-15: 10001100.00011001.1110 1111 .00000000 = 140.25.239.0/24

Define Host Addresses for Subnet #14-3 (140.25.227.0/24)

Figure 25 shows the host addresses that can be assigned to Subnet #14-3 (140.25.227.0/24).

FIGURE 25. Host Addresses for Subnet #14-3 (140.25.227.0/24)



Each of the subnets of Subnet #14-3 has 8 bits in the host number field. This means that each subnet represents a block of 254 valid host addresses (28 -2). The hosts are numbered 1 through 254.

The valid host addresses for Subnet #14-3 are listed in the following sample code. The underlined portion of each address identifies the extended network prefix, while the bold digits identify the 8-bit host number field:

```
Subnet #14 3: 10001100.00011001.11100011 .00000000 = 140.25.227.0/24
Host #1 10001100.00011001.11100011 .00000001 = 140.25.227.1/24
Host #2 10001100.00011001.11100011 .00000010 = 140.25.227.2/24
Host #3 10001100.00011001.11100011 .00000011 = 140.25.227.3/24
Host #4 10001100.00011001.11100011 .00000100 = 140.25.227.4/24
Host #5 10001100.00011001.11100011 .00000101 = 140.25.227.5/24
.
.
Host #253 10001100.00011001.11100011 .11111101 = 140.25.227.253/24
Host #254 10001100.00011001.11100011 .11111110 = 140.25.227.254/24
```

The broadcast address for Subnet #14-3 is the all-1s host address or:

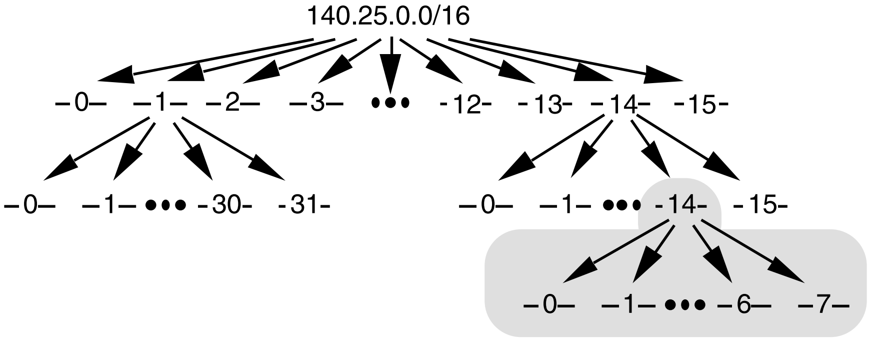
10001100.00011001.11100011. 11111111 = 140.25.227.255

The broadcast address for Subnet #14-3 is exactly one less than the base address for Subnet #14-4 (140.25.228.0).

Define the Sub-Subnets for Subnet #14-14 (140.25.238.0/24)

After Subnet #14 is divided into 16 subnets, Subnet #14-14 is subdivided into eight equally sized address blocks, as shown in Figure 26.

FIGURE 26. Sub-Subnets for Subnet #14-14 (140.25.238.0/24)

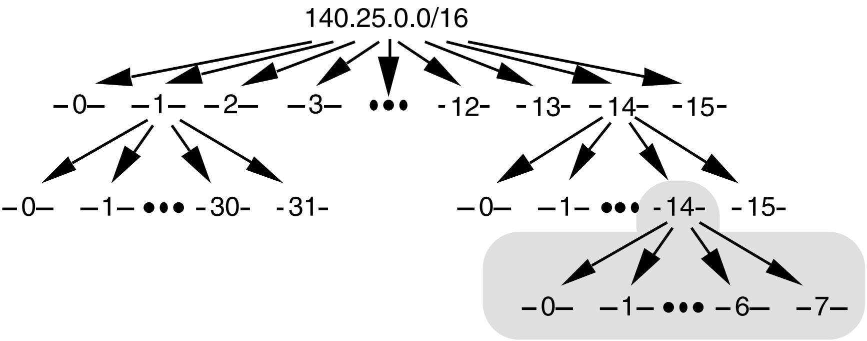


Since 8 = 23, three more bits are required to identify each of the eight subnets. This means that the organization will need to use a /27 as the extended network prefix length.

The eight subnets of the 140.25.238.0/24 address block are listed in the following sample code. The subnets are numbered 0 through 7. The underlined portion of each sub-subnet address identifies the extended network prefix, while the bold digits identify the 3 bits representing the subnet-number field:

Subnet #14-14: 10001100.00011001.11101110 .00000000 = 140.25.238.0/24
Subnet#14-14-0: 10001100.00011001.11101110.000 00000 = 140.25.238.0/27
Subnet#14-14-1: 10001100.00011001.11101110.001 00000 = 140.25.238.32/27
Subnet#14-14-2: 10001100.00011001.11101110.010 00000 = 140.25.238.64/27
Subnet#14-14-3: 10001100.00011001.11101110.011 00000 = 140.25.238.96/27
Subnet#14-14-4: 10001100.00011001.11101110.100 00000 = 140.25.238.128/27
Subnet#14-14-5: 10001100.00011001.11101110.101 00000 = 140.25.238.160/27
Subnet#14-14-6: 10001100.00011001.11101110.110 00000 = 140.25.238.192/27
Subnet#14-14-7: 10001100.00011001.11101110.111 00000 = 140.25.238.224/27

FIGURE 27. Host Addresses for Subnet #14-14-2 (140.25.238.64/27)



Each of the subnets of Subnet #14-14 has 5 bits in the host number field. This means that each subnet represents a block of 30 valid host addresses (25 -2). The hosts will be numbered 1 through 30.

The valid host addresses for Subnet #14-14-2 are listed in the following sample code. The underlined portion of each address identifies the extended network prefix, while the bold digits identify the 5-bit host number field:

Subnet#14-14-2: 10001100.00011001.11101110.010 00000 = 140.25.238.64/27
Host #1 10001100.00011001.11101110.010 00001 = 140.25.238.65/27
Host #2 10001100.00011001.11101110.010 00010 = 140.25.238.66/27
Host #3 10001100.00011001.11101110.010 00011 = 140.25.238.67/27
Host #4 10001100.00011001.11101110.010 00100 = 140.25.238.68/27
Host #5 10001100.00011001.11101110.010 00101 = 140.25.238.69/27
.
.
Host #29 10001100.00011001.11101110.010 11101 = 140.25.238.93/27
Host #30 10001100.00011001.11101110.010 11110 = 140.25.238.94/27

The broadcast address for Subnet #14-14-2 is the all-1s host address or:

10001100.00011001.11011100.010 11111 = 140.25.238.95

The broadcast address for Subnet #6-14-2 is exactly one less than the base address for Subnet #14-14-3 (140.25.238.96).

CIDR Address Allocation Example

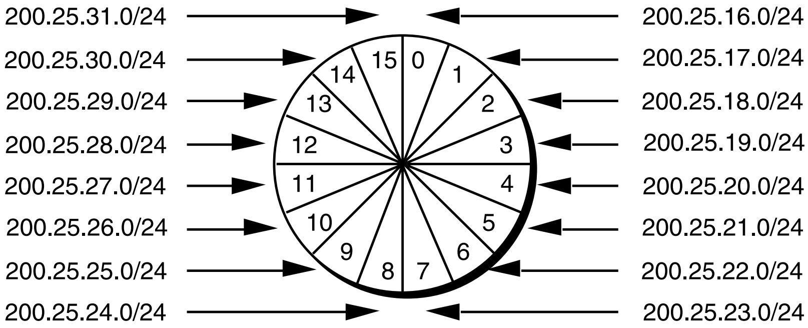
For this example, assume that an ISP owns the address block 200.25.0.0/16. This block represents 65,536 (216) IP addresses (or 256 /24s).

The ISP wants to allocate the smaller 200.25.16.0/20 address block, which represents 4,096 (212) IP addresses (or 16 /24s).

Address Block 11001000.00011001.00010000.00000000 200.25.16.0/20

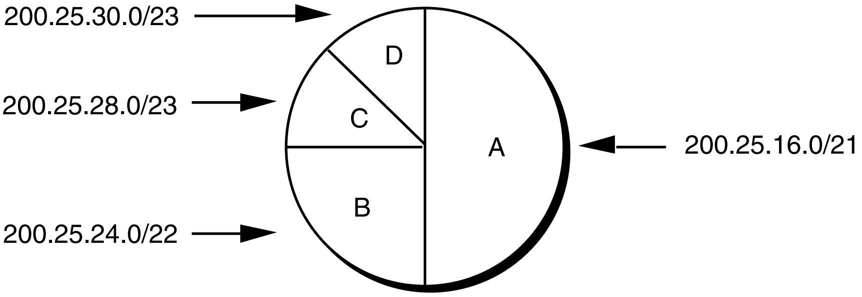
In a classful environment, the ISP is forced to use the /20 as 16 individual /24s.

FIGURE 30. Slicing the Pie-Classful Enviornment



However, in a classless environment, the ISP is free to cut up the pie any way it wants. It could slice the original pie into pieces (each one-half of the address space) and assign one portion to Organization A, then cut the other half into two pieces (each one-fourth of the address space) and assign one piece to Organization B, and then slice the remaining fourth into two pieces (each one-eighth of the address space) and assign them to Organization C and Organization D. Each of the organizations is free to allocate the address space within its “Intranetwork” as desired. This example is illustrated in Figure 31.

FIGURE 31. Slicing the Pie-Classless Enviornment



The following steps explain how to assign addresses with classless interdomain routing.

Step #1: Divide the address block 200.25.16.0/20 into two equally sized slices. Each block represents one-half of the address space, or 2,048 (211) IP addresses.

ISP’s Block 11001000.00011001.00010000.00000000 200.25.16.0/20
Org A: 11001000.00011001.00010000.00000000 200.25.16.0/21
Reserved: 11001000.00011001.00011000.00000000 200.25.24.0/21

Step #2: Divide the reserved block (200.25.24.0/21) into two equally sized slices. Each block represents one-fourth of the address space, or 1,024 (210) IP addresses.

Reserved 11001000.00011001.00011000.00000000 200.25.24.0/21
Org B: 11001000.00011001.00011000.00000000 200.25.24.0/22
Reserved 11001000.00011001.00011100.00000000 200.25.28.0/22

Step #3: Divide the reserved address block (200.25.28.0/22) into two equally sized blocks. Each block represents one-eighth of the address space, or 512 (29) IP addresses.

Reserved 11001000.00011001.00011100.00000000 200.25.28.0/22
Org C: 11001000.00011001.00011100.00000000 200.25.28.0/23
Org D: 11001000.00011001.00011110.00000000 200.25.30.0/23

Comparing CIDR to VLSM

CIDR and VLSM both allow a portion of the IP address space to be recursively divided into subsequently smaller pieces. The difference is that with VLSM, the recursion is performed on the address space previously assigned to an organization and is invisible to the global Internet. CIDR, on the other hand, permits the recursive allocation of an address block by an Internet Registry to a high-level ISP, a mid-level ISP, a low-level ISP, and a private organization’s network.

Like VLSM, the successful deployment of CIDR has three prerequisites:

- The routing protocols must carry network prefix information with each route advertisement.
- All routers must implement a consistent forwarding algorithm based on the longest match.
- For route aggregation to occur, addresses must be assigned so that they are topologically significant.

Controlling the Growth of Internet’s Routing Tables

CIDR helps control the growth of the Internet’s routing tables by reducing the amount of routing information. This process requires that the Internet be divided into addressing domains. Within a domain, detailed information is available about all of the networks that reside in the domain. Outside of an addressing domain, only the common network prefix is advertised. This allows a single routing table entry to specify a route to many individual network addresses.

FIGURE 32. Reduced Size of Internet Routing Tables

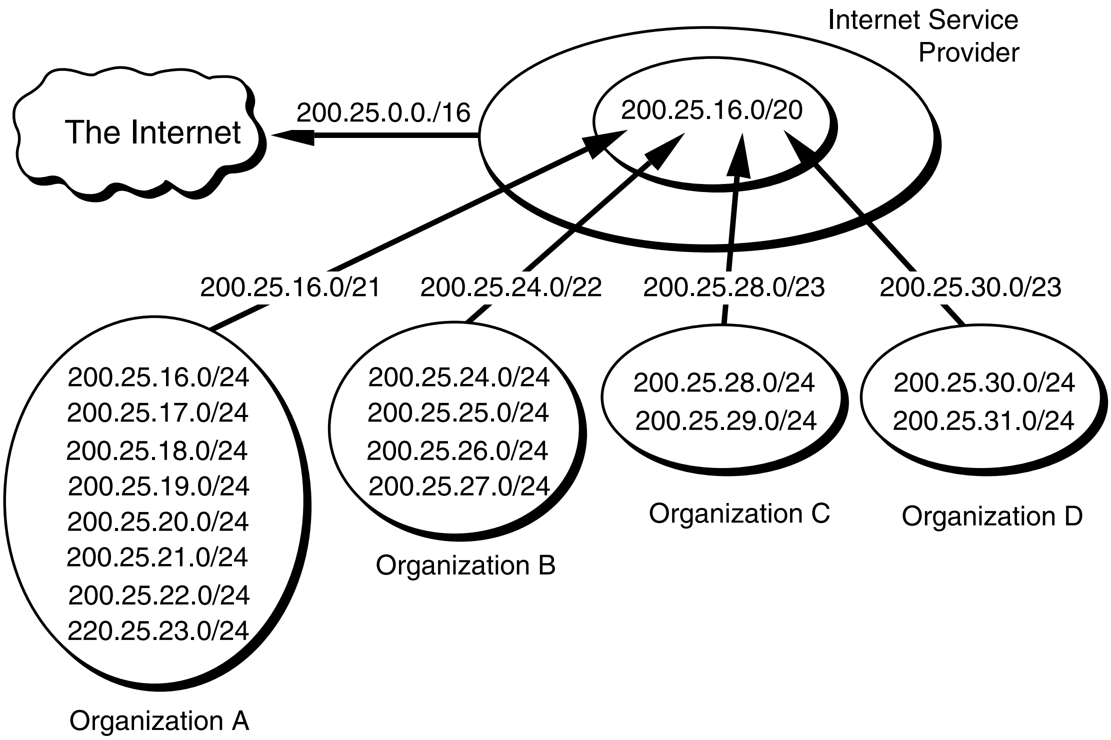


Figure 32 illustrates how the allocation described in the previous CIDR example helps reduce the size of the Internet routing tables. Assume that a portion of the ISP's address block (200.25.16.0/20) has been allocated as described in the previous example:

- Organization A aggregates eight /24s into a single advertisement (200.25.16.0/21)
- Organization B aggregates four /24s into a single advertisement (200.25.24.0/22)
- Organization C aggregates two /24s into a single advertisement (200.25.28.0/23)
- Organization D aggregates two /24s into a single advertisement (200.25.30.0/23)

Then the ISP can inject the 256 /24s in its allocation into the Internet with a single advertisement-200.25.0.0/16.

Note that route aggregation by means of BGP-4 (the protocol that allows CIDR aggregation) is not automatic. The network engineers must configure each router to perform the required aggregation. The successful deployment of CIDR allows the number of individual networks on the Internet to expand while minimizing the number of routes in the Internet routing tables.

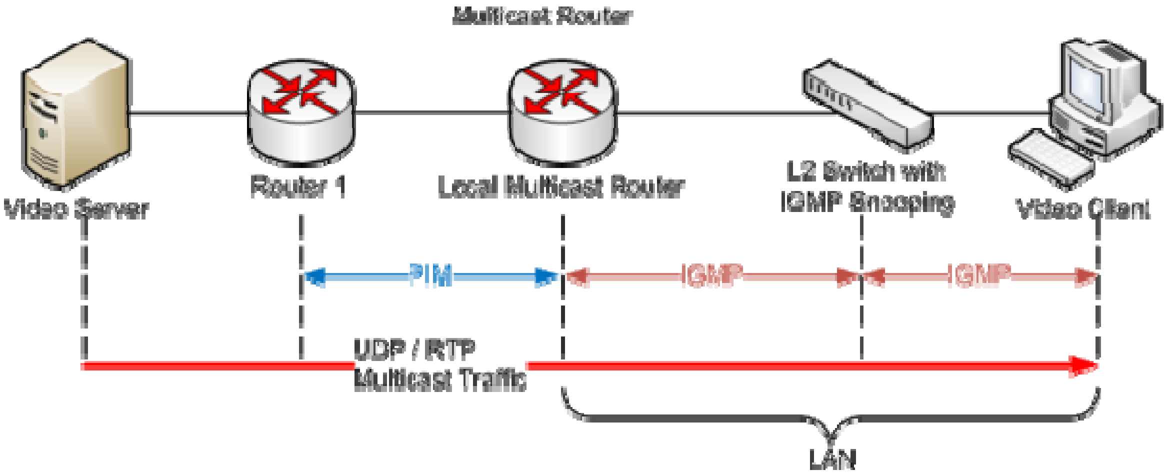
Internet Control Message Protocol (ICMP) and Internet Group Management Protocol (IGMP)

Internet Control Message Protocol (ICMP):

- ICMP (RFC 792) is used by hosts and routers to pass network-layer necessary information
  - Error reporting
  - Router signaling
- One of the mechanisms to ensure Internet Protocol runs error-free since IP provides no guarantee of datagram delivery.
- Example:
  - "Destination network unreachable" is a message sent when a host with a certain IP address cannot be found.
  - This message originated from a router and is sent when it was not able to find a path to the host.
- ICMP defines five error messages
  - Destination Unreachable
  - Parameter Problems
  - Redirect
  - Source Quench
  - Time Exceeded
- ICMP also supports informational message:
- Echo Request/Echo Reply
- ICMP is a useful protocol, however only few network application make use of it since its functionality is limited to diagnostic and error notification
- Famous application is ping which is used to determine if host is alive or inaccessible and the delay between sending a packet and receiving a response

Internet Group Management Protocol (IGMP):

- The **Internet Group Management Protocol** (**IGMP**) is a communications protocol used by hosts and adjacent routers on IP networks to establish multicast group memberships. IGMP is an integral part of IP multicast.



- IGMP operates between the client computer and a local multicast router. Switches featuring IGMP snooping derive useful information by observing these IGMP transactions. Protocol Independent Multicast (PIM) is then used between the local and remote multicast routers, to direct multicast traffic from the multicast server to many multicast clients.

- **IGMP snooping** is the process of listening to Internet Group Management Protocol (IGMP) network traffic. The feature allows a network switch to listen in on the IGMP conversation between hosts and routers. By listening to these conversations the switch maintains a map of which links need which IP multicast streams. Multicasts may be filtered from the links which do not need them and thus controls which ports receive specific multicast traffic.

Notes:

- A Request for Comments (RFC) is a formal document from the Internet Engineering Task Force (IETF) that is the result of committee drafting and subsequent review by interested parties.

- Some RFCs are informational in nature. Of those that are intended to become Internet standards, the final version of the RFC becomes the standard and no further comments or changes are permitted.

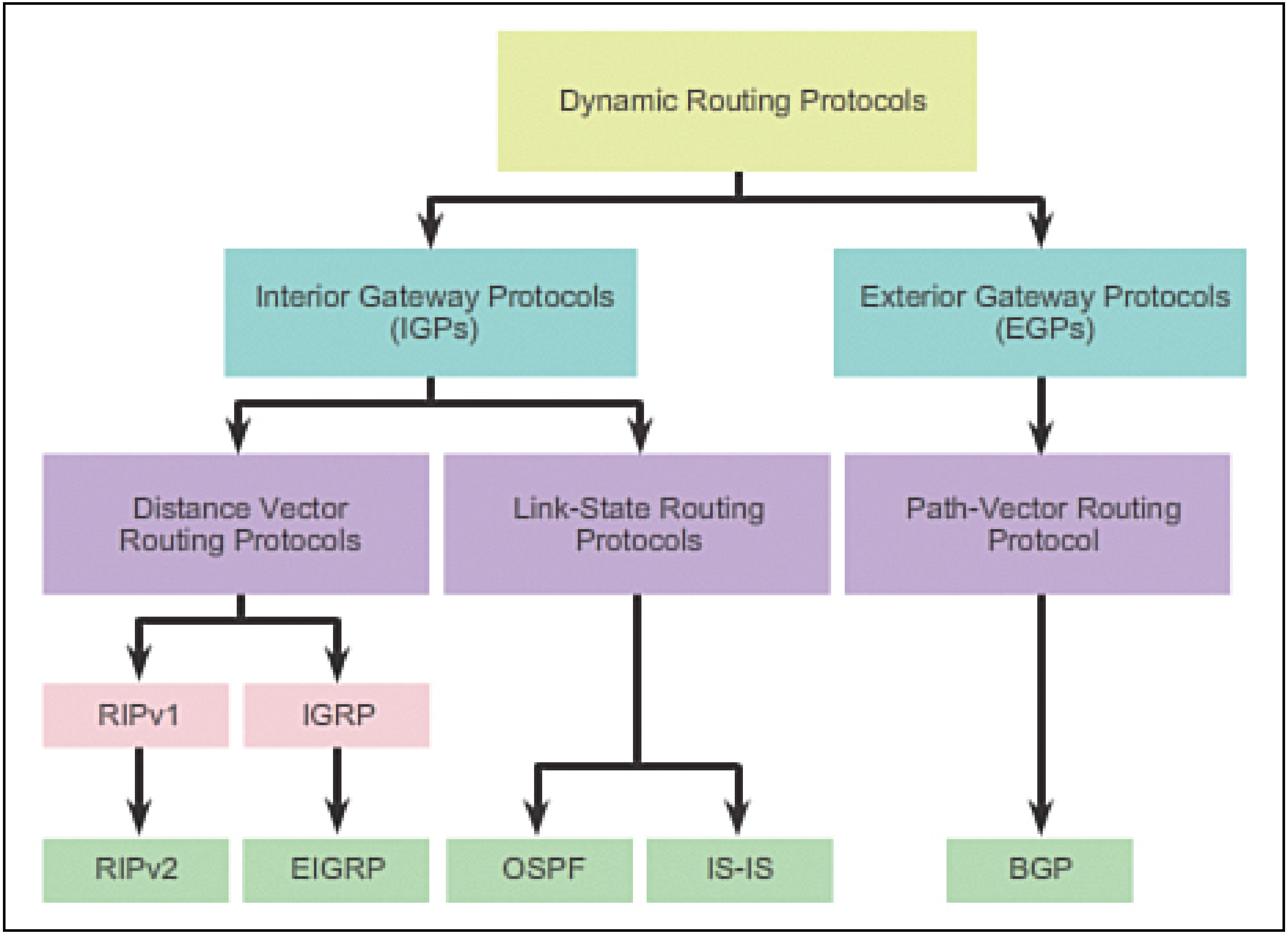
Computer Networks

Routing Algorithms

Static v.s. Dynamic Routing (Basics)

- Static Routing Tables are entered manually
- Strengths of Static Routing
  - Ease of use
  - Control
  - Efficiency
- Weaknesses of Static Routing
  - Not Scalable
  - Not adaptable to link failures

- Dynamic Routing Tables are created through the exchange of information between routers on the availability and status of the networks to which an individual router is connected to. Two Types
  - Distance Vector Protocols
    - ***RIP: Routing Information Protocol***
  - Link State Protocols
    - ***OSPF: Open Shortest Path First***


Dynamic Routing Protocols

Interior Gateway Protocols (IGPs)

Exterior Gateway Protocols (EGPs)

Distance Vector Routing Protocols

Link-State Routing Protocols

Path-Vector Routing Protocol

RIPv1

IGRP

RIPv2

EIGRP

OSPF

IS-IS

BGP

Choosing the Right Protocol

- Interior Routing Protocols
  - Used within an autonomous system
  - Used within an area of administrative control
- Exterior Routing Protocols
  - Used between autonomous systems
  - Used to peer with networks in which you have no administrative control

Choosing the Right Protocol (Cont.)

- Interior Routing Protocols
  - Static
  - RIP
  - OSPF
  - EIGRP
  - ISIS
- Exterior Routing Protocols
  - BGP

NOTE: This is not an exhaustive list of protocols available but merely a list of those commonly used.

Choosing the Right Protocol (Cont.)

- Static Routing
  - May be suitable on small networks
  - Administration intensive as changes have to be made on each router

Choosing the Right Protocol (Cont.)

- Dynamic Routing Protocol Types
  - Distance Vector
    - Routing Information Protocol(RIP)
    - Interior Gateway Routing Protocol(IGRP)
    - Enhanced Interior Gateway Routing Protocol(EIGRP)
  - Link State
    - Open Shortest Path First(OSPF)
    - Intermediate System to Intermediate System(ISIS)
  - Path Vector
    - Border Gateway Protocol(BGP)

Choosing the Right Protocol (Cont.)

- Routing Information Protocol(RIP)
  - RFC 1058(RIPv1), 1988
    - Classful, no support for VLSM
    - No support for authentication
  - RFC 2453(RIPv2), 1998
    - Classless, support for CIDR
    - Support for authentication
  - Uses hop count as routing metric
  - Slow to converge
  - Not very scalable
    - Limited to 15 hops

Choosing the Right Protocol (Cont.)

- Interior Gateway Routing Protocol(IGRP)
    - Invented by Cisco to overcome limitations of RIP
    - Allows for hop count up to 255
    - Allows for multiple route metrics
        - Bandwidth
        - Delay
        - Load
        - Reliability
    - Classful, no support for VLSM

Choosing the Right Protocol

- Enhanced Interior Gateway Routing Protocol(EIGRP)
  - Replaced IGRP
  - Maintains a Topology table
    - Successors, feasible successors
  - Allows for multiple route metrics
  - Classless, support for CIDR
  - Very fast to converge
  - Maintains neighbor relationships
  - Diffusing Update Algorithm (DUAL)
  - Not as CPU intensive as OSPF

Notes

- **Routing Information Protocol (RIP)**
  - A distance vector protocol that has 2 versions
  - RIPv1 – a classful routing protocol
  - RIPv2 - a classless routing protocol
- **Enhanced Interior Gateway Routing Protocol (EIGRP)**
  - A distance vector routing protocols that has some features of link state routing protocols
  - A Cisco proprietary routing protocol

Choosing the Right Protocol (Cont.)

- Open Shortest Path First(OSPF)
  - RFC 2328(OSPFv2), 1998
  - Maintains neighbor relationships
  - Concept of Areas
    - Different areas can be used to control flooding of routing information
  - Classless, supports VLSM
  - Fast to converge
  - CPU Intensive Dijkstra Algorithm
  - Designing can be complicated

Choosing the Right Protocol (Cont.)

- Intermediate System to Intermediate System(ISIS)
  - RFC 1142, 1990
  - Dijkstra Algorithm
  - Mainly used by large service providers
  - Does not use IP to carry routing information
    - Uses ISO addresses
  - Level Concept
    - Level 1 or Intra Area
    - Level 2 or Inter Area
    - Level 1/2 or Both
  - Classless, supports VLSM

Choosing the Right Protocol (Cont.)

- Border Gateway Protocol(BGP)
    - RFC 4271(BGPv4), 2006
    - Peers manually defined
    - Used typically for multi-homing to ISP(s)
    - Very scalable
    - Makes decisions based upon AS Path
    - Lots of policy options
    - Very granular control

IP Packet Delivery

- Two Processes are required to accomplish IP packet delivery:
  - **Routing (**Establish end-to-end paths)
    - discovering and selecting the path to the destination
    - layer-3 functionality
  - **Forwarding**
    - determine next hop

Routing Tables

- Routing Tables are built up by the routing algorithms with components:
  - ***Destination Network Address***: The network portion of the IP address for the destination network
  - ***Subnet Mask***: used to distinguish the network address from the host address
  - ***The IP address of the next hop*** to which the interface forwards the IP packet for delivery
  - The ***Interface*** with which the route is associated

Forwarding Tables

After the routing lookup is completed and the next hop is determined, The IP packet is forwarded according to:

- ***Local delivery model***
  - destination and host are on the same local network
- ***Remote delivery model***
  - destination and host are on different networks

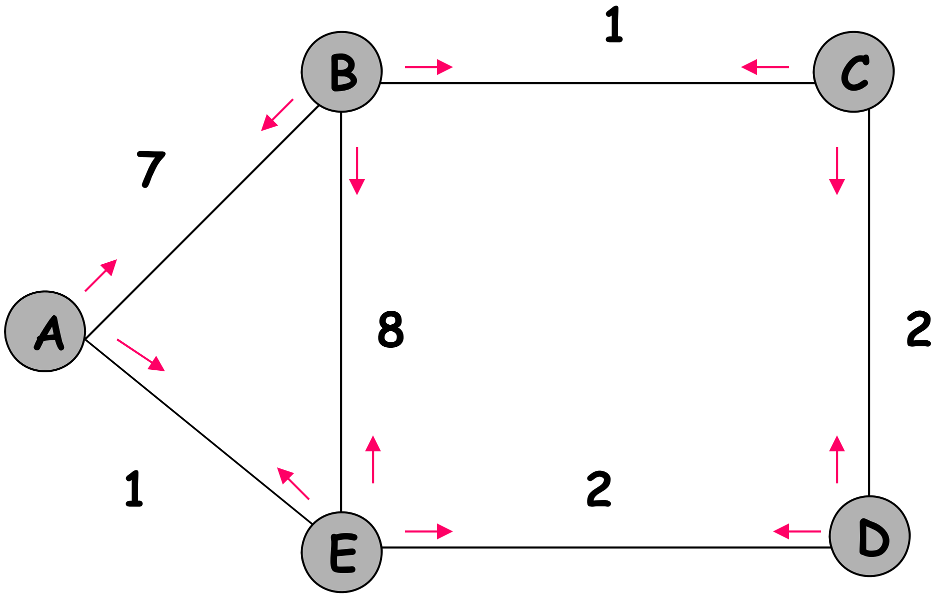
Routing Metrics

- Used by dynamic routing protocols to establish preference for a particular route.
- Support ***Route Diversity*** and ***Load Balancing***
- Most Common routing metrics:
  - ***Hop Count*** (minimum # of hops)
  - ***Bandwidth/Throughput*** (maximum throughput)
  - ***Load*** (actual usage)
  - ***Delay*** (shortest delay)
  - ***Cost***

Dynamic Routing Protocols (1)

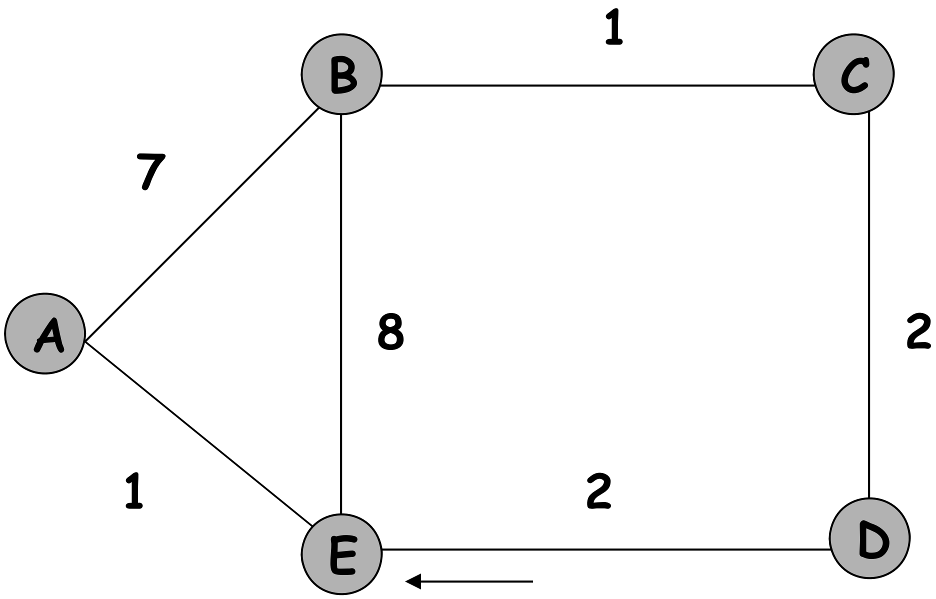
- ***Distance Vector (DV) Protocols***
  - based on the Bellman-Ford algorithm
  - Each router on the network compiles a list of the networks it can reach (in the form of a distance vector)
  - exchange this list with its ***neighboring routers*** only
  - Upon receiving vectors from each of its neighbors, the router computes its own ***distance*** to each neighbor.
  - for every network X, router finds that neighbor who is closer to X than to any other neighbor. Router updates its cost to X.

Example: Initial Distances



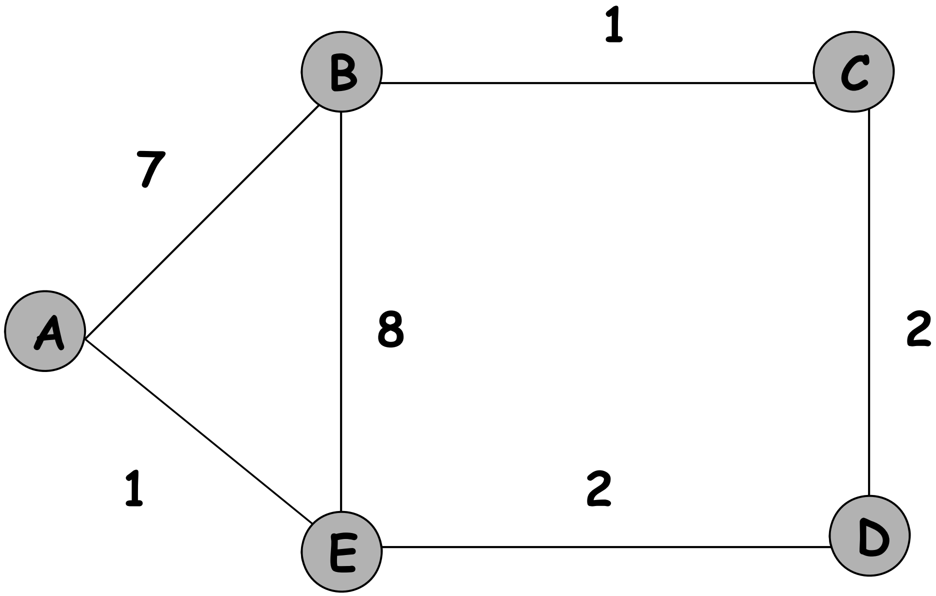
Info at node	Distance to node A	B	C	D	E
A	0	7	~	~	1
B	7	0	1	~	8
C	~	1	0	2	~
D	~	~	2	0	2
E	1	8	~	2	0

Router E receives Router D Table



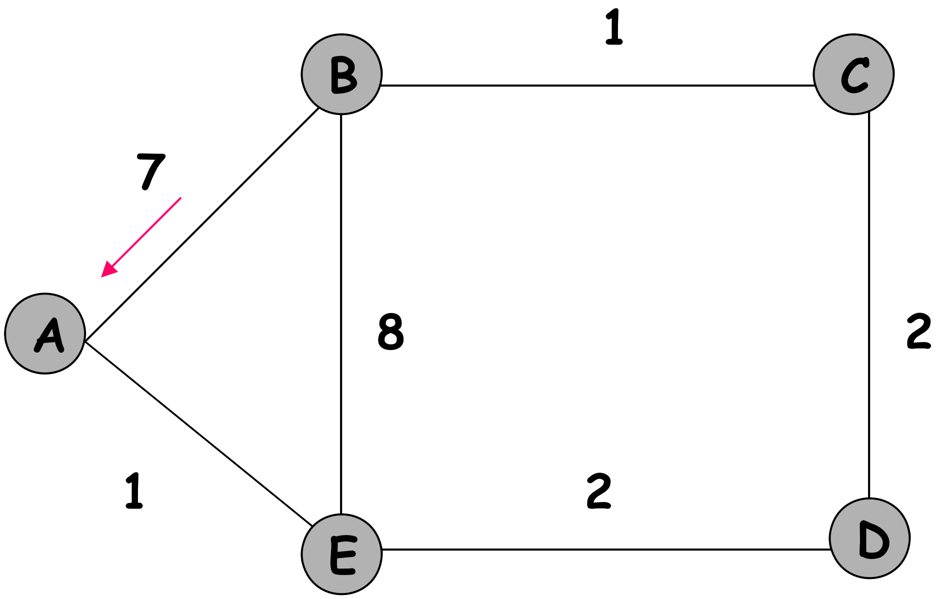
Info at node	Distance to node A	B	C	D	E
A	0	7	~	~	1
B	7	0	1	~	8
C	~	1	0	2	~
D	~	~	2	0	2
E	1	8	~	2	0

Router E updates cost to Router C



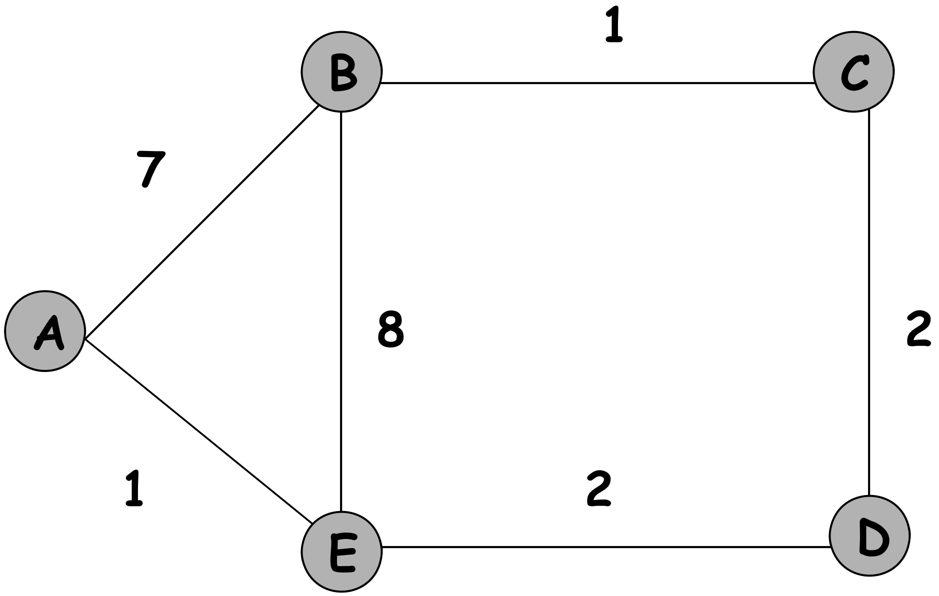
Info at node	Distance to node				
	A	B	C	D	E
A	0	7	~	~	1
B	7	0	1	~	8
C	~	1	0	2	~
D	~	~	2	0	2
E	1	8	4	2	0

Router A receives Router B Table



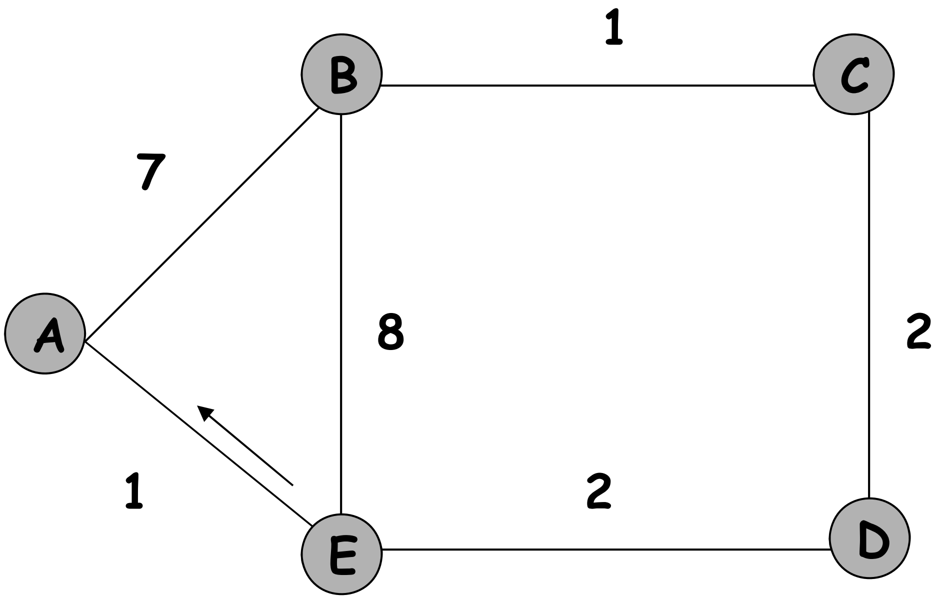
Info at node	Distance to node				
	A	B	C	D	E
A	0	7	~	~	1
B	7	0	1	~	8
C	~	1	0	2	~
D	~	~	2	0	2
E	1	8	4	2	0

Router A updates Cost to Router C



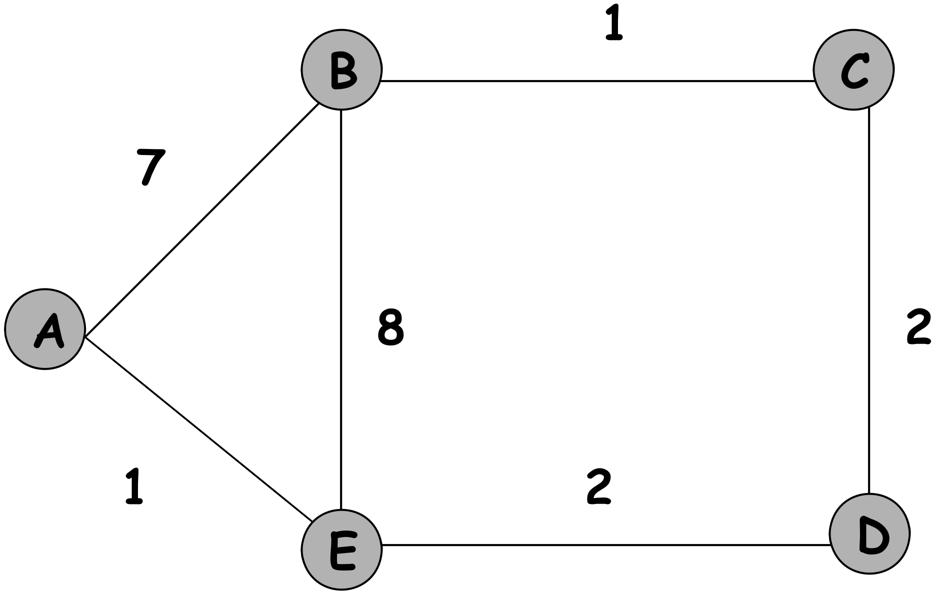
Info at node	Distance to node				
	A	B	C	D	E
A	0	7	8	~	1
B	7	0	1	~	8
C	~	1	0	2	~
D	~	~	2	0	2
E	1	8	4	2	0

Router A receives Router E Table



Info at node	Distance to node A	B	C	D	E
A	0	7	8	~	1
B	7	0	1	~	8
C	~	1	0	2	~
D	~	~	2	0	2
E	1	8	4	2	0

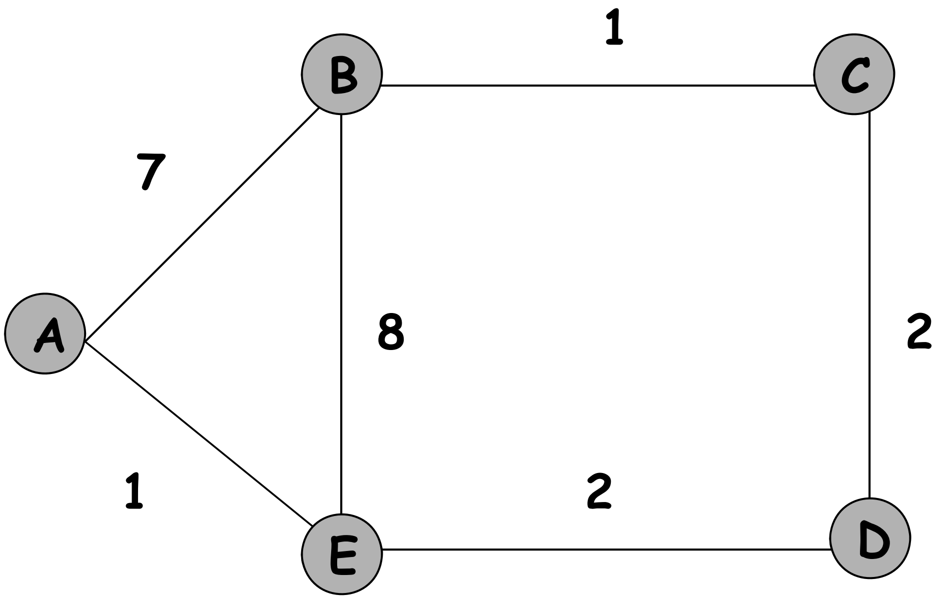
Router A updates Costs to Routers C&D



Info at node	A	B	C	D	E
A	0	7	5	3	1
B	7	0	1	~	8
C	~	1	0	2	~
D	~	~	2	0	2
E	1	8	4	2	0

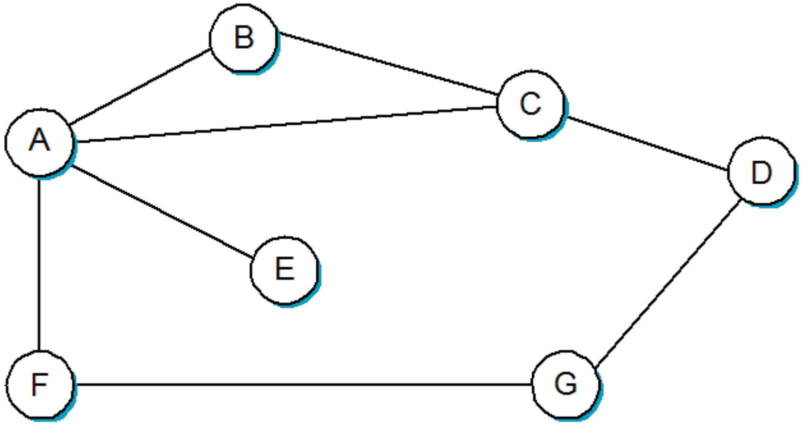
Distance to node

Final Distances



Info at node	Distance to node A	B	C	D	E
A	0	6	5	3	1
B	6	0	1	3	5
C	5	1	0	2	4
D	3	3	2	0	2
E	1	5	4	2	0

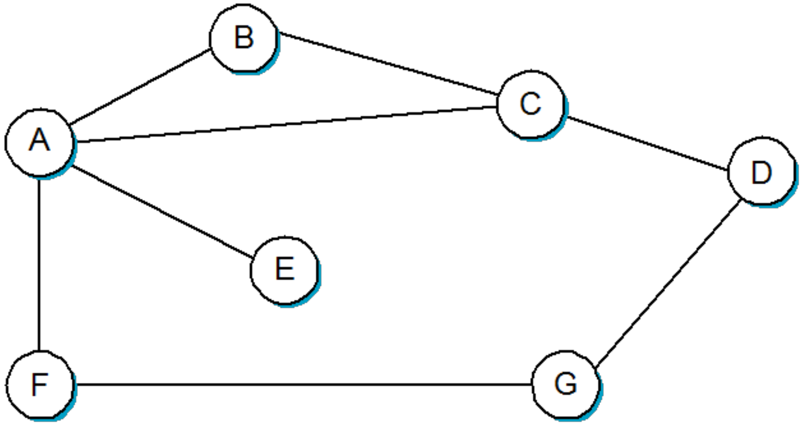
Another Example (More Realistic)



- Internal Information at each node ----->

	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

Routing Tables

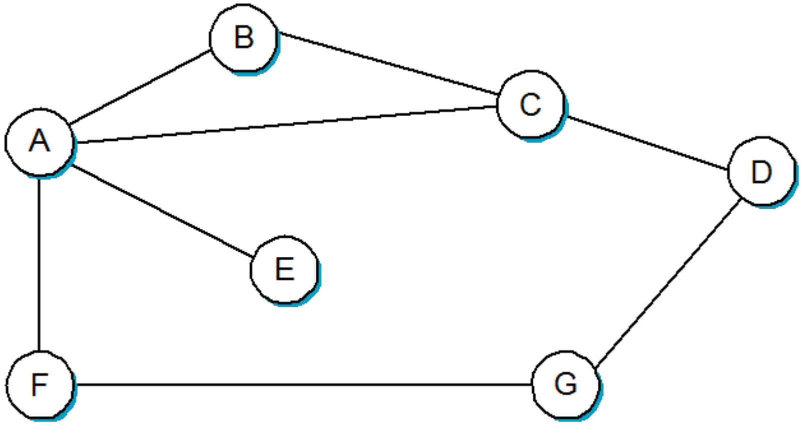


- With this information, routing table at A is -->

	Cost	Next Hop
B	1	B
C	1	C
D	∞	-
E	1	E
F	1	F
G	∞	-

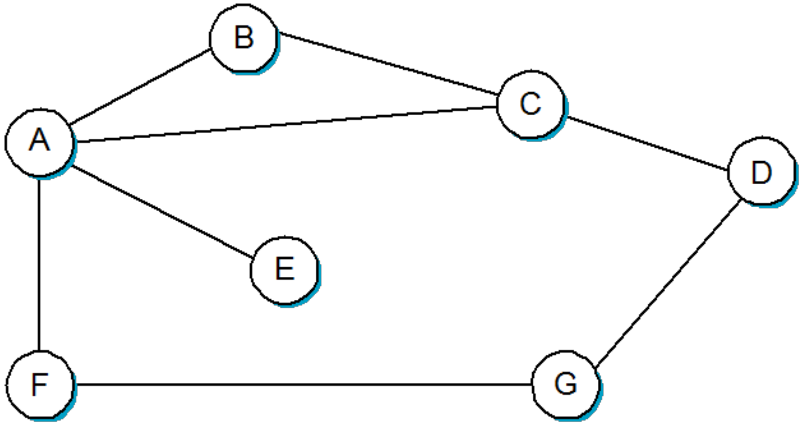
Evolution of the table.

- Each node sends a message to neighbors with a list of distances.
- F --> A with G is at a distance 1
- C --> A with D at distance 1.



	Cost	Next Hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

Final Distance Matrix



	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

Computer Networks

Dynamic Routing Protocols (2)

Link-State Protocols

LS Protocols: Dijkstra's algorithm

- ***Link-State (LS) Protocols***
  - Based on an algorithm by Dijkstra
  - Each router on the network is assumed to know the state of the links to all its neighbors
  - Each router will disseminate (via reliable flooding of ***link state packets***, LSPs) the information about its link states to all routers in the network.
  - In this case, every router will have enough information to build a complete map of the network and therefore is able to construct a ***Shortest Path Spanning Tree*** from itself to every other router

Link State Packets

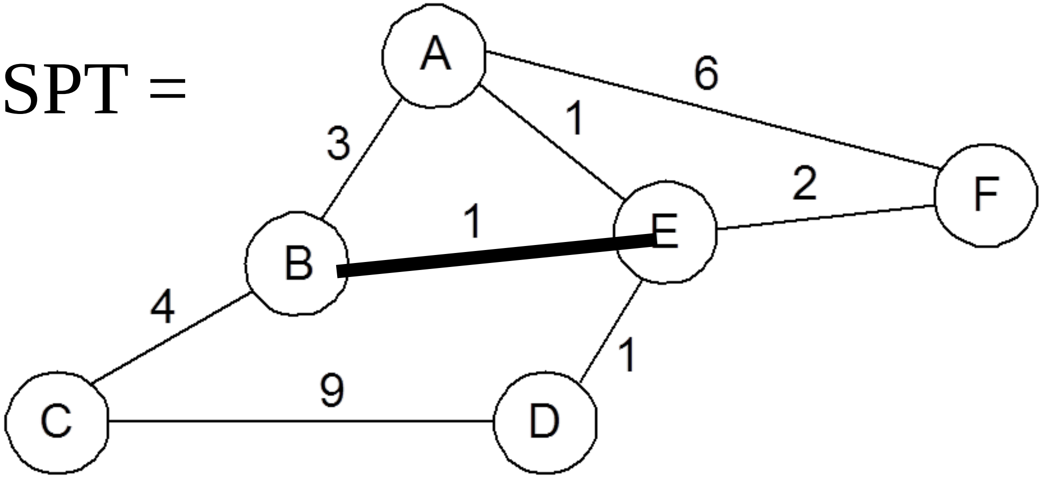
- The link state packets consist of the following information:
  - The ***address*** of the node creating the LSP
  - A ***list*** of directly connected neighbors to that node with the cost of the link to each neighbor
  - A ***sequence number*** to make sure it is the most recent one
  - A ***time-to-live*** to insure that an LSP doesn’t circulate indefinitely
- A node (router) will only send an LSP if there is a change of status to some of its links or if a timer expires

SPT (Shortest Path Tree) algorithm (Dijkstra)

- SPT = {*a*}
- for all nodes *v*
  - if *v* is adjacent to *a* then ***D(v)*** **= cost (a, v)**
  - else ***D(v) = infinity***
- Loop
  - *find w* not in SPT, where *D(w)* is min
  - add *w* in SPT
  - for all *v* adjacent to *w* and not in SPT
    - ***D(v) = min (D(v), D(w) + C(w, v))***
- until all nodes are in SPT

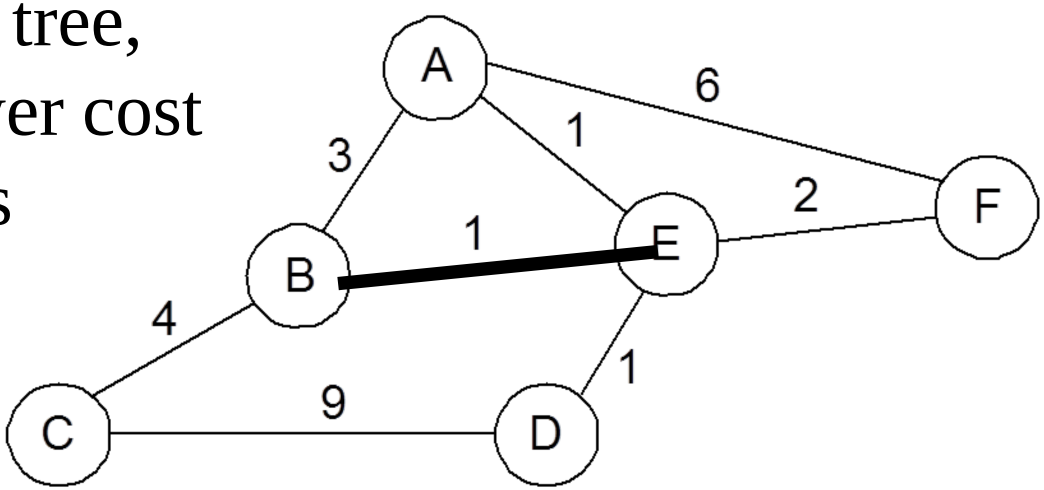
Dijkstra’s Shortest Path Algorithm (1)

- Initialize shortest path tree SPT = {B}
- For each n not in SPT, C(n) = l(s,n)
  - C(E) = 1, C(A) = 3, C(C) = 4, C(others) = infinity
- Add closest node to the tree: SPT = SPT U {E} since C(E) is minimum for all w not in SPT.
  - No shorter path to E can ever be found via some other roundabout path.
- Shortest path tree SPT = {B, E}



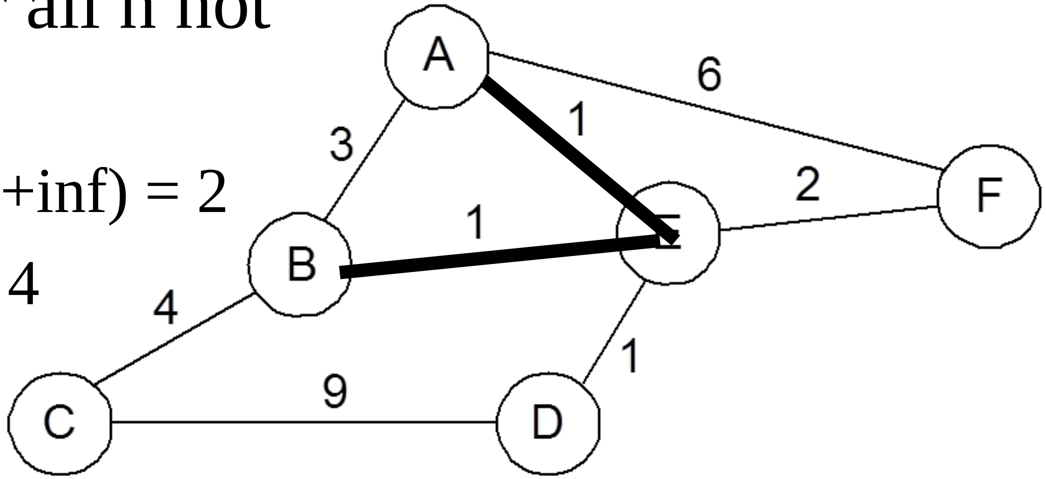
Dijkstra’s Shortest Path Algorithm (2)

- Recalculate C(n) = MIN (C(n), C(E) + l(E,n)) for all nodes n not yet in SPT
  - C(A) = MIN(C(A)=3, 1 + 1) = MIN(3,2) = 2
  - C(D) = MIN(infinity, 1 + 1) = 2
  - C(F) = MIN(infinity, 1 + 2) = 3
  - C(C) = MIN(4, 1 + infinity) = 4
- Each new node in tree, could create a lower cost path, so redo costs



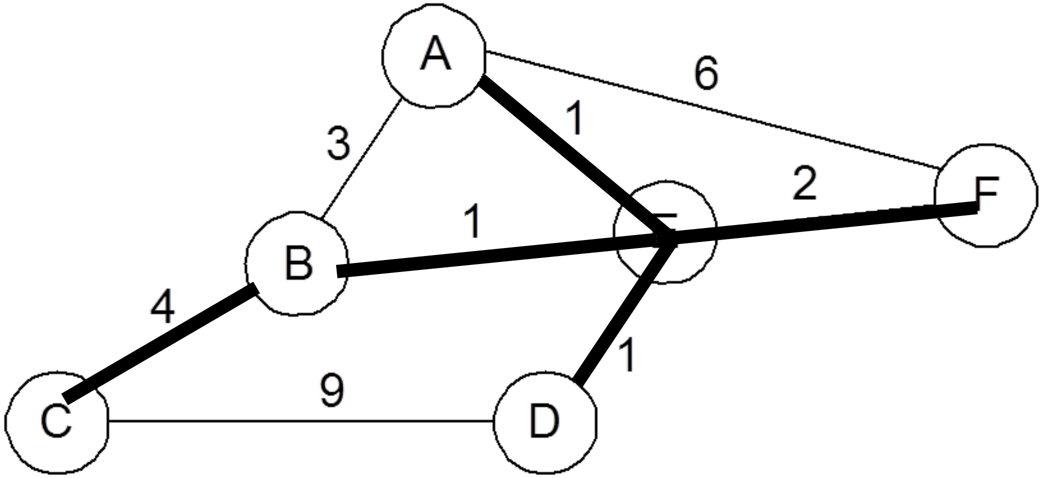
Dijkstra’s Shortest Path Algorithm (3)

- Loop again, select node with the lowest cost path:
  - C(A) = 2, C(D) = 2, C(F) = 3, C(C) = 4
  - SPT = SPT U {A} = {B, E, A}
  - No shorter path can be found from B to A, regardless of any new nodes added to tree
- Recalc: C(n) = MIN (C(n), C(A) + l(A,n)) for all n not yet in SPT
  - C(D) = MIN(2, 2+inf) = 2
  - C(F) = 3, C(C) = 4

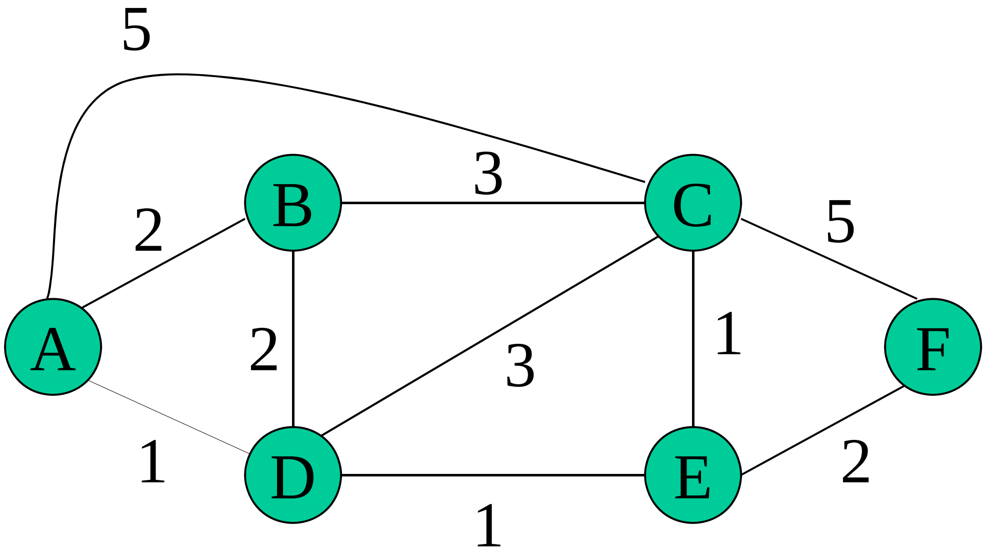


Dijkstra's Shortest Path Algorithm (4)

- Continue to loop, adding lowest cost node at each step and updating costs
- SPT crawls outward
- Remember to store the links in SPT as they are added (each node's predecessor is stored)
- Each node has to store the entire topology, or database of link costs

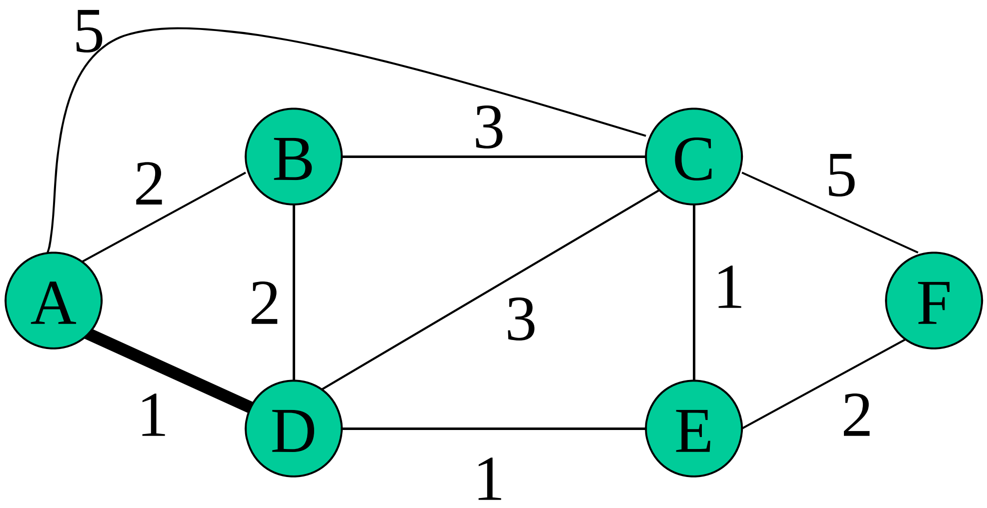


Example 2



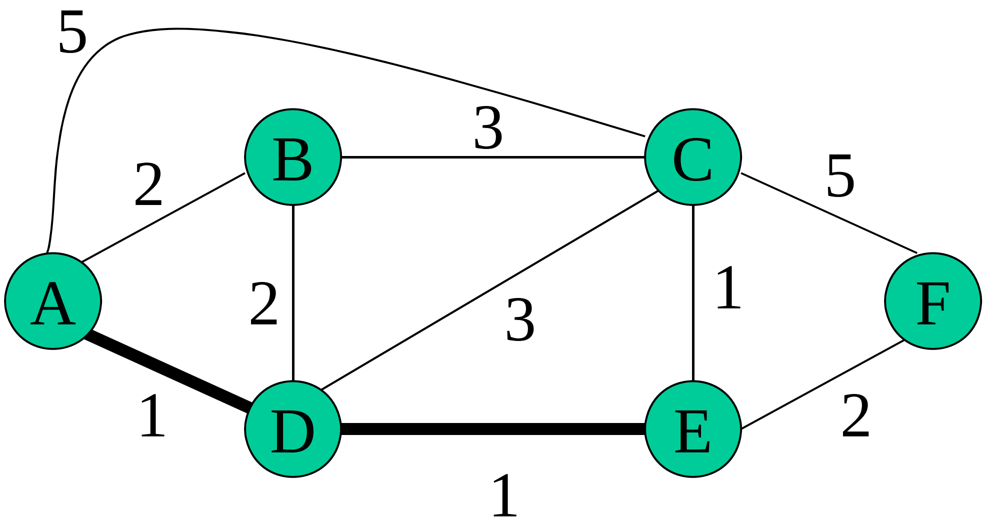
		B	C	D	E	F
step	SPT	D(b), P(b)	D(c), P(c)	D(d), P(d)	D(e), P(e)	D(f), P(f)
0	A	2, A	5, A	1, A	~	~

Example 2 (Continued)



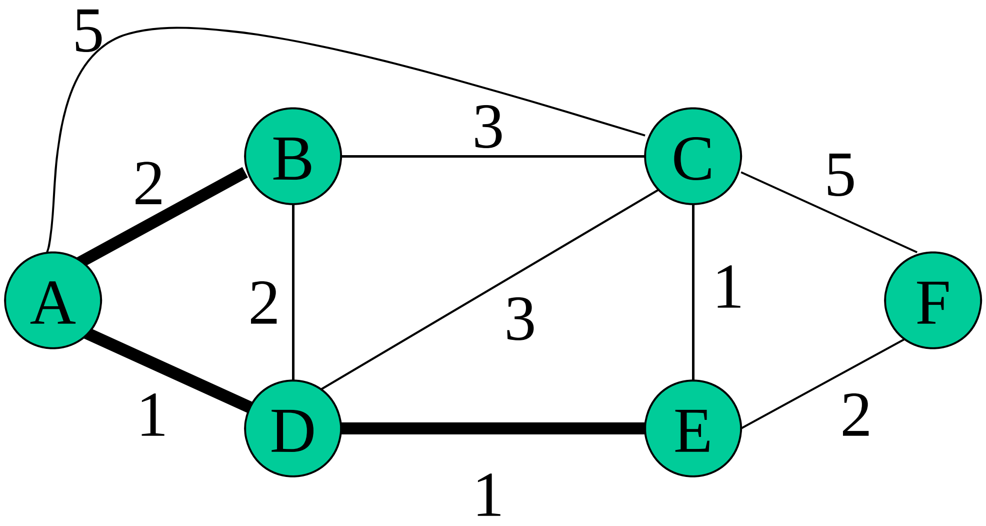
		B	C	D	E	F
step	SPT	D(b), P(b)	D(c), P(c)	D(d), P(d)	D(e), P(e)	D(f), P(f)
0	A	2, A	5, A	1, A	~	~
1	AD	2, A	4, D		2, D	~

Example 2 (Continued)



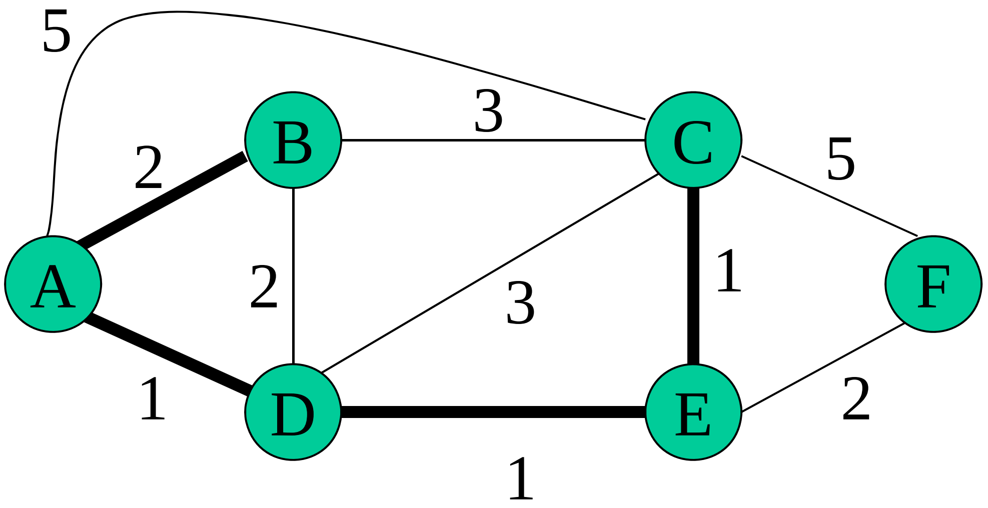
		B	C	D	E	F
step	SPT	D(b), P(b)	D(c), P(c)	D(d), P(d)	D(e), P(e)	D(f), P(f)
0	A	2, A	5, A	1, A	~	~
1	AD	2, A	4, D		2, D	~
2	ADE	2, A	3, E			4, E

Example 2 (Continued)



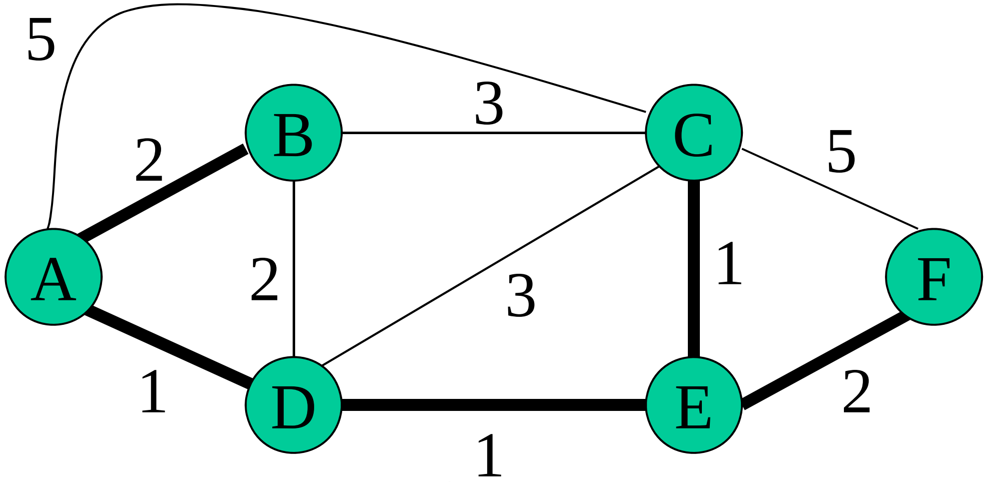
		B	C	D	E	F
step	SPT	D(b), P(b)	D(c), P(c)	D(d), P(d)	D(e), P(e)	D(f), P(f)
0	A	2, A	5, A	1, A	~	~
1	AD	2, A	4, D		2, D	~
2	ADE	2, A	3, E			4, E
3	ADEB		3, E			4, E

Example 2 (Continued)



		B	C	D	E	F
step	SPT	D(b), P(b)	D(c), P(c)	D(d), P(d)	D(e), P(e)	D(f), P(f)
0	A	2, A	5, A	1, A	~	~
1	AD	2, A	4, D		2, D	~
2	ADE	2, A	3, E			4, E
3	ADEB		3, E			4, E
4	ADEBC					4, E

Example 2 (Continued)

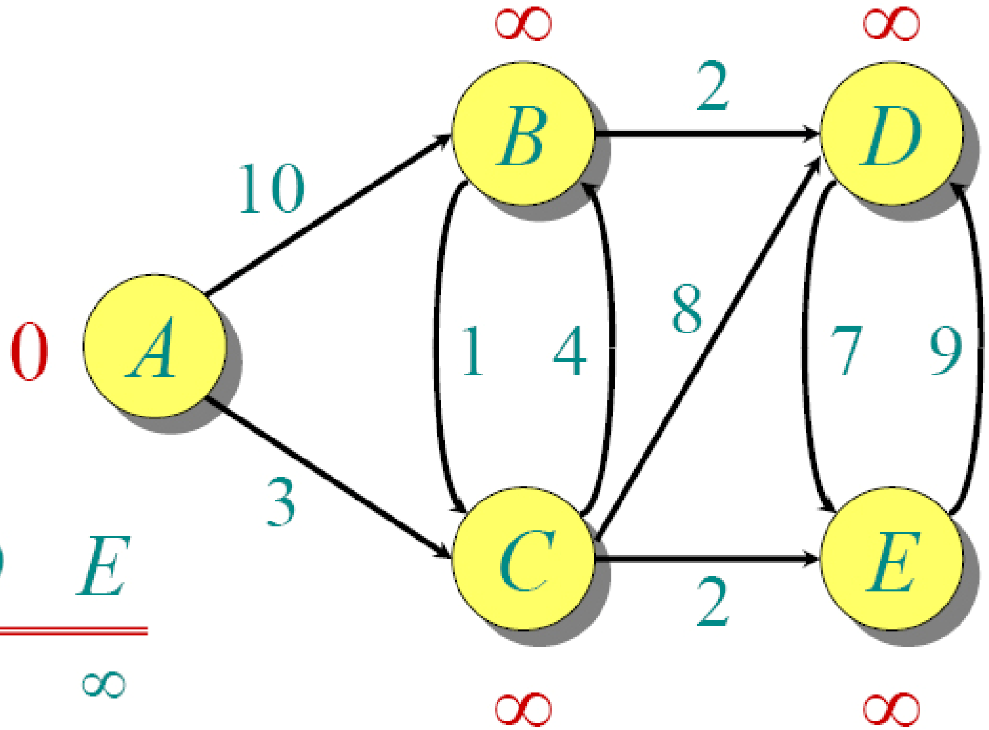


B C D E F

step	SPT	D(b), P(b)	D(c), P(c)	D(d), P(d)	D(e), P(e)	D(f), P(f)
0	A	2, A	5, A	1, A	~	~
1	AD	2, A	4, D		2, D	~
2	ADE	2, A	3, E			4, E
3	ADEB		3, E			4, E
4	ADEBC					4, E
5	ADEBCF					

Example 3

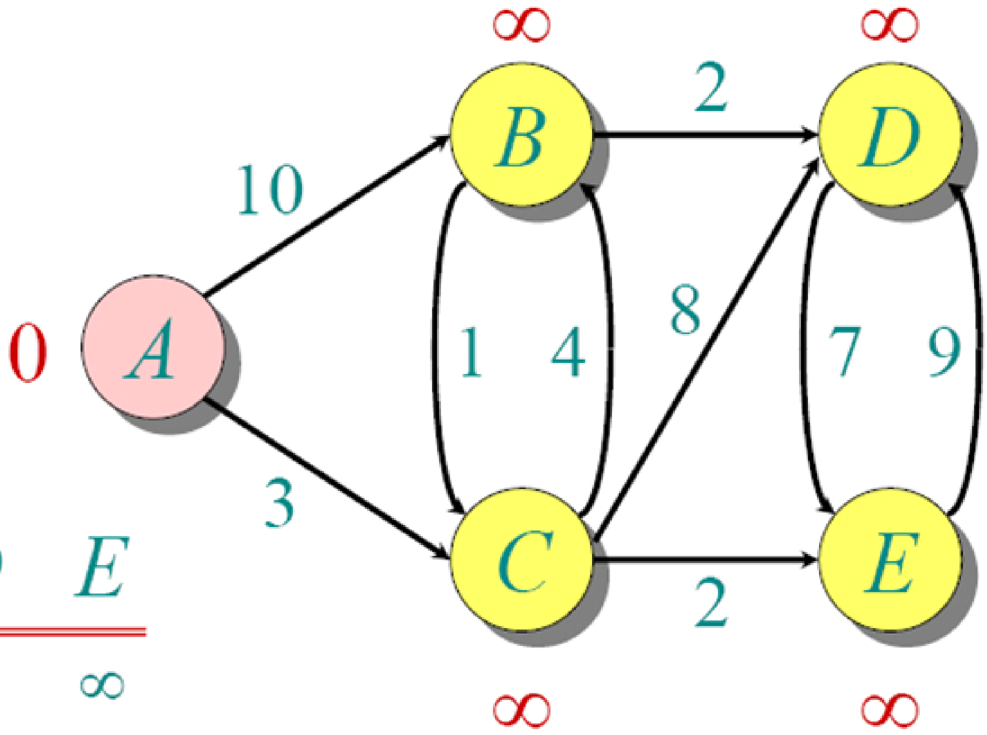
Initialize:



Q:	*A*	*B*	*C*	*D*	*E*
	0	∞	∞	∞	∞

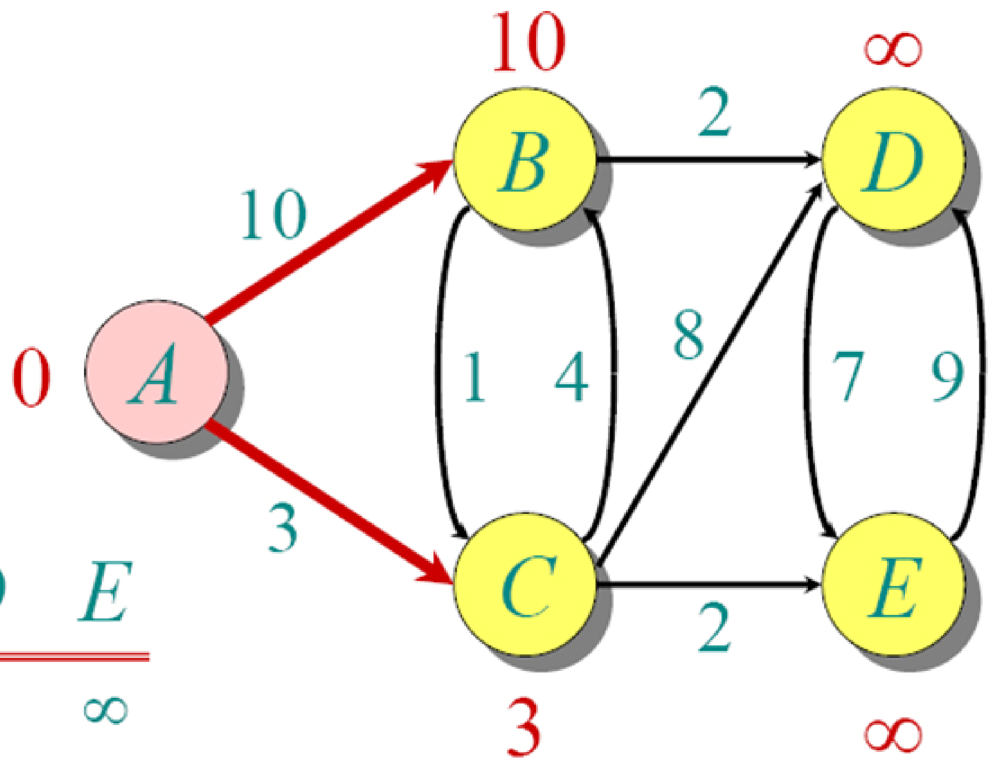
S: {}

Example 3 (Continued)



Q:	A	B	C	D	E
	0	∞	∞	∞	∞

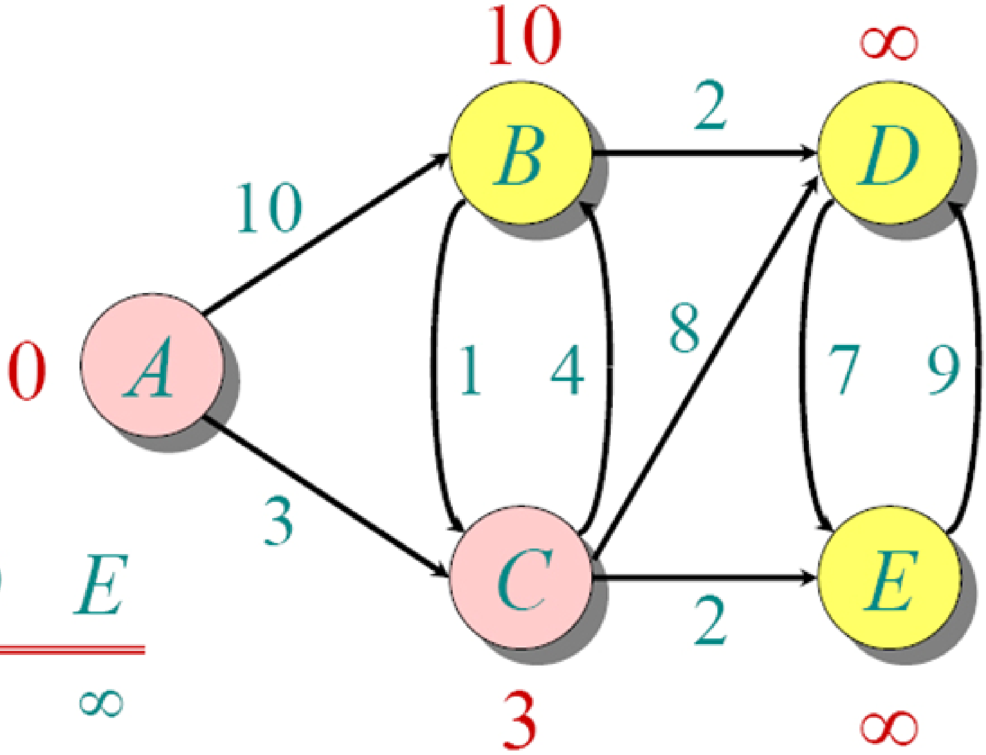
Example 3 (Continued)



Q:	*A*	*B*	*C*	*D*	*E*
	0	∞	∞	∞	∞
		10	3	∞	∞

S: { *A* }

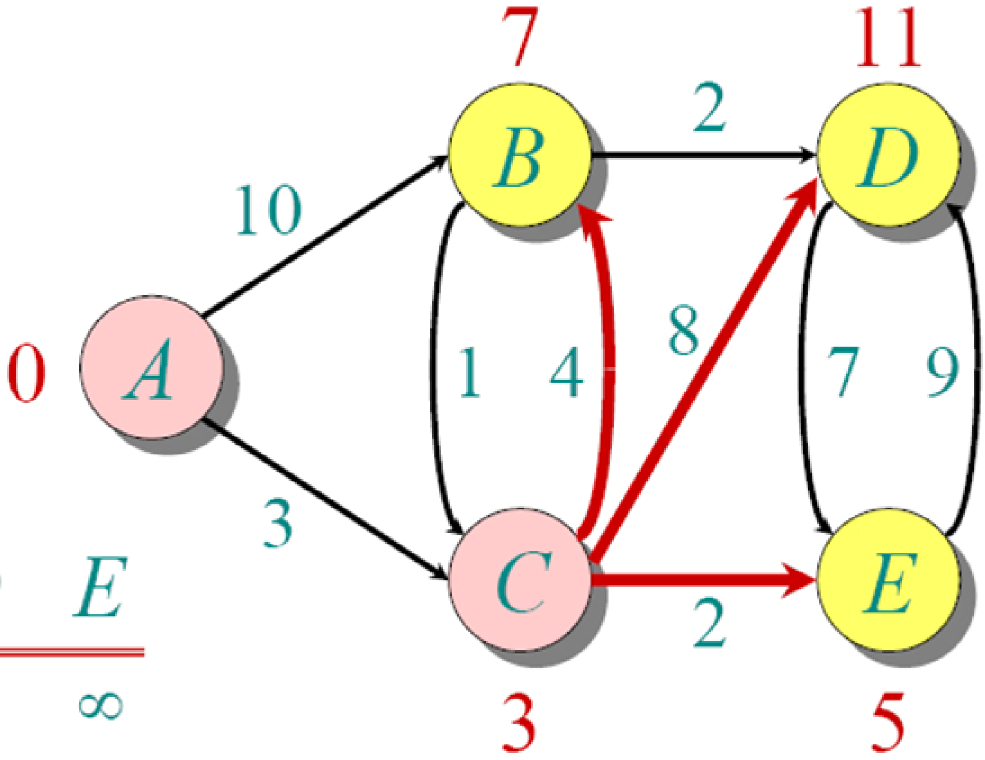
Example 3 (Continued)



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞

S: { A, C }

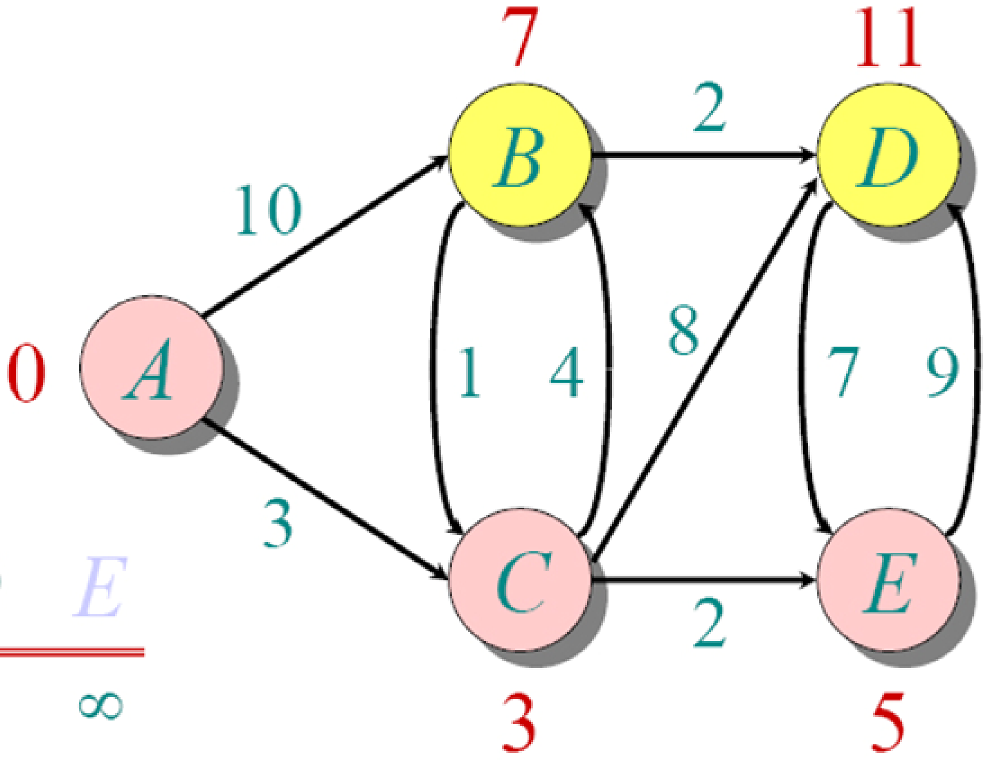
Example 3 (Continued)



Q:	*A*	*B*	*C*	*D*	*E*
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5

S: { *A*, *C* }

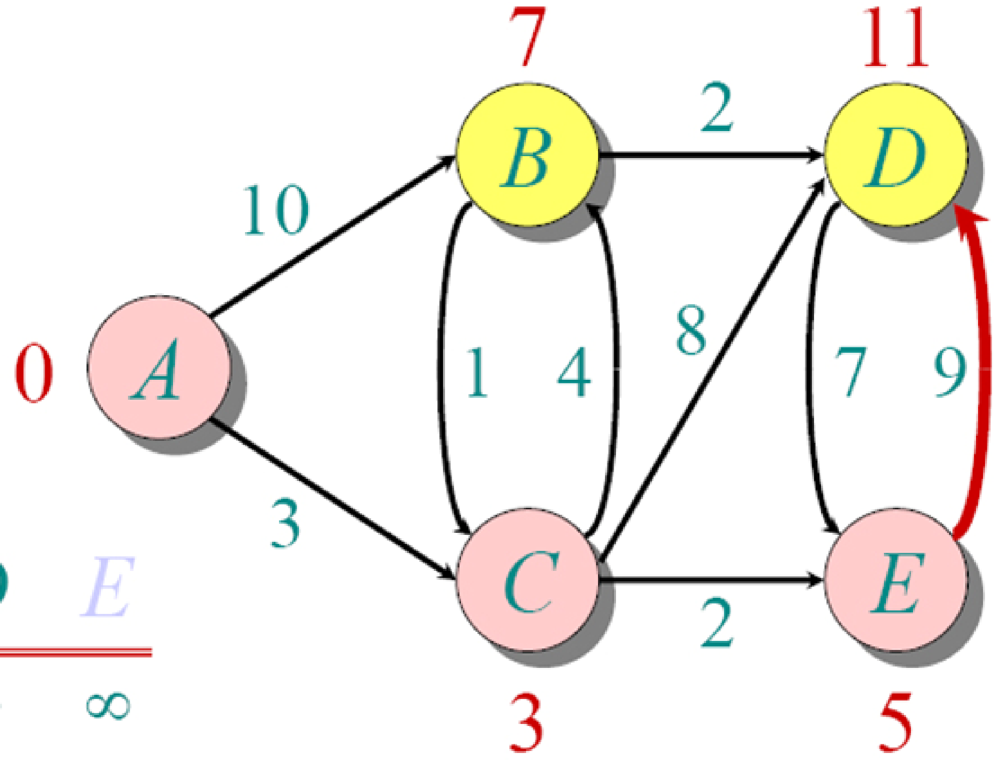
Example 3 (Continued)



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5

S: { *A*, *C*, *E* }

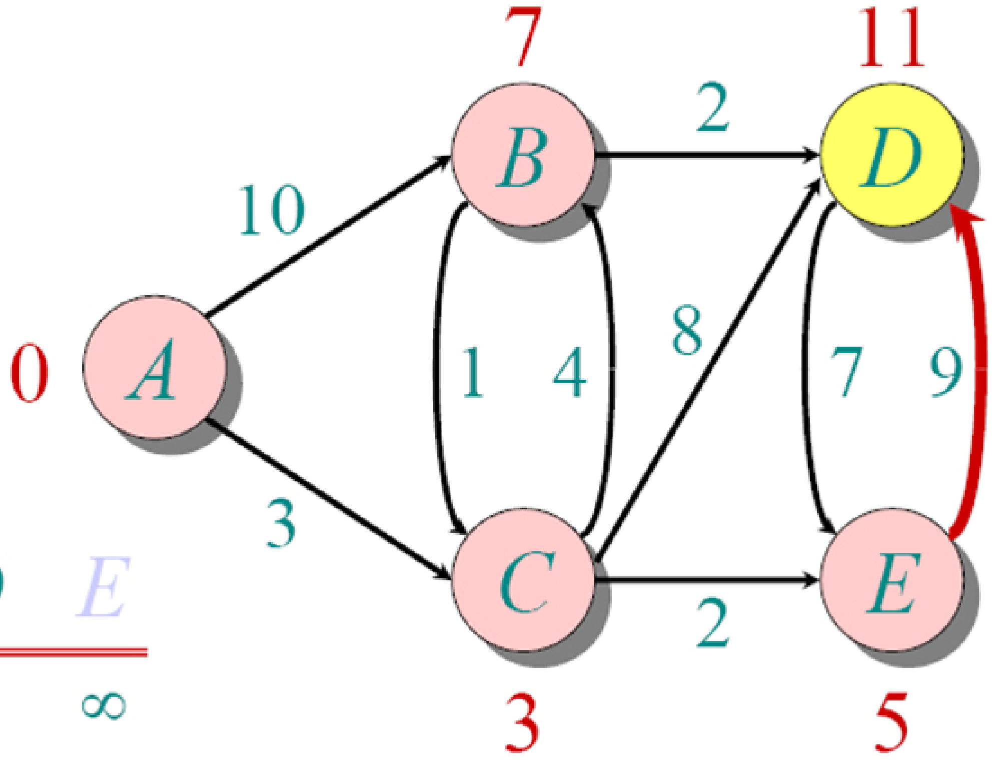
Example 3 (Continued)



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	

S: { A, C, E }

Example 3 (Continued)

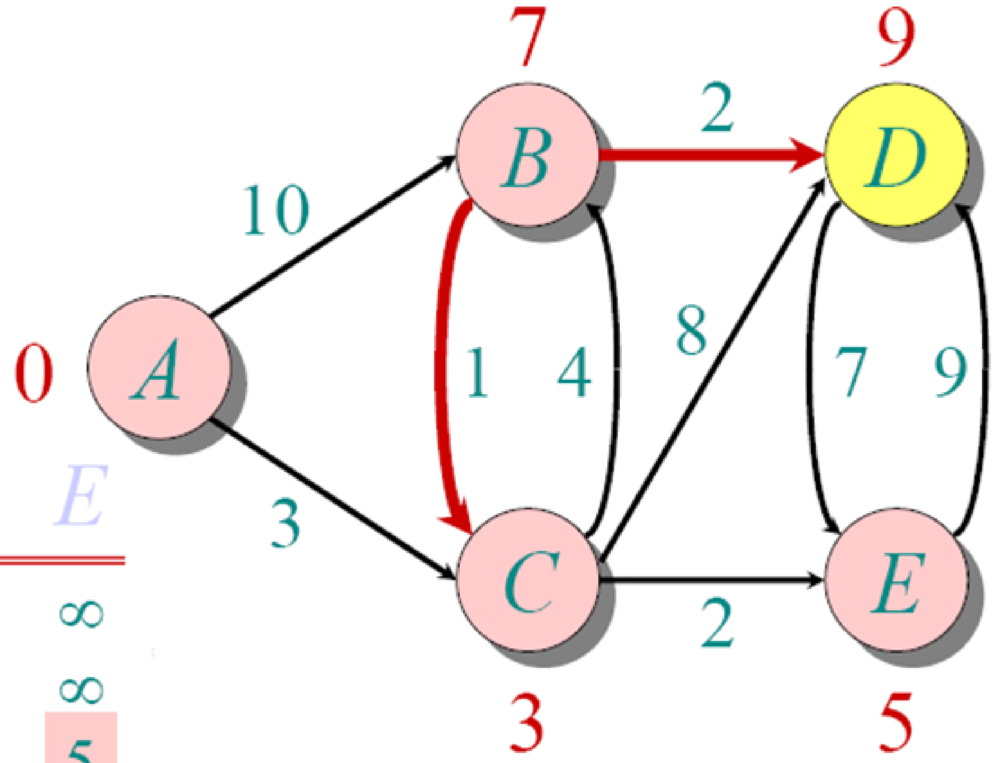


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	

S: { A, C, E, B }

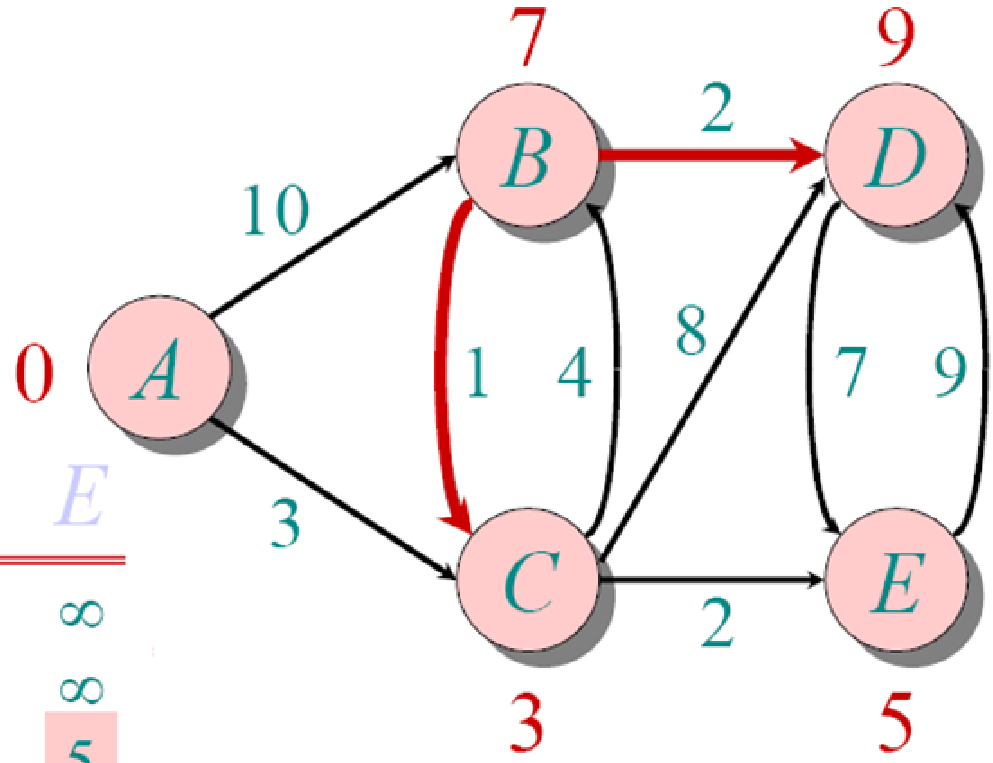
Example 3 (Continued)

Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	
				9	



S: { *A*, *C*, *E*, *B* }

Example 3 (Continued)



Q:	*A*	*B*	*C*	*D*	*E*
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	
				9	

S: { *A*, *C*, *E*, *B*, *D* }

Introduction to Dynamic Routing Protocols

Dynamic routing protocols play an important role in today's networks. The following sections describe several important benefits that dynamic routing protocols provide. In many networks, dynamic routing protocols are typically used with static routes.

Perspective and Background

Dynamic routing protocols have evolved over several years to meet the demands of changing network requirements. Although many organizations have migrated to more recent routing protocols such as Enhanced Interior Gateway Routing Protocol (EIGRP) and Open Shortest Path First (OSPF), many of the earlier routing protocols, such as Routing Information Protocol (RIP), are still in use today.

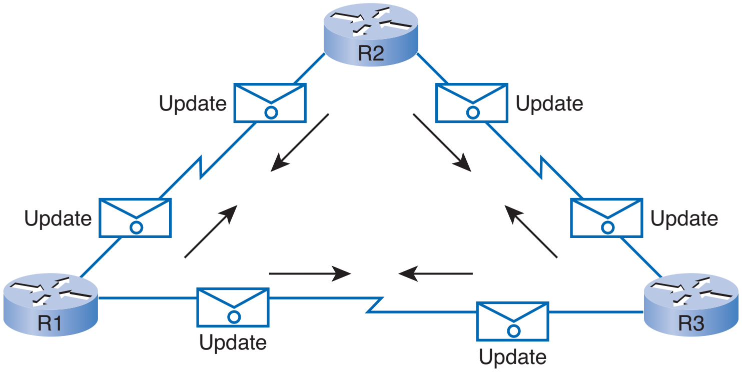
Note

This chapter presents an overview of the different dynamic routing protocols. More details about RIP, EIGRP, and OSPF routing protocols will be discussed in later chapters. The IS-IS and BGP routing protocols are explained in the CCNP curriculum. IGRP is the predecessor to EIGRP and is now considered obsolete.

Role of Dynamic Routing Protocol

What exactly are dynamic routing protocols? Routing protocols are used to facilitate the exchange of routing information between routers. Routing protocols allow routers to dynamically learn information about remote networks and automatically add this information to their own routing tables, as shown in Figure 3-2.

Figure 3-2 Routers Dynamically Pass Updates



Routing protocols determine the best path to each network, which is then added to the routing table. One of the primary benefits of using a dynamic routing protocol is that routers exchange routing information whenever there is a topology change. This exchange allows routers to automatically learn about new networks and also to find alternate paths if there is a link failure to a current network.

Compared to static routing, dynamic routing protocols require less administrative overhead. However, the expense of using dynamic routing protocols is dedicating part of a router's resources for protocol operation, including CPU time and network link bandwidth. Despite the benefits of dynamic routing, static routing still has its place. There are times when static routing is more appropriate and other times when dynamic routing is the better choice. More often than not, you will find a combination of both types of routing in any network that has a moderate level of complexity. You will learn about the advantages and disadvantages of static and dynamic routing later in this chapter.

Network Discovery and Routing Table Maintenance

Two important processes concerning dynamic routing protocols are initially discovering remote networks and maintaining a list of those networks in the routing table.

Purpose of Dynamic Routing Protocols

A routing protocol is a set of processes, algorithms, and messages that are used to exchange routing information and populate the routing table with the routing protocol's choice of best paths. The purpose of a routing protocol includes

- Discovering remote networks
- Maintaining up-to-date routing information
- Choosing the best path to destination networks
- Having the ability to find a new best path if the current path is no longer available

The components of a routing protocol are as follows:

- **Data structures:** Some routing protocols use tables or databases for their 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 processing 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 do other tasks to learn and maintain accurate information about the network.

Dynamic Routing Protocol Operation

All routing protocols have the same purpose: to learn about remote networks and to quickly adapt whenever there is a change in the topology. The method that a routing protocol uses to accomplish this depends on the algorithm it uses and the operational characteristics of that protocol. The operations of a dynamic routing protocol vary depending on the type of routing protocol and the specific operations of that routing protocol. The specific operations of RIP, EIGRP, and OSPF are examined in later chapters. In general, the operations of a dynamic routing protocol can be described as follows:

1. The router sends and receives routing messages on its interfaces.
2. The router shares routing messages and routing information with other routers that are using the same routing protocol.
3. Routers exchange routing information to learn about remote networks.
4. When a router detects a topology change, the routing protocol can advertise this change to other routers.

Note

Understanding dynamic routing protocol operation and concepts and using these protocols in real networks require a solid knowledge of IP addressing and subnetting. Three subnetting scenarios are available in *Routing Protocols and Concepts, CCNA Exploration Labs and Study Guide* (ISBN 1-58713-204-4) for your practice.

Dynamic Routing Protocol Advantages

Dynamic routing protocols provide several advantages, which will be discussed in this section. In many cases, the complexity of the network topology, the number of networks, and the need for the network to automatically adjust to changes require the use of a dynamic routing protocol.

Before examining the benefits of dynamic routing protocols in more detail, you need to consider the reasons why you would use static routing. Dynamic routing certainly has several advantages over static routing; however, static routing is still used in networks today. In fact, networks typically use a combination of both static and dynamic routing.

Table 3-1 compares dynamic and static routing features. From this comparison, you can list the advantages of each routing method. The advantages of one method are the disadvantages of the other.

Table 3-1 Dynamic Versus Static Routing

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

Static Routing Usage, Advantages, and Disadvantages

Static routing has several primary uses, including the following:

- Providing ease of routing table maintenance in smaller networks that are not expected to grow significantly.
- Routing to and from stub networks (see Chapter 2).
- Using a single default route, used to represent a path to any network that does not have a more specific match with another route in the routing table.

Static routing advantages are as follows:

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

Static routing disadvantages are as follows:

- Configuration and maintenance are 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 entire network for proper implementation.

Dynamic Routing Advantages and Disadvantages

Dynamic routing advantages are as follows:

- Administrator has less work in 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 are as follows:

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

Classifying Dynamic Routing Protocols

Figure 3-1 showed how routing protocols can be classified according to various characteristics. This chapter will introduce you to these terms, which will be discussed in more detail in later chapters.

This section gives an overview of the most common IP routing protocols. Most of these routing protocols will be examined in detail later in this book. For now, we will give a very brief overview of each protocol.

Routing protocols can be classified into different groups according to their characteristics:

- IGP or EGP
- Distance vector or link-state
- Classful or classless

The sections that follow discuss these classification schemes in more detail.

The most commonly used routing protocols are as follows:

- **RIP:** A distance vector interior routing protocol
- **IGRP:** The distance vector interior routing protocol developed by Cisco (deprecated from Cisco IOS Release 12.2 and later)
- **OSPF:** A link-state interior routing protocol
- **IS-IS:** A link-state interior routing protocol
- **EIGRP:** The advanced distance vector interior routing protocol developed by Cisco
- **BGP:** A path vector exterior routing protocol

Note

IS-IS and BGP are beyond the scope of this book.

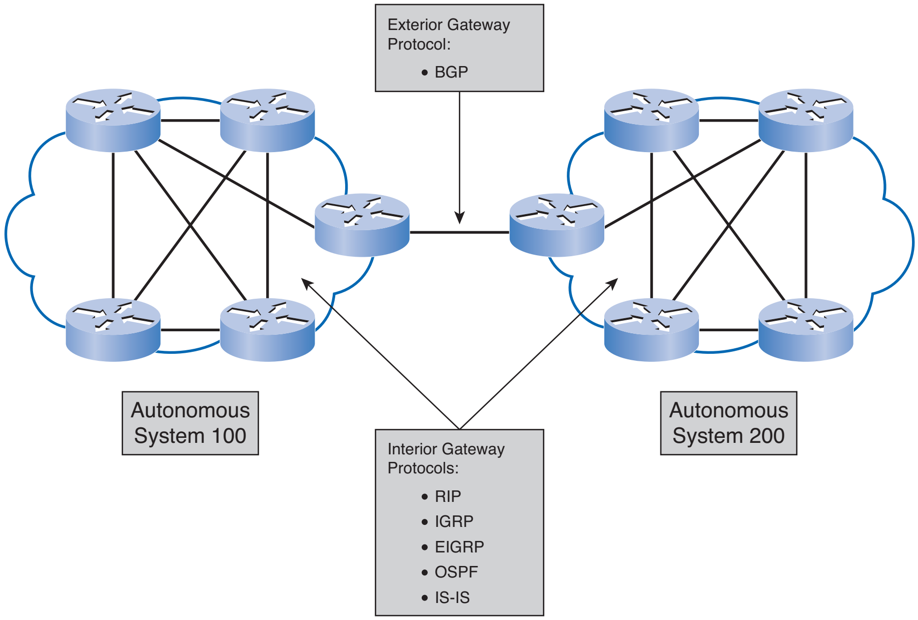
IGP and EGP

An ***autonomous system*** (AS)—otherwise known as a ***routing domain***—is a collection of routers under a common administration. Typical examples are a company's internal network and an ISP's network. Because the Internet is based on the autonomous system concept, two types of routing protocols are required: interior and exterior routing protocols. These protocols are

- ***Interior gateway protocols (IGP)*****:** Used for intra-autonomous system routing, that is, routing inside an autonomous system
- ***Exterior gateway protocols (EGP)*****:** Used for inter-autonomous system routing, that is, routing between autonomous systems

Figure 3-3 is a simplified view of the difference between IGPs and EGPs. The autonomous system concept will be explained in more detail later in the chapter. Even though this is an oversimplification, for now, think of an autonomous system as an ISP.

Figure 3-3 IGP Versus EGP Routing Protocols



IGPs are used for routing within a routing domain, those networks within the control of a single organization. An autonomous system is commonly composed of many individual networks belonging to companies, schools, and other institutions. An IGP is used to route within the autonomous system and also used to route within the individual networks themselves. For example, The Corporation for Education Network Initiatives in California (CENIC) operates an autonomous system composed of California schools, colleges, and universities. CENIC uses an IGP to route within its autonomous system to interconnect all of these institutions. Each of the educational institutions also uses an IGP of its own choosing to route within its own individual network. The IGP used by each entity provides best-path determination within its own routing domains, just as the IGP used by CENIC provides best-path routes within the autonomous system itself. IGPs for IP include RIP, IGRP, EIGRP, OSPF, and IS-IS.

Routing protocols (and more specifically, the algorithm used by that routing protocol) use a metric to determine the best path to a network. The metric used by the routing protocol RIP is *hop count*, which is the number of routers that a packet must traverse in reaching another network. OSPF uses *bandwidth* to determine the shortest path.

EGPs, on the other hand, are designed for use between different autonomous systems that are under the control of different administrations. BGP is the only currently viable EGP and is the routing protocol used by the Internet. BGP is a ***path vector protocol*** that can use many different attributes to measure routes. At the ISP level, there are often more important issues than just choosing the fastest path. BGP is typically used between ISPs and sometimes between a company and an ISP. BGP is not part of this course or CCNA; it is covered in CCNP.

Packet Tracer
□ **Activity**

Characteristics of IGP and EGP Routing Protocols (3.2.2)

In this activity, the network has already been configured within the autonomous systems. You will configure a default route from AS2 and AS3 (two different companies) to the ISP (AS1) to simulate the exterior gateway routing that would take place from both companies to their ISP. Then you will configure a static route from the ISP (AS1) to AS2 and AS3 to simulate the exterior gateway routing that would take place from the ISP to its two customers, AS2 and AS3. View the routing table before and after both static routes and default routes are added to observe how the routing table has changed. Use file e2-322.pka on the CD-ROM that accompanies this book to perform this activity using Packet Tracer.

Distance Vector and Link-State Routing Protocols

Interior gateway protocols (IGP) can be classified as two types:

- Distance vector routing protocols
- Link-state routing protocols

Distance Vector Routing Protocol Operation

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. Distance vector protocols typically use the Bellman-Ford algorithm for the best-path route determination.

Some distance vector protocols periodically send complete routing tables to all connected neighbors. In large networks, these routing updates can become enormous, causing significant traffic on the links.

Although the Bellman-Ford algorithm eventually accumulates enough knowledge to maintain a database of reachable networks, the algorithm does not allow a router to know the exact topology of an internetwork. The router only knows the routing information received from its neighbors.

Distance vector protocols use routers as signposts along the path to the final destination. The only information a router knows about a remote network is the distance or metric to reach that network and which path or interface to use to get there. Distance vector routing protocols do not have an actual map of the network topology.

Distance vector protocols work best in situations where

- The network is simple and flat and does not require a hierarchical design.
- The administrators do not have enough knowledge to configure and troubleshoot link-state protocols.
- Specific types of networks, such as hub-and-spoke networks, are being implemented.
- Worst-case convergence times in a network are not a concern.

Chapter 4, "Distance Vector Routing Protocols," covers distance vector routing protocol functions and operations in greater detail. You will also learn about the operations and configuration of the distance vector routing protocols RIP and EIGRP.

Link-State Protocol Operation

In contrast to distance vector routing protocol operation, a router configured with a ***link-state*** routing protocol can create a "complete view," or topology, of the network by gathering information from all the other routers. Think of using a link-state routing protocol as having a complete map of the network topology. The signposts along the way from source to destination are not necessary, because all link-state routers are using an identical "map" of the network. A ***link-state router*** uses the link-state information to create a topology map and to select the best path to all destination networks in the topology.

With some distance vector routing protocols, routers send periodic updates of their routing information to their neighbors. Link-state routing protocols do not use periodic updates. After the network has ***converged***, a link-state update is only sent when there is a change in the topology.

Link-state protocols work best in situations where

- The network design is hierarchical, usually occurring in large networks.
- The administrators have a good knowledge of the implemented link-state routing protocol.
- Fast convergence of the network is crucial.

Link-state routing protocol functions and operations will be explained in later chapters. You will also learn about the operations and configuration of the link-state routing protocol OSPF in Chapter 11, "OSPF."

Classful and Classless Routing Protocols

All routing protocols can also be classified as either

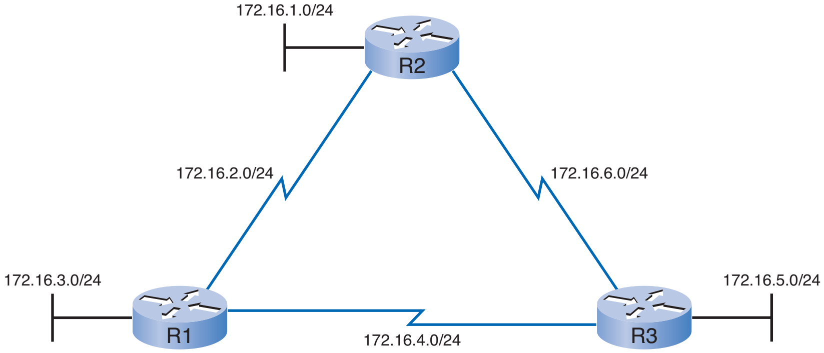
- Classful routing protocols
- Classless routing protocols

Classful Routing Protocols

Classful routing protocols do not send subnet mask information in routing updates. The first routing protocols, such as RIP, were classful. This was at a time when network addresses were allocated based on classes: Class A, B, or C. A routing protocol did not need to include the subnet mask in the routing update because the network mask could be determined based on the first octet of the network address.

Classful routing protocols can still be used in some of today's networks, but because they do not include the subnet mask, they cannot be used in all situations. Classful routing protocols cannot be used when a network is subnetted using more than one subnet mask. In other words, classful routing protocols do not support variable-length subnet masks (***VLSM***). Figure 3-4 shows an example of a network using the same subnet mask on all its subnets for the same major network address. In this situation, either a classful or classless routing protocol could be used.

Figure 3-4 Classful Routing



Classful: Subnet mask is the same throughout the topology.

There are other limitations to classful routing protocols, including their inability to support ***discontiguous*** networks. Later chapters discuss classful routing protocols, discontiguous networks, and VLSM in greater detail.

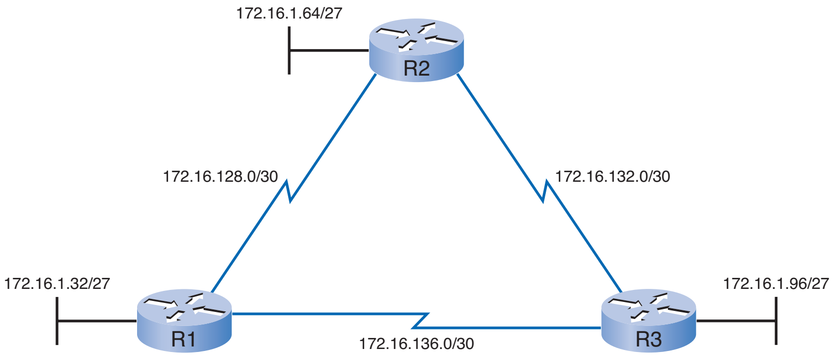
Classful routing protocols include RIPv1 and IGRP.

Classless Routing Protocols

Classless routing protocols include the subnet mask with the network address in routing updates. Today's networks are no longer allocated based on classes, and the subnet mask cannot be determined by the value of the first octet. Classless routing protocols are required in most networks today because of their support for VLSM, discontiguous networks, and other features that will be discussed in later chapters.

In Figure 3-5, notice that the classless version of the network is using both /30 and /27 subnet masks in the same topology. Also notice that this topology is using a discontiguous design.

Figure 3-5 Classless Routing



Classless: Subnet mask can vary in the topology.

Classless routing protocols are RIPv2, EIGRP, OSPF, IS-IS, and BGP.

Dynamic Routing Protocols and Convergence

An important characteristic of a routing protocol is how quickly it converges when there is a change in the topology.

Convergence is when the routing tables of all routers are at a state of consistency. The network has converged when all routers have complete and accurate information about the network. Convergence time is the time it takes routers to share information, calculate best paths, and update their routing tables. A network is not completely operable until the network has converged; therefore, most networks require short convergence times.

Convergence is both collaborative and independent. The routers share information with each other but must independently calculate the impacts of the topology change on their own routes. Because they develop an agreement with the new topology independently, they are said to *converge* on this consensus.

Convergence properties include the speed of propagation of routing information and the calculation of optimal paths. Routing protocols can be rated based on the speed to convergence; the faster the convergence, the better the routing protocol. Generally, RIP and IGRP are slow to converge, whereas EIGRP, OSPF, and IS-IS are faster to converge.

Packet Tracer
□ **Activity**

Convergence (3.2.5)

In this activity, the network has already been configured with two routers, two switches, and two hosts. A new LAN will be added, and you will watch the network converge. Use file e2-325.pka on the CD-ROM that accompanies this book to perform this activity using Packet Tracer.

Metrics

Metrics are a way to measure or compare. Routing protocols use metrics to determine which route is the best path.

Purpose of a Metric

There are cases when a routing protocol learns of more than one route to the same destination. To select the best path, the routing protocol must be able to evaluate and differentiate among the available paths. For this purpose, a metric is used. A metric is a value used by routing protocols to assign costs to reach remote networks. The metric is used to determine which path is most preferable when there are multiple paths to the same remote network.

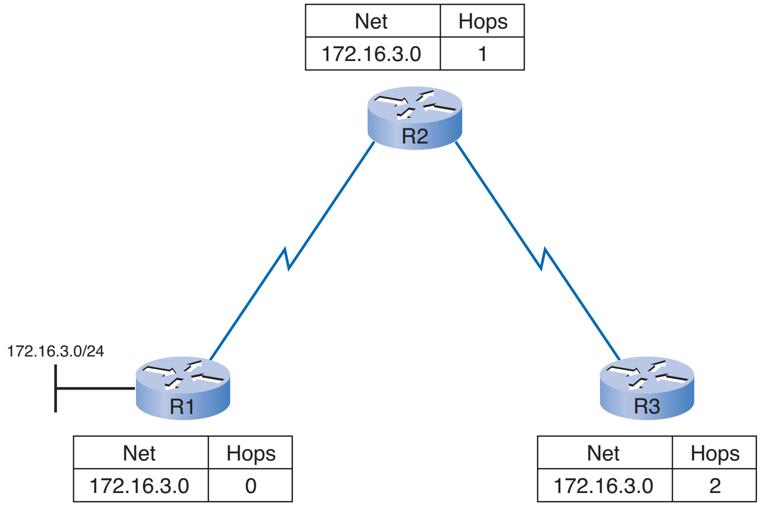
Each routing protocol calculates its metric in a different way. For example, RIP uses hop count, EIGRP uses a combination of bandwidth and delay, and the Cisco implementation of OSPF uses bandwidth. Hop count is the easiest metric to envision. The *hop count* refers to the number of routers a packet must cross to reach the destination network.

For Router R3 in Figure 3-6, network 172.16.3.0 is two hops, or two routers, away. For Router R2, network 172.16.3.0 is one hop away, and for Router R1, it is 0 hops (because the network is directly connected).

Note

The metrics for a particular routing protocol and a discussion of how they are calculated will be presented in the chapter for that routing protocol.

Figure 3-6 Metrics



Metrics and Routing Protocols

Different routing protocols use different metrics. The metric used by one routing protocol is not comparable to the metric used by another routing protocol.

Metric Parameters

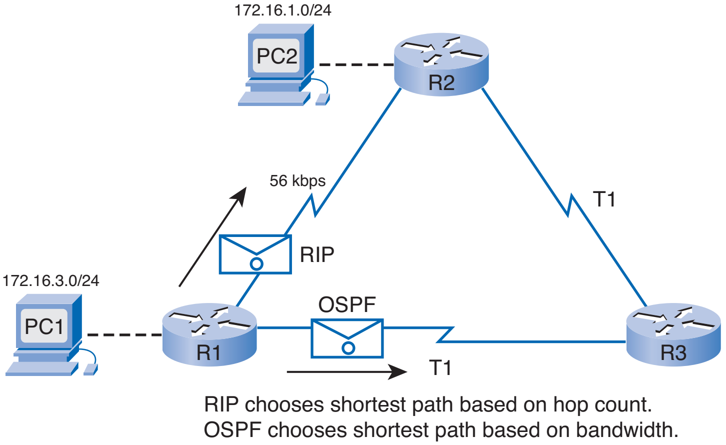
Two different routing protocols might choose different paths to the same destination because of using different metrics.

Figure 3-7 shows how R1 would reach the 172.16.1.0/24 network. RIP would choose the path with the least amount of hops through R2, whereas OSPF would choose the path with the highest bandwidth through R3.

Metrics used in IP routing protocols include the following:

- **Hop count:** A simple metric that counts the number of routers a packet must traverse.
- **Bandwidth:** Influences path selection by preferring the path with the highest bandwidth.
- **Load:** Considers the traffic utilization of a certain link.
- **Delay:** Considers the time a packet takes to traverse a path.
- **Reliability:** Assesses the probability of a link failure, calculated from the interface error count or previous link failures.
- **Cost:** A value determined either by the IOS or by the network administrator to indicate preference for a route. Cost can represent a metric, a combination of metrics, or a policy.

Figure 3-7 Hop Count Versus Bandwidth


172.16.1.0/24
PC2
R2
56 kbps
T1
RIP
172.16.3.0/24
PC1
R1
OSPF
R3
T1
RIP chooses shortest path based on hop count.
OSPF chooses shortest path based on bandwidth.

Note

At this point, it is not important to completely understand these metrics; they will be explained in later chapters.

Metric Field in the Routing Table

The routing table displays the metric for each dynamic and static route. Remember from Chapter 2 that static routes always have a metric of 0.

The list that follows defines the metric for each routing protocol:

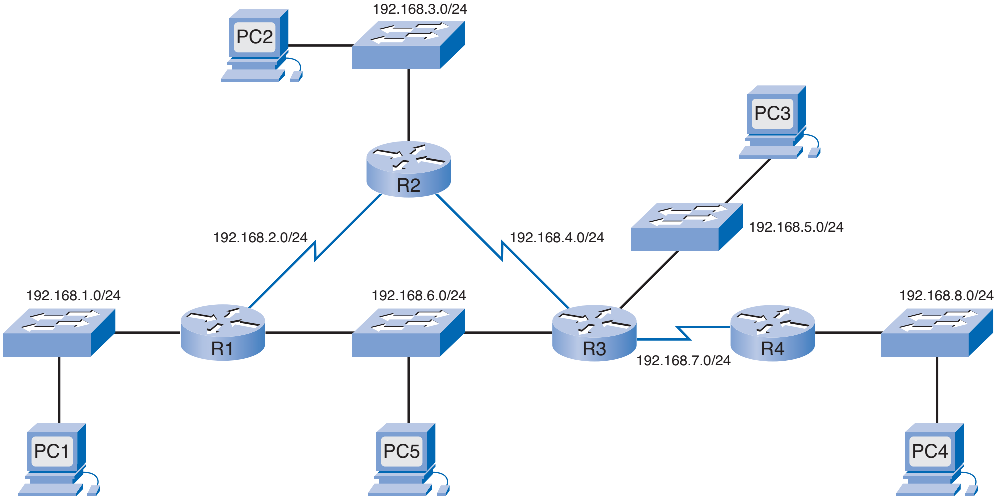
- **RIP: Hop count:** Best path is chosen by the route with the lowest hop count.
- **IGRP and EIGRP: Bandwidth, delay, reliability, and load:** Best path is chosen by the route with the smallest composite metric value calculated from these multiple parameters. By default, only bandwidth and delay are used.
- **IS-IS and OSPF: Cost:** Best path is chosen by the route with the lowest cost. The Cisco implementation of OSPF uses bandwidth to determine the cost. IS-IS is discussed in CCNP.

Routing protocols determine best path based on the route with the lowest metric.

In Figure 3-8, all the routers are using the RIP routing protocol.

The metric associated with a certain route can be best viewed using the **show ip route** command. The metric value is the second value in the brackets for a routing table entry. In Example 3-1, R2 has a route to the 192.168.8.0/24 network that is two hops away. The highlighted **2** in the command output is where the routing metric is displayed.

Figure 3-8 Best Path Determined in a Network Using RIP



Example 3-1 Routing Table for R2

```
R2# show ip route

<output omitted>

Gateway of last resort is not set

R    192.168.1.0/24 [120/1] via 192.168.2.1, 00:00:24, Serial0/0/0
C    192.168.2.0/24 is directly connected, Serial0/0/0
C    192.168.3.0/24 is directly connected, FastEthernet0/0
C    192.168.4.0/24 is directly connected, Serial0/0/1
R    192.168.5.0/24 [120/1] via 192.168.4.1, 00:00:26, Serial0/0/1
R    192.168.6.0/24 [120/1] via 192.168.2.1, 00:00:24, Serial0/0/0
                    [120/1] via 192.168.4.1, 00:00:26, Serial0/0/1
R    192.168.7.0/24 [120/1] via 192.168.4.1, 00:00:26, Serial0/0/1
R    192.168.8.0/24 [120/2] via 192.168.4.1, 00:00:26, Serial0/0/1
```

Load Balancing

You now know that individual routing protocols use metrics to determine the best route to reach remote networks. But what happens when two or more routes to the same destination have identical metric values? How will the router decide which path to use for packet forwarding? In this case, the router does not choose only one route. Instead, the router *load-balances* between these equal-cost paths. The packets are forwarded using all equal-cost paths.

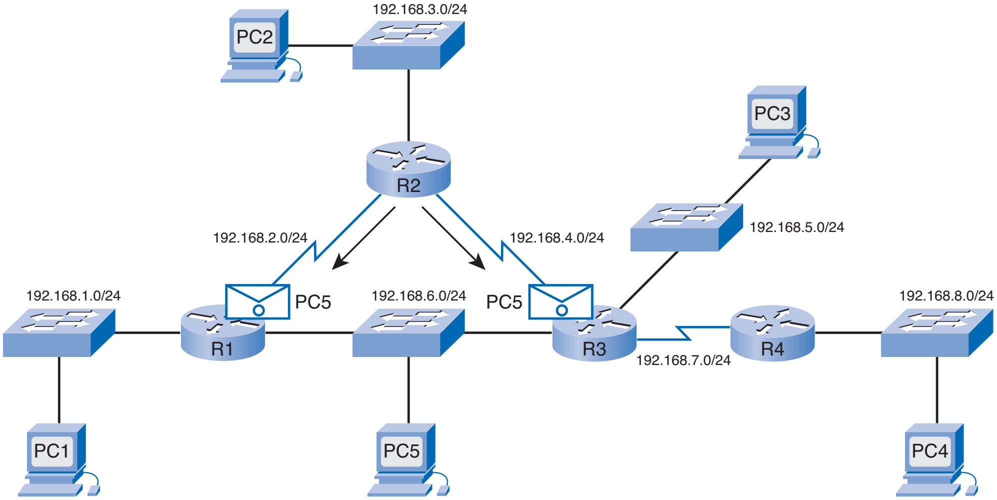
To see whether load balancing is in effect, check the routing table. Load balancing is in effect if two or more routes are associated with the same destination.

Note

Load balancing can be done either per packet or per destination. How a router actually load-balances packets between the equal-cost paths is governed by the switching process. The switching process will be discussed in greater detail in a later chapter.

Figure 3-9 shows an example of load balancing, assuming that R2 load-balances traffic to PC5 over two equal-cost paths.

Figure 3-9 Load Balancing Across Equal-Cost Paths



R2 load balances traffic destined for the 192.168.6.0/24 network.

The **show ip route** command in Example 3-1 reveals that the destination network 192.168.6.0 is available through 192.168.2.1 (Serial 0/0/0) and 192.168.4.1 (Serial 0/0/1). The equal-cost routes are shown again here:

```
R2# show ip route

<output omitted>
R    192.168.6.0/24 [120/1] via 192.168.2.1, 00:00:24, Serial0/0/0
                    [120/1] via 192.168.4.1, 00:00:26, Serial0/0/1
```

All the routing protocols discussed in this course are capable of automatically load-balancing traffic for up to four equal-cost routes by default. EIGRP is also capable of load-balancing across unequal-cost paths. This feature of EIGRP is discussed in the CCNP courses.

Administrative Distance

The following sections introduce the concept of administrative distance. Administrative distance will also be discussed within each chapter that focuses on a particular routing protocol.

Purpose of Administrative Distance

Before the routing process can determine which route to use when forwarding a packet, it must first determine which routes to include in the routing table. There can be times when a router learns a route to a remote network from more than one routing source. The routing process will need to determine which routing source to use. ***Administrative distance*** is used for this purpose.

Multiple Routing Sources

You know that routers learn about adjacent networks that are directly connected and about remote networks by using static routes and dynamic routing protocols. In fact, a router might learn of a route to the same network from more than one source. For example, a static route might have been configured for the same network/subnet mask that was learned dynamically by a dynamic routing protocol, such as RIP. The router must choose which route to install.

Note

You might be wondering about equal-cost paths. Multiple routes to the same network can only be installed when they come from the same routing source. For example, for equal-cost routes to be installed, they both must be static routes or they both must be RIP routes.

Although less common, more than one dynamic routing protocol can be deployed in the same network. In some situations, it might be necessary to route the same network address using multiple routing protocols such as RIP and OSPF. Because different routing protocols use different metrics—RIP uses hop count and OSPF uses bandwidth—it is not possible to compare metrics to determine the best path.

So, how does a router determine which route to install in the routing table when it has learned about the same network from more than one routing source? Cisco IOS makes the determination based on the administrative distance of the routing source.

Purpose of Administrative Distance

Administrative distance (AD) defines the preference of a routing source. Each routing source—including specific routing protocols, static routes, and even directly connected networks—is prioritized in order of most to least preferable using an administrative distance value. Cisco routers use the AD feature to select the best path when they learn about the same destination network from two or more different routing sources.

Administrative distance is an integer value from 0 to 255. The lower the value, the more preferred the route source. An administrative distance of 0 is the most preferred. Only a directly connected network has an administrative distance of 0, which cannot be changed.

Note

It is possible to modify the administrative distance for static routes and dynamic routing protocols. This is discussed in CCNP courses.

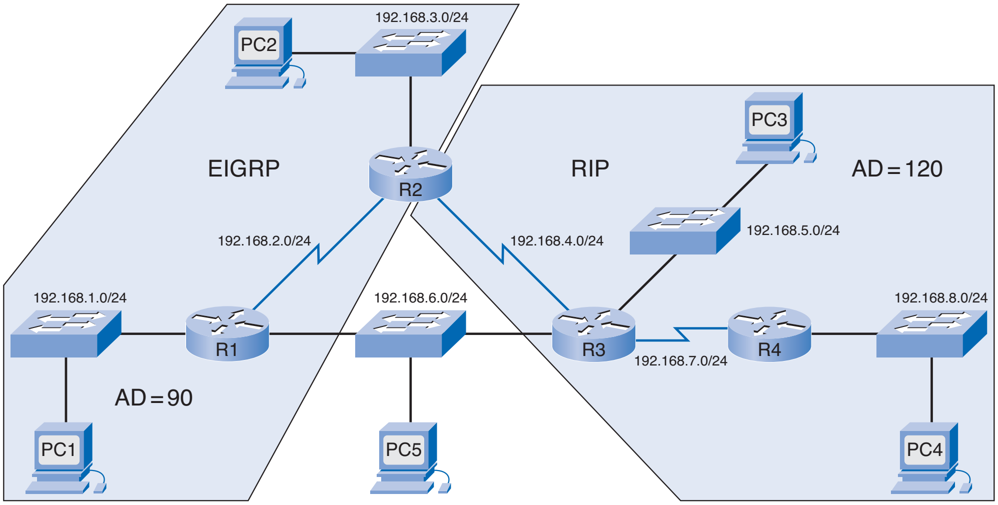
An administrative distance of 255 means the router will not believe the source of that route, and it will not be installed in the routing table.

Note

The term *trustworthiness* is commonly used when defining administrative distance. The lower the administrative distance value, the more trustworthy the route.

Figure 3-10 shows a topology with R2 running both EIGRP and RIP. R2 is running EIGRP with R1 and RIP with R3.

Figure 3-10 Comparing Administrative Distances



R1 and R3 do not “speak” the same routing protocol.

Example 3-2 displays the **show ip route** command output for R2.

Example 3-2 Routing Table for R2

```
R2# show ip route

<output omitted>

Gateway of last resort is not set

D     192.168.1.0/24 [90/2172416] via 192.168.2.1, 00:00:24, Serial0/0
C     192.168.2.0/24 is directly connected, Serial0/0/0
C     192.168.3.0/24 is directly connected, FastEthernet0/0
C     192.168.4.0/24 is directly connected, Serial0/0/1
R     192.168.5.0/24 [120/1] via 192.168.4.1, 00:00:08, Serial0/0/1
D     192.168.6.0/24 [90/2172416] via 192.168.2.1, 00:00:24, Serial0/0/0
R     192.168.7.0/24 [120/1] via 192.168.4.1, 00:00:08, Serial0/0/1
R     192.168.8.0/24 [120/2] via 192.168.4.1, 00:00:08, Serial0/0/1
```

The AD value is the first value in the brackets for a routing table entry. Notice that R2 has a route to the 192.168.6.0/24 network with an AD value of 90.

```
D     192.168.6.0/24 [90/2172416] via 192.168.2.1, 00:00:24, Serial0/0/0
```

R2 is running both RIP and EIGRP routing protocols. Remember, it is not common for routers to run multiple dynamic routing protocols, but is used here to demonstrate how administrative distance works. R2 has learned of the 192.168.6.0/24 route from R1 through EIGRP updates and from R3 through RIP updates. RIP has an administrative distance of 120, but EIGRP has a lower administrative distance of 90. So, R2 adds the route learned using EIGRP to the routing table and forwards all packets for the 192.168.6.0/24 network to Router R1.

What happens if the link to R1 becomes unavailable? Would R2 not have a route to 192.168.6.0? Actually, R2 still has RIP route information for 192.168.6.0 stored in the RIP database. This can be verified with the **show ip rip database** command, as shown in Example 3-3.

Example 3-3 Verifying RIP Route Availability

```
R2# show ip rip database

192.168.3.0/24     directly connected, FastEthernet0/0
192.168.4.0/24     directly connected, Serial0/0/1
```

```
192.168.5.0/24
    [1] via 192.168.4.1, Serial0/0/1
192.168.6.0/24
    [1] via 192.168.4.1, Serial0/0/1
192.168.7.0/24
    [1] via 192.168.4.1, Serial0/0/1
192.168.8.0/24
    [2] via 192.168.4.1, Serial0/0/1
```

Table 3-2 Default Administrative Distances

Route Source	AD
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

Summary

Dynamic routing protocols are used by routers to automatically learn about remote networks from other routers. In this chapter, you were introduced to several different dynamic routing protocols.

You learned the following about routing protocols:

- They can be classified as classful or classless.
- They can be a distance vector, link-state, or path vector type.
- They can be an interior gateway protocol or an exterior gateway protocol.

The differences in these classifications will become better understood as you learn more about these routing concepts and protocols in later chapters.

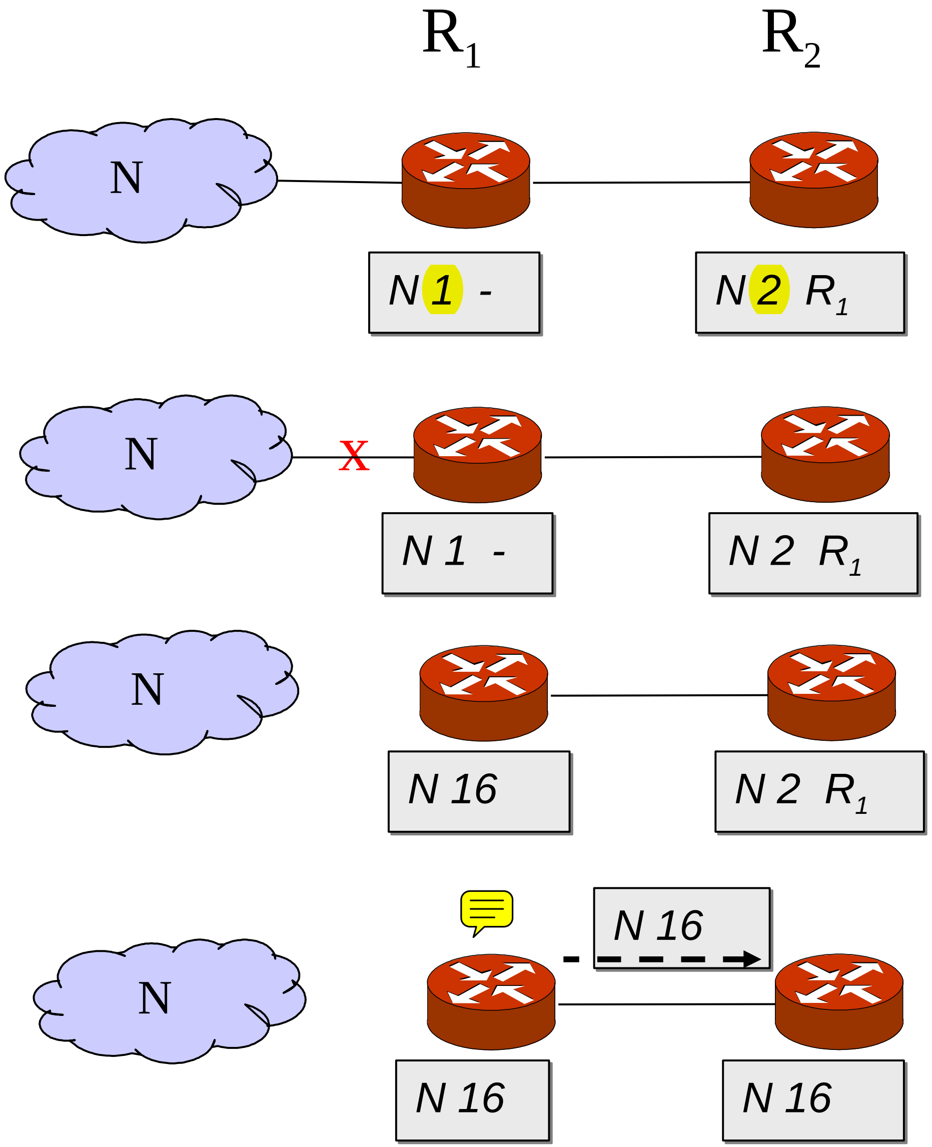
Routing protocols not only discover remote networks but also have a procedure for maintaining accurate network information. When there is a change in the topology, it is the function of the routing protocol to inform other routers about this change. When there is a change in the network topology, some routing protocols can propagate that information throughout the routing domain faster than other routing protocols.

The process of bringing all routing tables to a state of consistency is called convergence. Convergence is when all the routers in the same routing domain or area have complete and accurate information about the network.

Metrics are used by routing protocols to determine the best path or shortest path to reach a destination network. Different routing protocols can use different metrics. Typically, a lower metric means a better path. Five hops to reach a network is better than ten hops.

Routers sometimes learn about multiple routes to the same network from both static routes and dynamic routing protocols. When a Cisco router learns about a destination network from more than one routing source, it uses the administrative distance value to determine which source to use. Each dynamic routing protocol has a unique administrative value, along with static routes and directly connected networks. The lower the administrative value, the more preferred the route source. A directly connected network is always the preferred source, followed by static routes and then various dynamic routing protocols.

RIP Problem: Count to Infinity (Two-node instability)



Initially, R_1 and R_2 both have a route to N with metric 1 and 2, respectively.

The link between R_1 and N fails.

Now R_1 removes its route to N, by setting its metric to 16 (infinity).

Now two things can happen: Either R_1 reports its route to R_2. Everything is fine.

RIP Problem: Count to Infinity

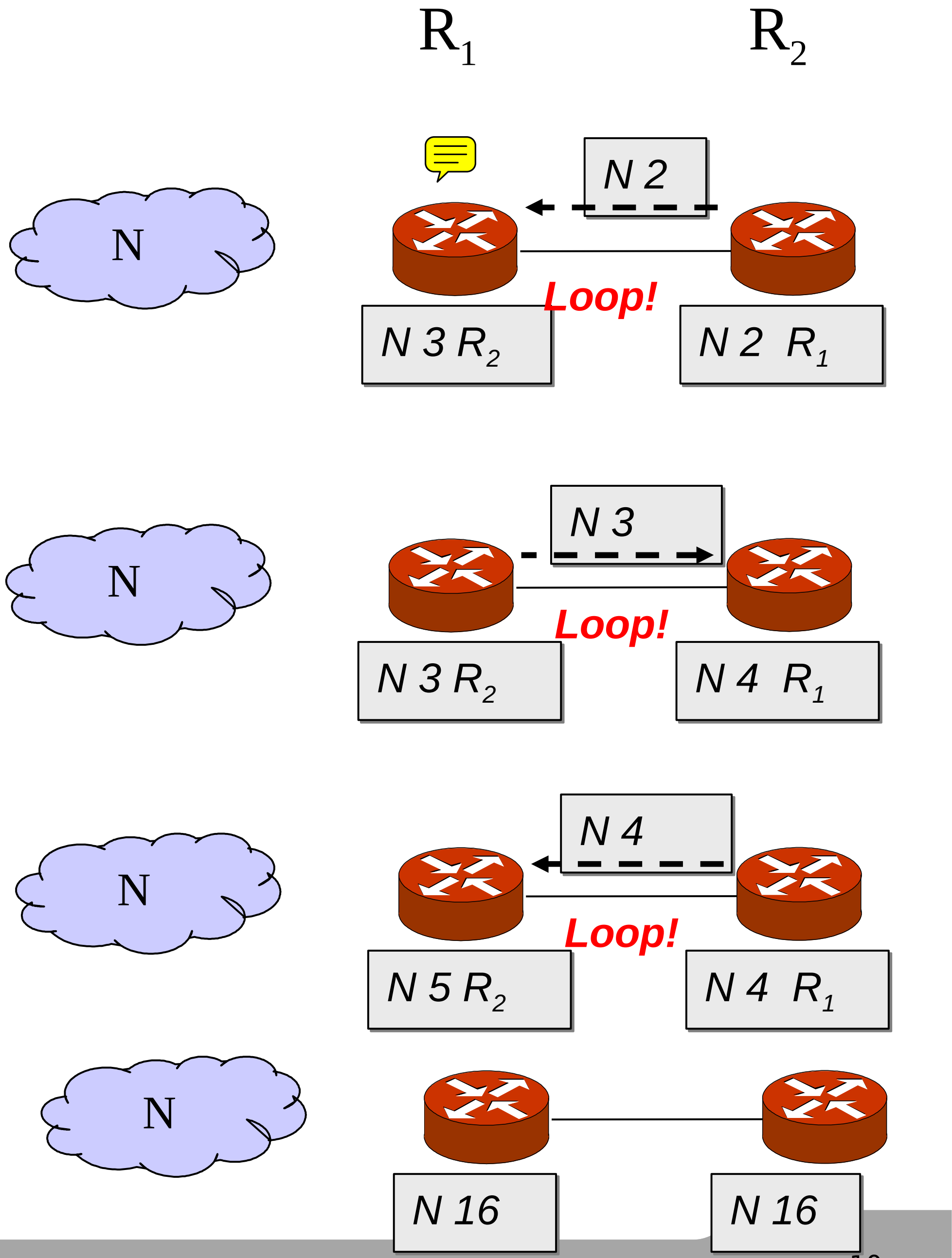
The other alternative is that R_2, which still has a route to N, advertises it to R_1. Now things start to go wrong: packets to N are looped until their TTL expires!

Eventually (~10-20s), R_1 sends an update to R_2. The cost to N increases, but the loop remains.

Yet some time later, R_2 sends an update to R_1.

…

Finally, the cost reaches infinity at 16, and N is unreachable. The loop is broken!



Solutions to count-to-infinity in RIP

- Infinity
- Split Horizon
- Split Horizon with Poison Reverse
- Triggered Updates
- Hold-down

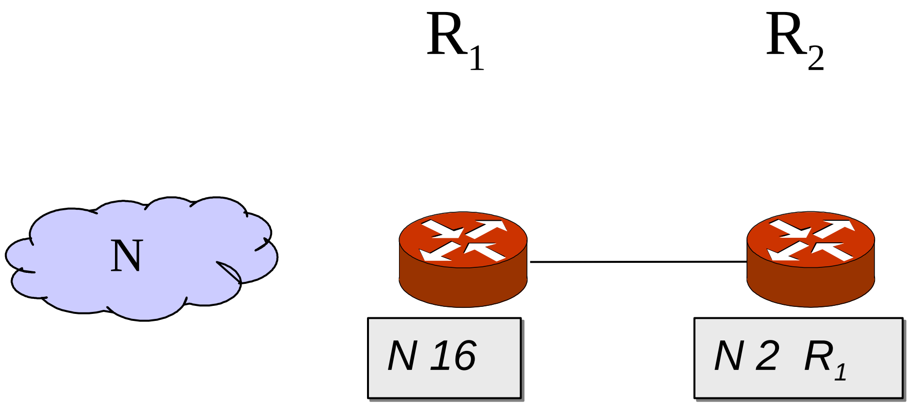
Infinity

- Counting to infinity takes a long time.
- Thus infinity is set something more limited, namely 16.
- This limits the routing domain to 15 hops, and also makes counting to infinity a little faster,...

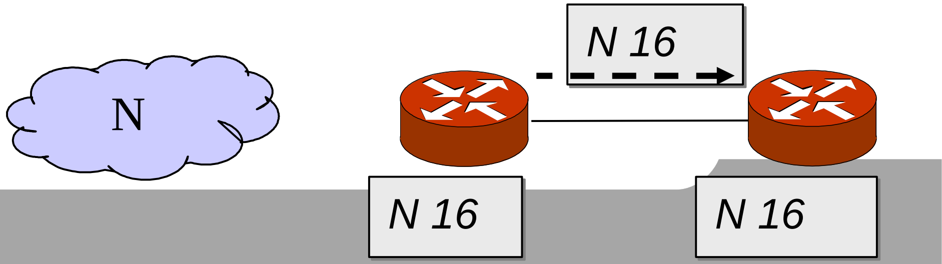
Split Horizon

- Do not send routes back over the same interface from which the route arrived.
- This helps in avoiding “mutual deception”: two routers tell each other they can reach a destination via each other.
- Split Horizon MUST be supported on all RIP routers

R_2, does not announce the route to N to R_1 since that is where it came from.



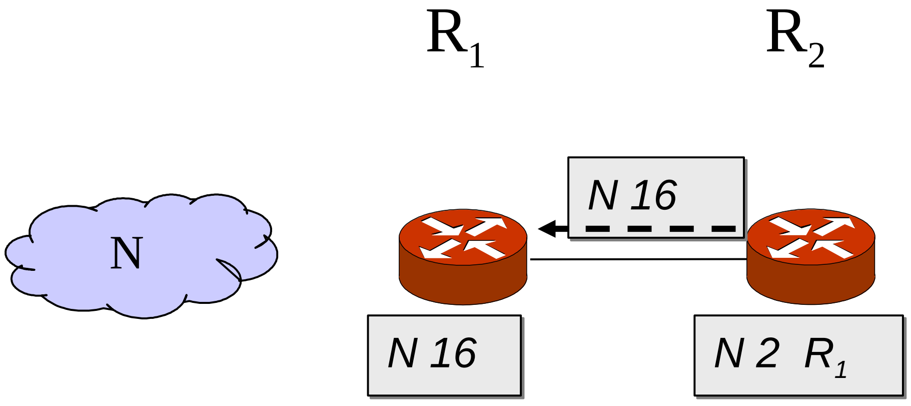
Eventually, R_1 reports its route to R_2 and everything is fine.



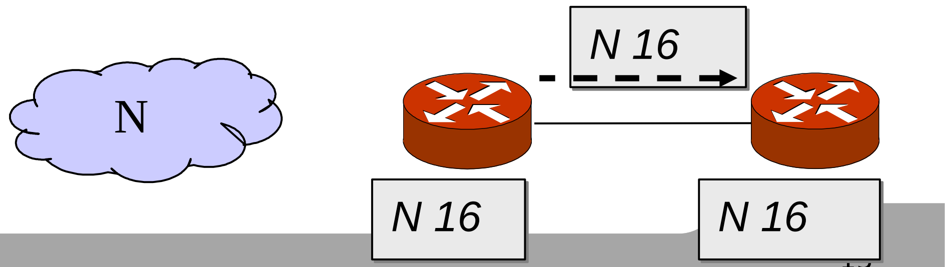
Split Horizon + Poison Reverse

- Advertise reverse routes with a metric of 16 (i.e., unreachable).
- Does not add inormation but breaks loops faster
- Adds protocol overhead
- Poison reverse SHOULD be supported by all RIP implementations, but it is OK to be able to turn it off.

R_2 always announces an unreachable route to N to R_1.

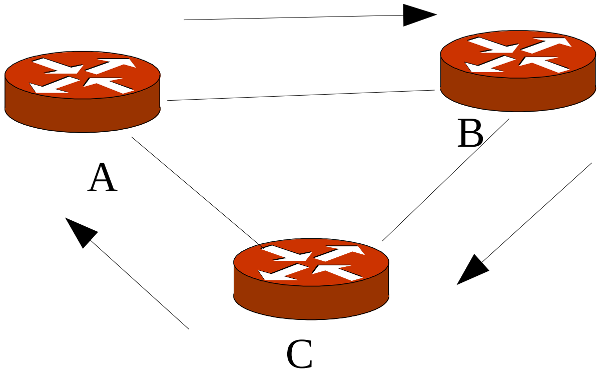


Eventually, R_1 reports its route to R_2 and everything is fine.



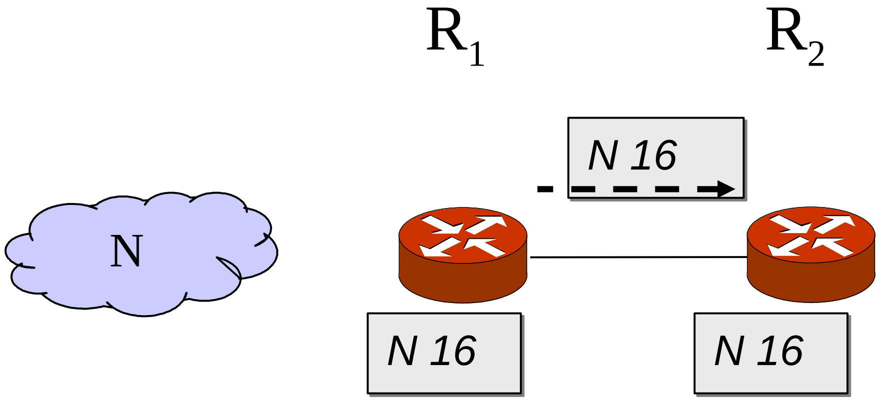
Remaining problems

- More than two routers involved in mutual deception
  - A may believe it has a route through B, B through C, and C through A
- In this case, split horizon with poison reverse does not help



Triggered Update

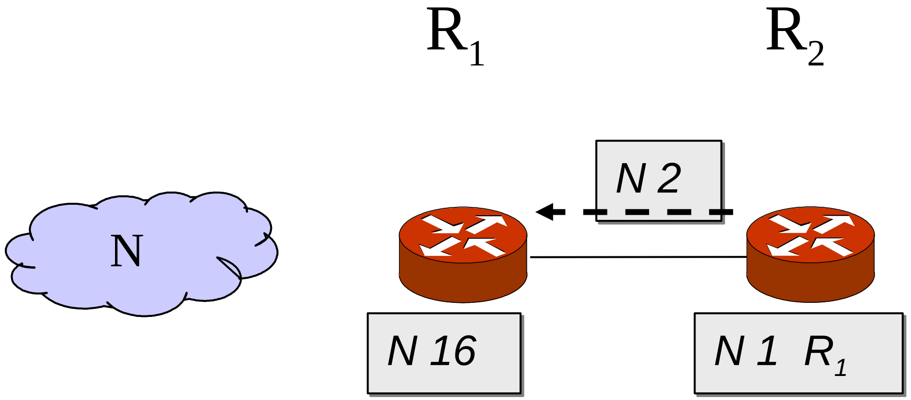
- Send out update immediately when metric changes
- But only the changed route, not the complete table
- This may lead to a cascade of updates
    - Apply the rule above recursively!
    - RIP filters these updates by not allowing more than one every 1-5 seconds.
- A router may use triggered update only when deleting routes (16).
- CISCO also implements “flash updates”: on boot, broadcast a request -> all neighbours answer with updates.



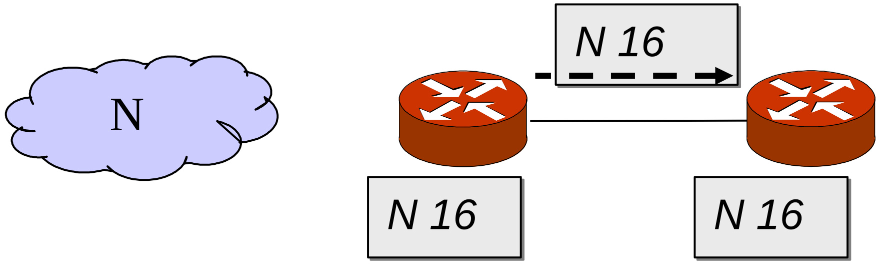
R_1 Immediately announces the broken link when it happens.

Hold Down

- When a route is removed, no update of this route is accepted for some period of time (hold-down time)- to give everyone a chance to remove the route.
- Hold-down is not in the RIP RFC, but CISCO implements it



R_1 ignores updates to N from R_2 for some period of time.



Eventually, R_1 sends the update to R_2.

RIP Timers

- Update
  - Time between each update: 30s with small random offset to avoid synchronization problems
- Timeout
  - If no updates are received, mark entry for deletion. It is then announced as unreachable: metric 16. Default: 180s.
- Garbage-collection
  - The entry is purged from the table – no longer announced. Default: 120s.
- Triggered-update timers
  - 1s – 5s (random)
  - Canceled by update timer

EIGRP
(Enhanced Interior Gateway Routing Protocol) – Part I

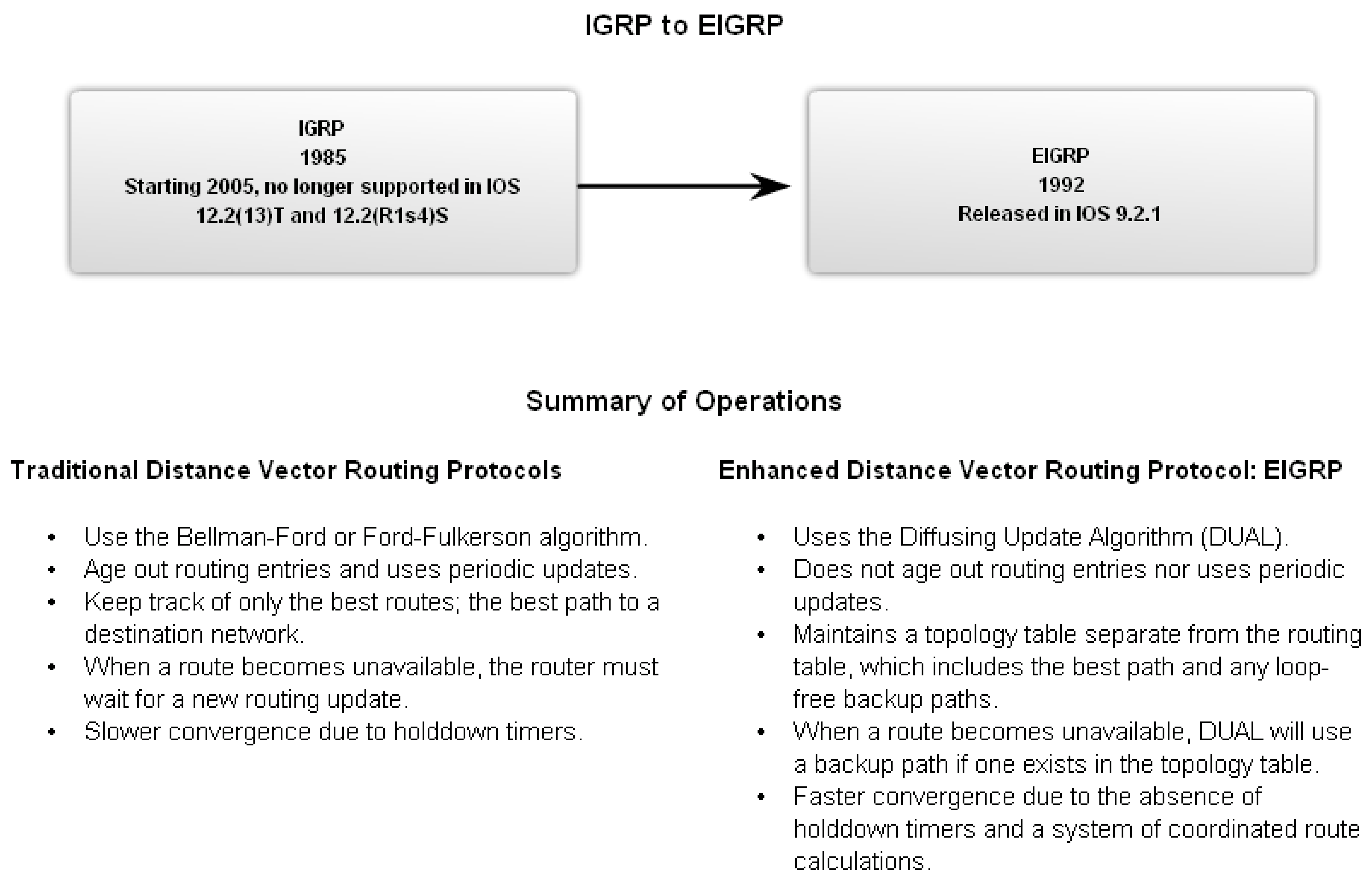
Introduction (1)

- A classless version of IGRP.
- EIGRP includes several features that are not commonly found in other distance vector routing protocols like RIP (RIPv1 and RIPv2) and IGRP.
- These features include:
  - Reliable Transport Protocol (RTP)
  - Bounded Updates
  - Diffusing Update Algorithm (DUAL)
  - Establishing Adjacencies
  - Neighbor and Topology Tables
- Although EIGRP may act like a link-state routing protocol, it is still a distance vector routing protocol.

Introduction (2)

- Note: The term hybrid routing protocol is sometimes used to define EIGRP.
- However, this term is misleading because EIGRP is not a hybrid between distance vector and link-state routing protocols
- it is solely a distance vector routing protocol. Therefore, Cisco is no longer using this term to refer to EIGRP.

EIGRP VS IGRP



IGRP to EIGRP

IGRP
1985
Starting 2005, no longer supported in IOS 12.2(13)T and 12.2(R1s4)S

EIGRP
1992
Released in IOS 9.2.1

Summary of Operations

Traditional Distance Vector Routing Protocols

- Use the Bellman-Ford or Ford-Fulkerson algorithm.
- Age out routing entries and uses periodic updates.
- Keep track of only the best routes; the best path to a destination network.
- When a route becomes unavailable, the router must wait for a new routing update.
- Slower convergence due to holddown timers.

Enhanced Distance Vector Routing Protocol: EIGRP

- Uses the Diffusing Update Algorithm (DUAL).
- Does not age out routing entries nor uses periodic updates.
- Maintains a topology table separate from the routing table, which includes the best path and any loop-free backup paths.
- When a route becomes unavailable, DUAL will use a backup path if one exists in the topology table.
- Faster convergence due to the absence of holddown timers and a system of coordinated route calculations.

The Algorithm

- Traditional distance vector routing protocols all use some variant of the Bellman-Ford or Ford-Fulkerson algorithm.
- These protocols, such as RIP and IGRP, age out individual routing entries, and therefore need to periodically send routing table updates.

- EIGRP uses the Diffusing Update Algorithm (DUAL).
- EIGRP does not send periodic updates and route entries do not age out.
- Instead, EIGRP uses a lightweight Hello protocol to monitor connection status with its neighbors.
- Only changes in the routing information, such as a new link or a link becoming unavailable cause a routing update to occur.

Path Determination (1)

- Traditional distance vector routing protocols such as RIP and IGRP keep track of only the preferred routes; the best path to a destination network.
- If the route becomes unavailable, the router waits for another routing update with a path to this remote network.
- EIGRP's DUAL maintains a topology table separate from the routing table.
  - including both the best path to a destination network and any backup paths that DUAL has determined to be loop-free.
- Loop-free means that the neighbor does not have a route to the destination network that passes through this router.

Path Determination (2)

- If a route becomes unavailable, DUAL will search its topology table for a valid backup path.
- If one exists, that route is immediately entered into the routing table.
- If one does not exist, DUAL performs a network discovery process to see if there happens to be a backup path that did not meet the requirement of the feasibility condition.

Convergence

- Traditional distance vector routing protocols such as RIP and IGRP use periodic updates.
- Due to the unreliable nature of periodic updates, traditional distance vector routing protocols are prone to routing loops and the count-to-infinity problem.
- RIP and IGRP use several mechanisms to help avoid these problems including holddown timers, which cause long convergence times.
- EIGRP does not use holddown timers. Instead, loop-free paths are achieved through a system of route calculations (diffusing computations) that are performed in a coordinated fashion among the routers.

Holddown timer

- **Holddown timer** works by having each router start a **timer** when they first receive information about a network that is unreachable. Until the **timer** expires, the router will discard any subsequent route messages that indicate the route is in fact reachable.

RTP and EIGRP Packet types

- Reliable Transport Protocol (RTP) is the protocol used by EIGRP for the delivery and reception of EIGRP packets.

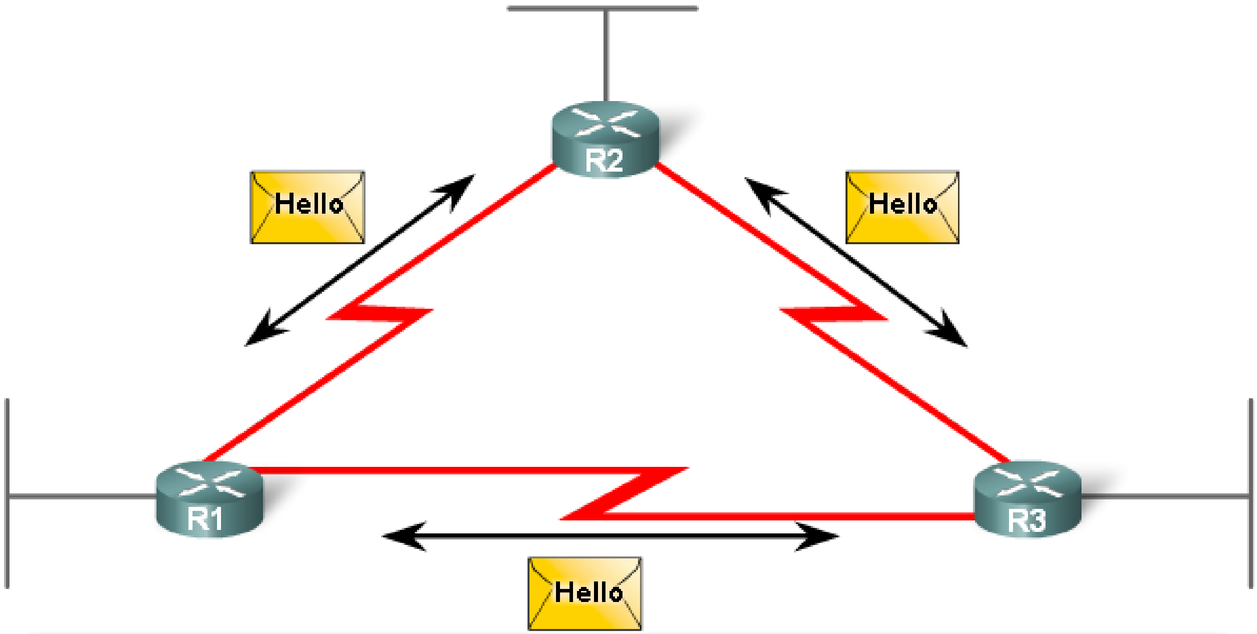
- Reliable RTP requires an acknowledgement to be returned by the receiver to the sender.
- An unreliable RTP packet does not require an acknowledgement.

- RTP can send packets either as a unicast or a multicast.
- Multicast EIGRP packets use the reserved multicast address of 224.0.0.10.

RTP and EIGRP Packet types (Cont.)

- EIGRP uses five different packet types, some in pairs.
- Hello packets
- Update packets
- Acknowledgement (ACK) packets
- Query and reply packets

Hello packets



Hello packet

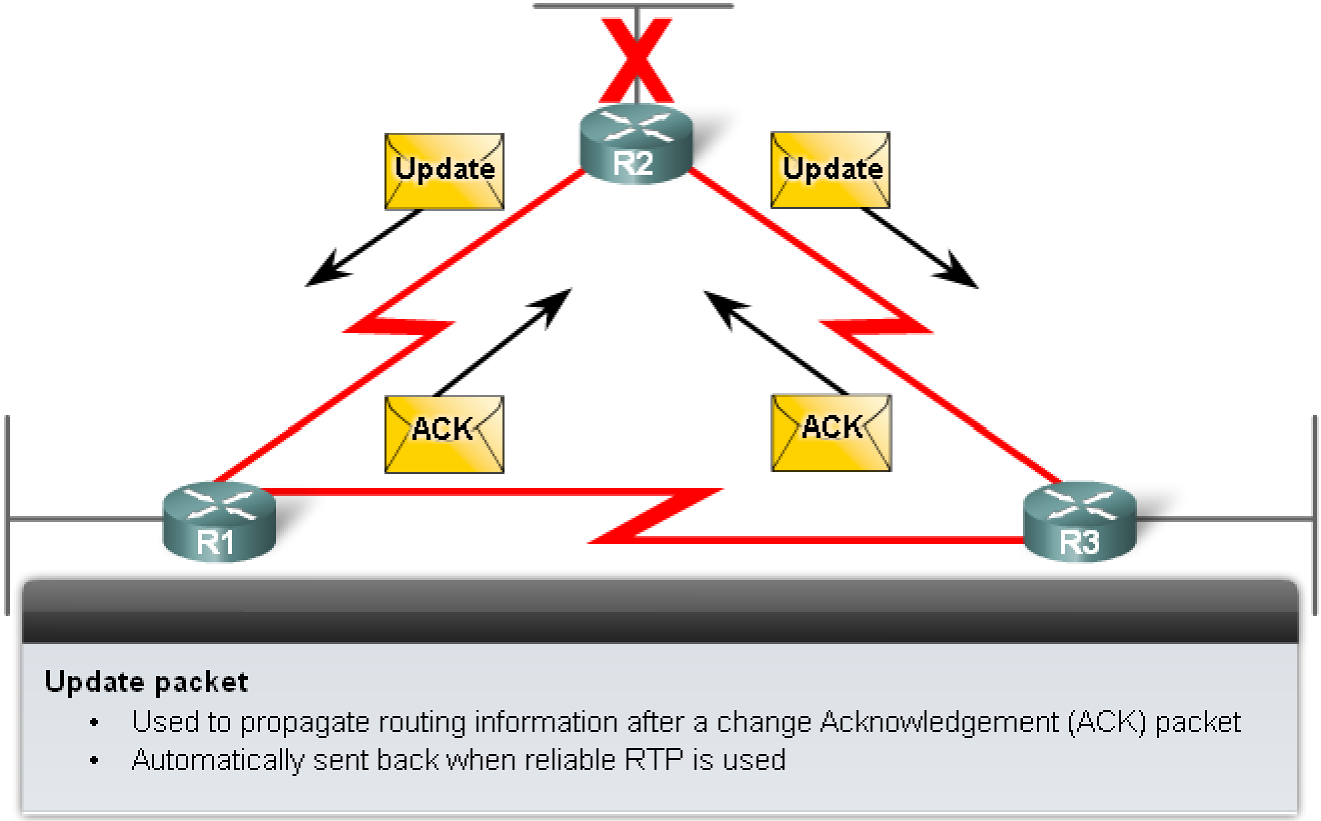
- Use to discover neighbors & form adjacencies
- Unreliable so no response required from recipient

Hello packets are used by EIGRP to discover neighbors and to form adjacencies with those neighbors. EIGRP hello packets are multicasts and use unreliable delivery.

Update packets

- Update packets are used by EIGRP to propagate routing information.
- Unlike RIP, EIGRP does not send periodic updates. Update packets are sent only when necessary.
- EIGRP updates contain only the routing information needed and are sent only to those routers that require it. EIGRP update packets use reliable delivery.
- Update packets are sent as a multicast when required by multiple routers, or as a unicast when required by only a single router.
- In the figure, because the links are point-to-point, the updates are sent as unicasts.

Update and ACK packets



Update packet

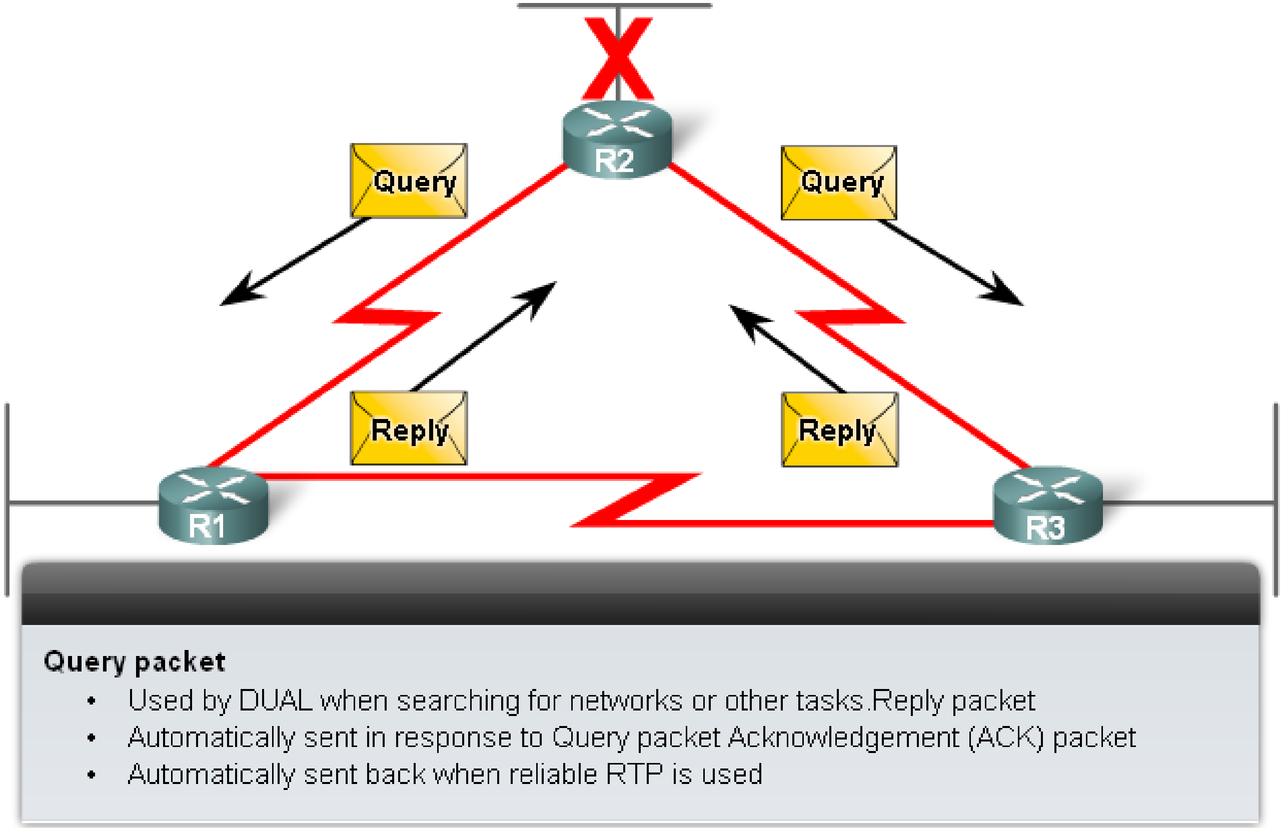
- Used to propagate routing information after a change Acknowledgement (ACK) packet
- Automatically sent back when reliable RTP is used

Acknowledgement (ACK) packets

- Acknowledgement (ACK) packets are sent by EIGRP when reliable delivery is used.
- RTP uses reliable delivery for EIGRP update, query, and reply packets.
- EIGRP acknowledgement packets are always sent as an unreliable unicast (unreliable delivery).

- In the figure, R2 has lost connectivity to the LAN attached to its FastEthernet interface.
- R2 immediately sends an Update to R1 and R3 noting the downed route.
- R1 and R3 respond with an acknowledgement.

Query and reply packets (1)



Query packet

- Used by DUAL when searching for networks or other tasks.Reply packet
- Automatically sent in response to Query packet Acknowledgement (ACK) packet
- Automatically sent back when reliable RTP is used

Query and reply packets are used by DUAL when searching for networks and other tasks. Queries and replies use reliable delivery. Queries can use multicast or unicast, whereas replies are always sent as unicast.

Query and reply packets (2)

- In the figure, R2 has lost connectivity to the LAN and it sends out queries to all EIGRP neighbors searching for any possible routes to the LAN.
- Because queries use reliable delivery, the receiving router must return an EIGRP acknowledgement.
- (To keep this example simple, acknowledgements were omitted in the graphic.)
- All neighbors must send a reply regardless of whether or not they have a route to the downed network.
- Because replies also use reliable delivery, routers such as R2, must send an acknowledgement.

EIGRP

Query	**Update**	**Reply**	**Hello**	**Acknowledge**
Reliable	**Reliable**	**Reliable**	**Unreliable** **(not require acknowledgment)**	**Unreliable** **(a hello packet that has no data)**
multicast	Multicast & unicast	unicast	multicast	unicast

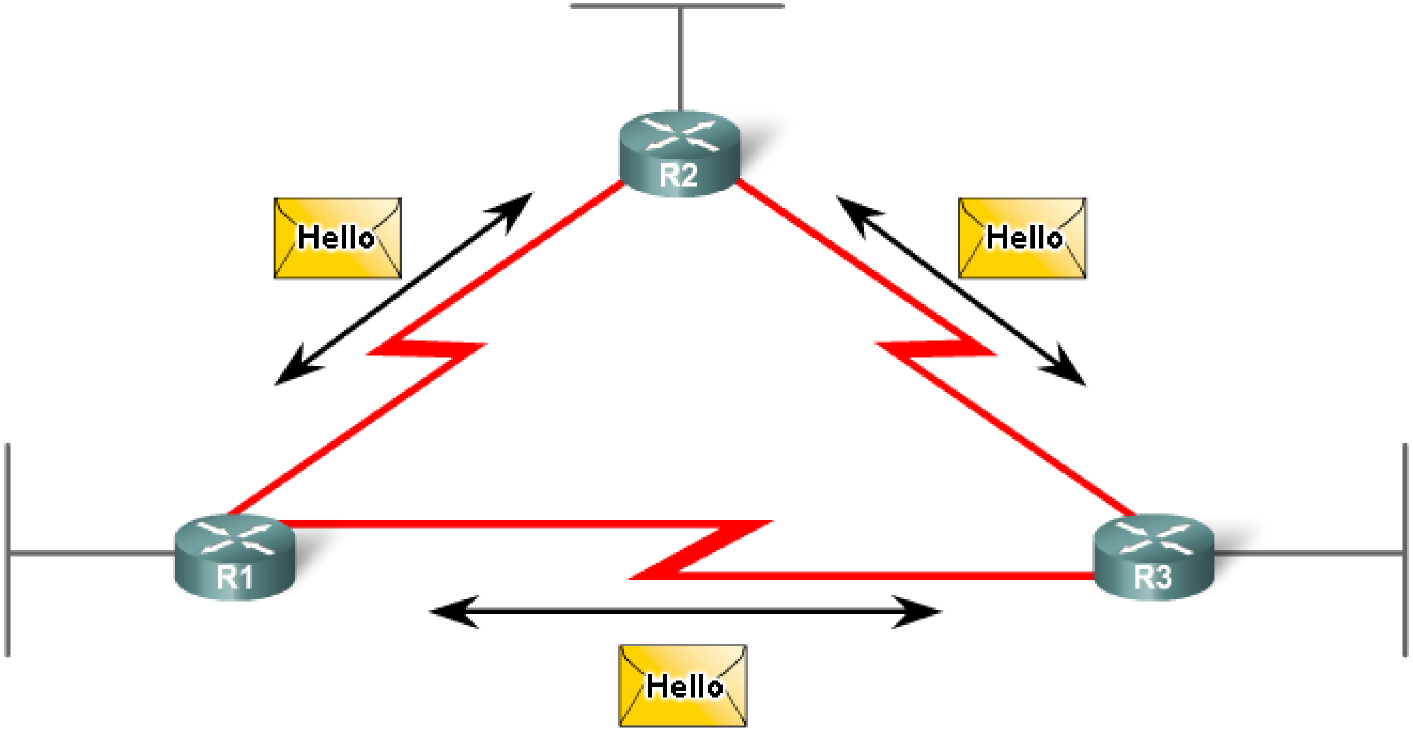
Hello Protocol (1)

- EIGRP routers discover neighbors and establish adjacencies with neighbor routers using the Hello packet.
- On most networks, EIGRP Hello packets are sent every 5 seconds.
- On multipoint nonbroadcast multiaccess networks (NBMA) such as X.25, Frame Relay, and ATM interfaces with access links of T1 (1.544 Mbps) or slower, Hellos are unicast every 60 seconds.
- An EIGRP router assumes that as long as it is receiving Hello packets from a neighbor, the neighbor and its routes remain viable.

Hello Protocol (2)

- Holdtime tells the router the maximum time the router should wait to receive the next Hello before declaring that neighbor as unreachable.
- By default, the hold time is three times the Hello interval, or 15 seconds on most networks and 180 seconds on low speed NBMA networks. If the hold time expires, EIGRP will declare the route as down and DUAL will search for a new path by sending out queries.

Default Hello Intervals and Hold Times for EIGRP



Bandwidth	Example Link	Default Hello Interval	Default Hold Time
1.544 Mbps	Multipoint Frame Relay	60 seconds	180 seconds
Greater than 1.544 Mbps	T1, Ethernet	5 seconds	15 seconds

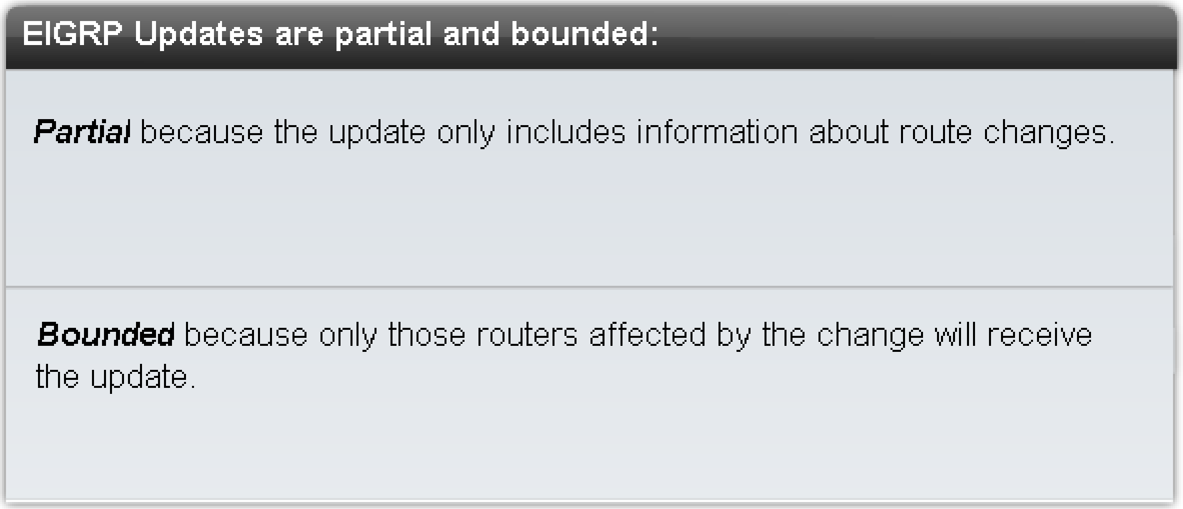
EIGRP Bounded update (1)

- EIGRP uses the term partial or bounded when referring to its update packets.
- Unlike RIP, EIGRP does not send periodic updates.
- Instead, EIGRP sends its updates only when the metric for a route changes.
- The term partial means that the update only includes information about the route changes.
- EIGRP sends these incremental updates when the state of a destination changes, instead of sending the entire contents of the routing table.

EIGRP Bounded update (2)

- The term bounded refers to the propagation of partial updates sent only to those routers that are affected by the change.
- The partial update is automatically "bounded" so that only those routers that need the information are updated.
- By sending only the routing information that is needed and only to those routers that need it, EIGRP minimizes the bandwidth required to send EIGRP packets.

EIGRP Bounded update (3)



EIGRP Updates are partial and bounded:

Partial because the update only includes information about route changes.

Bounded because only those routers affected by the change will receive the update.

Diffusing Update Algorithm (DUAL) (1)

- Routing loops can be extremely detrimental to network performance.
- Distance vector routing protocols such as RIP prevent routing loops with hold-down timers, split horizon.
- Although EIGRP uses both of these techniques, it uses them somewhat differently; the primary way that EIGRP prevents routing loops is with the DUAL algorithm.
- The DUAL algorithm is used to obtain loop-freedom at every instant throughout a route computation.
- This allows all routers involved in a topology change to synchronize at the same time.

Diffusing Update Algorithm (DUAL) (2)

- Routers that are not affected by the topology changes are not involved in the recomputation.
- This method provides EIGRP with faster convergence times than other distance vector routing protocols.
- The decision process for all route computations is done by the DUAL Finite State Machine.
- A finite state machine (FSM) is a model of behavior composed of a finite number of states, transitions between those states, and events or actions that create the transitions

Diffusing Update Algorithm (DUAL) (3)

- The DUAL FSM tracks all routes, uses its metric to select efficient, loop-free paths, and selects the routes with the least cost path to insert into the routing table.
- Because recomputation of the DUAL algorithm can be processor-intensive, it is advantageous to avoid recomputation whenever possible. Therefore, DUAL maintains a list of backup routes it has already determined to be loop-free.
- If the primary route in the routing table fails, the best backup route is immediately added to the routing table.

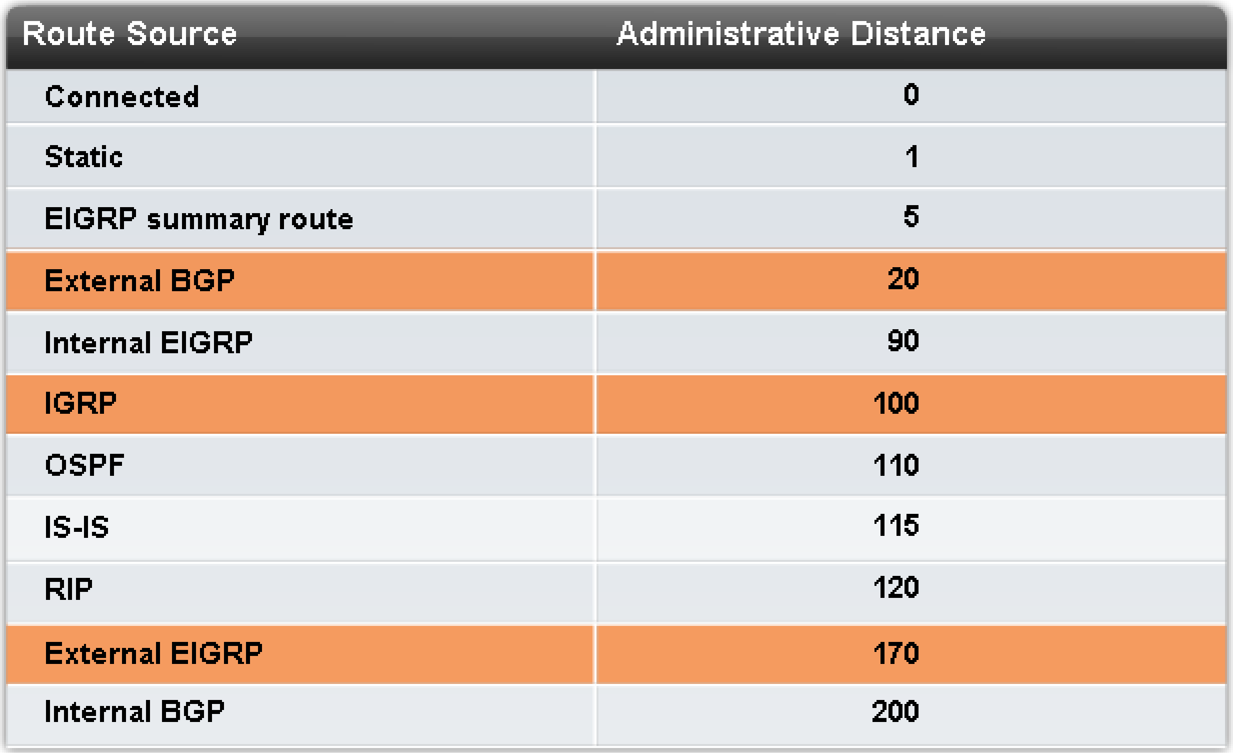
Administrative Distance (AD) (1)

- Administrative distance (AD) is the trustworthiness (or preference) of the route source.

- EIGRP has a default administrative distance of 90 for internal routes and 170 for routes imported from an external source, such as default routes.

- When compared to other interior gateway protocols (IGPs), EIGRP is the most preferred by the Cisco IOS because it has the lowest administrative distance.

Administrative Distance (AD) (2)



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

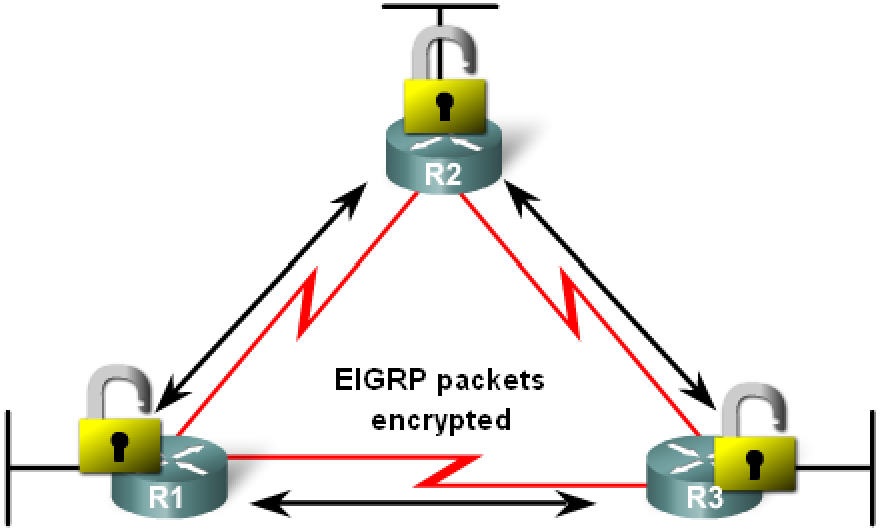
Authentication (1)

- EIGRP can be configured for authentication.
- RIPv2, EIGRP, OSPF, IS-IS, and BGP can all be configured to encrypt and authenticate their routing information.

- It is good practice to authenticate transmitted routing information.
- This practice ensures that routers will only accept routing information from other routers that have been configured with the same password or authentication information.

- Note: Authentication does not encrypt the router's routing table.

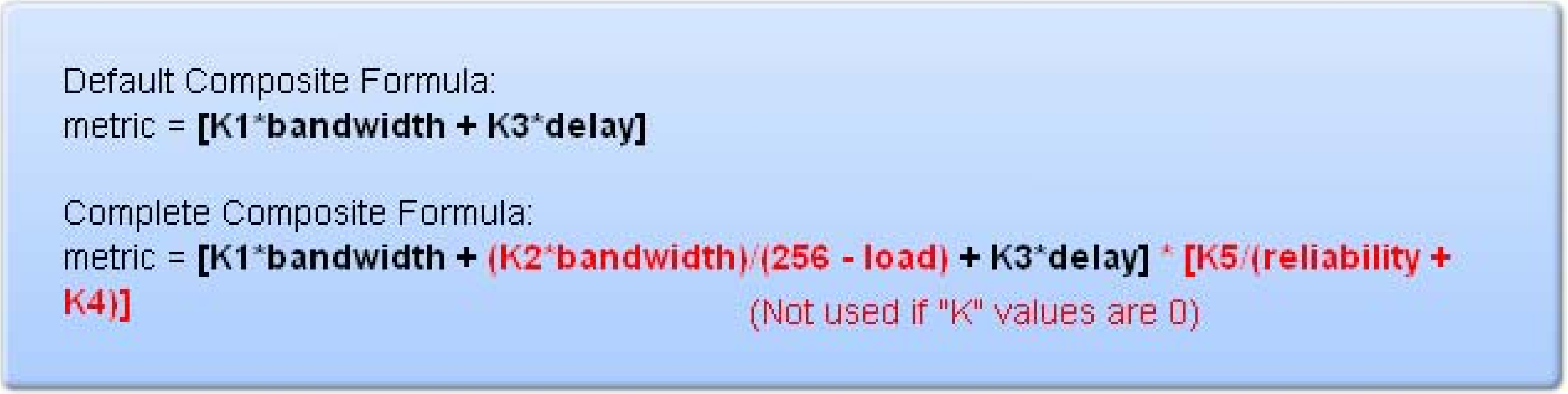
Authentication (2)



EIGRP Metric calculation

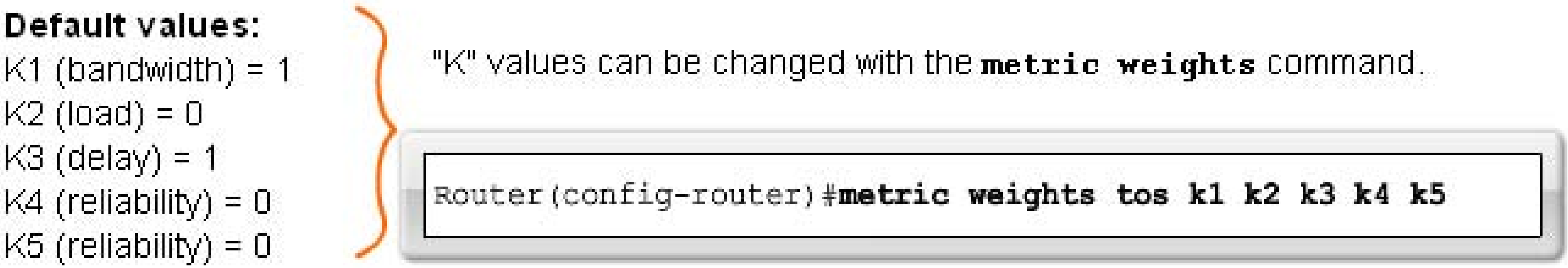
- EIGRP uses the following values in its composite metric to calculate the preferred path to a network:
  - Bandwidth
  - Delay
  - Reliability
  - Load
- By default, only bandwidth and delay are used to calculate the metric.
- Cisco recommends that reliability and load are not used unless the administrator has an explicit need to do so.

EIGRP composite metric



Default Composite Formula:
metric = **[K1*bandwidth + K3*delay]**

Complete Composite Formula:
metric = **[K1*bandwidth + (K2*bandwidth)/(256 - load) + K3*delay] * [K5/(reliability + K4)]**
(Not used if "K" values are 0)



Default values:
K1 (bandwidth) = 1
K2 (load) = 0
K3 (delay) = 1
K4 (reliability) = 0
K5 (reliability) = 0

"K" values can be changed with the **metric weights** command.

```
Router(config-router)#metric weights tos k1 k2 k3 k4 k5
```

The tos (Type of Service) value is left over from IGRP
and was never implemented. The tos value is always set to 0.

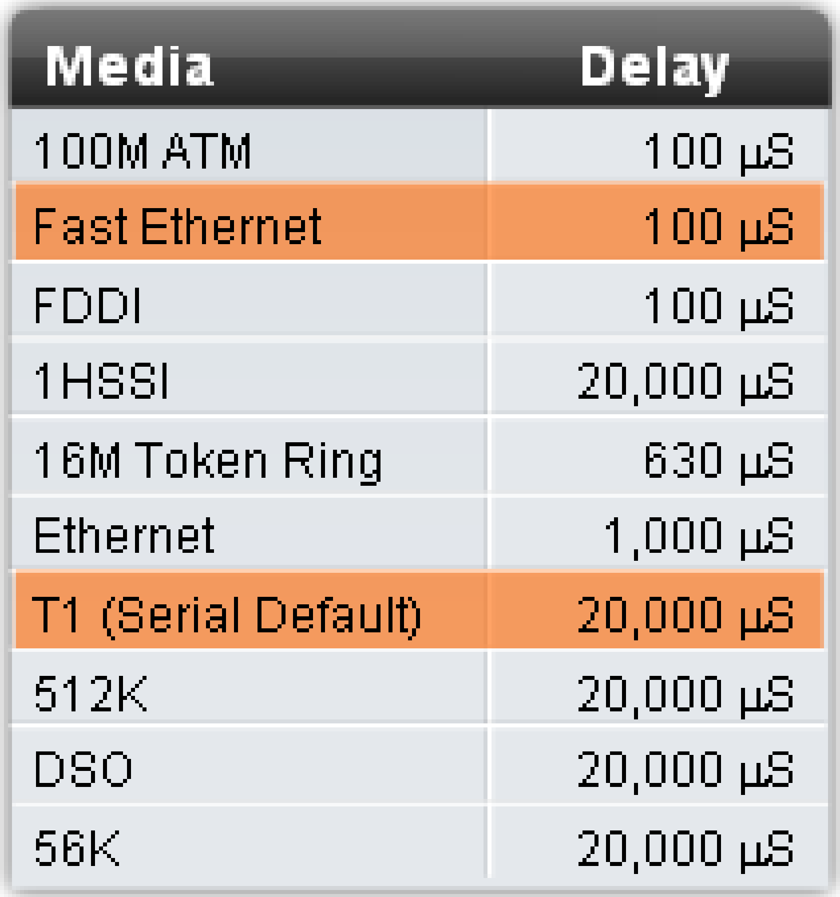
EIGRP Metrics (2)

- The bandwidth metric (1544 Kbit) is a static value used by some routing protocols such as EIGRP and OSPF to calculate their routing metric.
- The bandwidth is displayed in Kbit (kilobits).
- Most serial interfaces use the default bandwidth value of 1544 Kbit or 1,544,000 bps (1.544 Mbps).
- This is the bandwidth of a T1 connection.
- The value of the bandwidth may or may not reflect the actual physical bandwidth of the interface.
- Modifying the bandwidth value does not change the actual bandwidth of the link

EIGRP Metrics (3)

- Delay is a measure of the time it takes for a packet to traverse a route.
- The delay (DLY) metric is a static value based on the type of link to which the interface is connected and is expressed in microseconds.
- Delay is not measured dynamically. In other words, the router does not actually track how long packets are taking to reach the destination.
- The delay value, much like the bandwidth value, is a default value that can be changed by the network administrator.

EIGRP Metrics (4)



Media	Delay
100M ATM	100 µS
Fast Ethernet	100 µS
FDDI	100 µS
1HSSI	20,000 µS
16M Token Ring	630 µS
Ethernet	1,000 µS
T1 (Serial Default)	20,000 µS
512K	20,000 µS
DS0	20,000 µS
56K	20,000 µS

- The table in the figure shows the default delay values for various interfaces.
- Notice that the default value is 20,000 microseconds for Serial interfaces and 100 microseconds for FastEthernet interfaces.

EIGRP Metrics (5)

- Reliability is a measure of the probability that the link will fail or how often the link has experienced errors.
- Unlike delay, Reliability is measured dynamically with a value between 0 and 255, with 1 being a minimally reliable link and 255 one hundred percent reliable.
- Reliability is calculated on a 5-minute weighted average to avoid the sudden impact of high (or low) error rates.
- Reliability is expressed as a fraction of 255 - the higher the value, the more reliable the link.
- So, 255/255 would be 100 percent reliable, whereas a link of 234/255 would be 91.8 percent reliable.

EIGRP Metrics (6)

- Load reflects the amount of traffic utilizing the link.
- Like reliability, load is measured dynamically with a value between 0 and 255.
- Similar to reliability, load is expressed as a fraction of 255.
- However, in this case a lower load value is more desirable because it indicates less load on the link.
- So, 1/255 would be a minimally loaded link. 40/255 is a link at 16 percent capacity, and 255/255 would be a link that is 100 percent saturated.

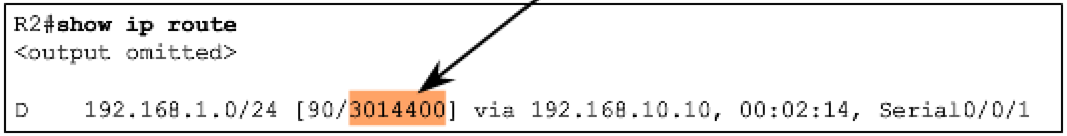
Metric calculation (1) -- IMPORTANT

Default metric = [K1*bandwidth + K3*delay] * 256

Since K1 and K3 both equal 1, the formula simplifies to: **bandwidth + delay**

bandwidth = speed of slowest link in route to the destination
delay = sum of the delays of each link in route to the destination

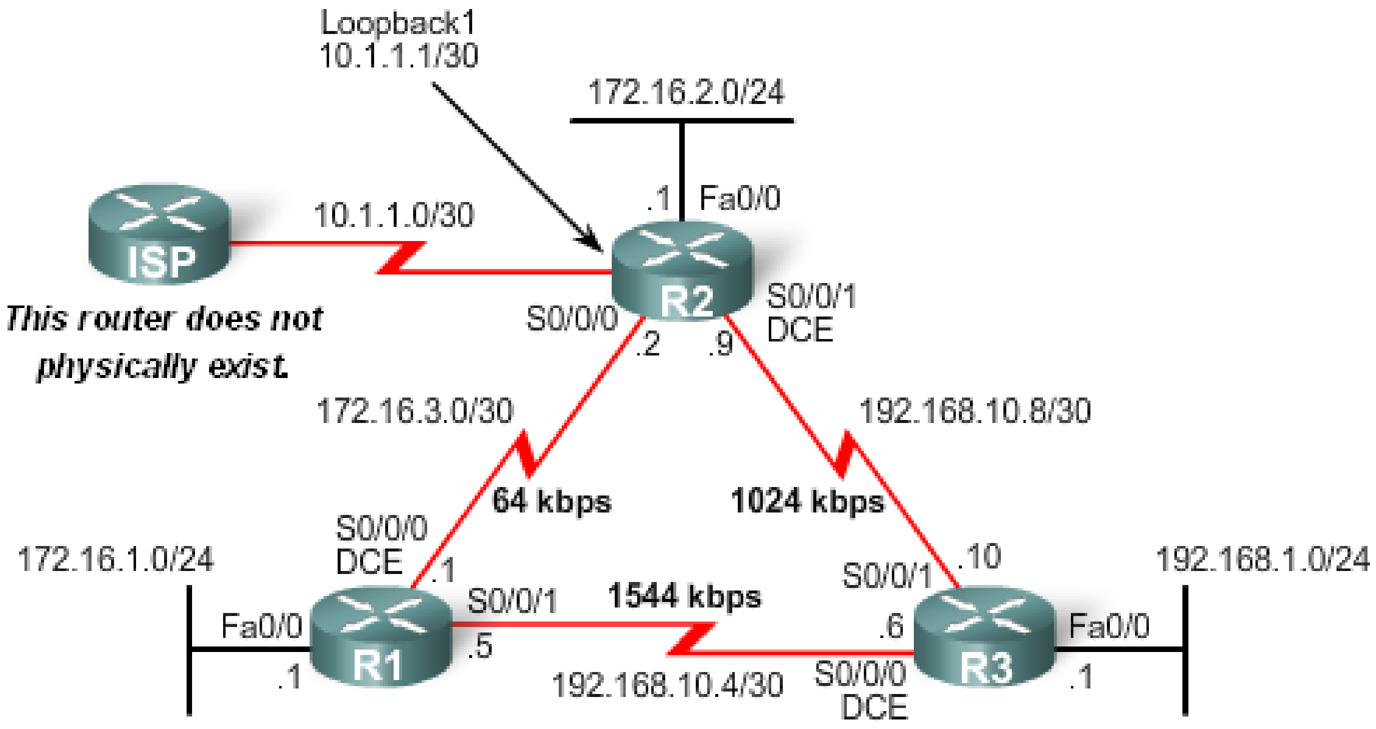
```
Slowest bandwidth:             (10,000,000/bandwidth kbps)  *  256
Plus the sum of the delays:  + (sum of delay/10)  *  256
                             =  EIGRP metric
```



```
R2#show ip route
<output omitted>

D     192.168.1.0/24 [90/3014400] via 192.168.10.10, 00:02:14, Serial0/0/1
```

Metric calculation (2)

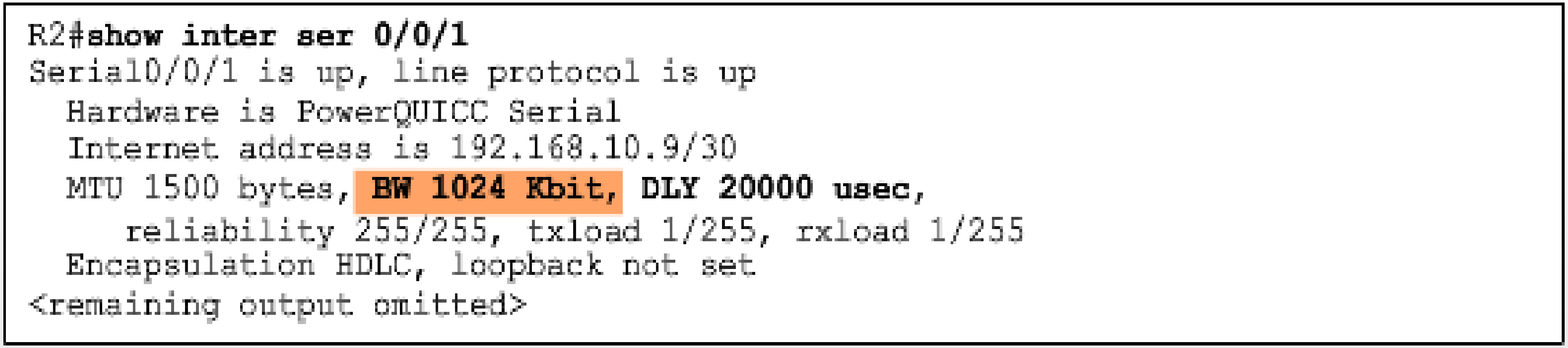
- The routing table output for R2 shows that the route to 192.168.1.0/24 has an EIGRP metric of 3,014,400



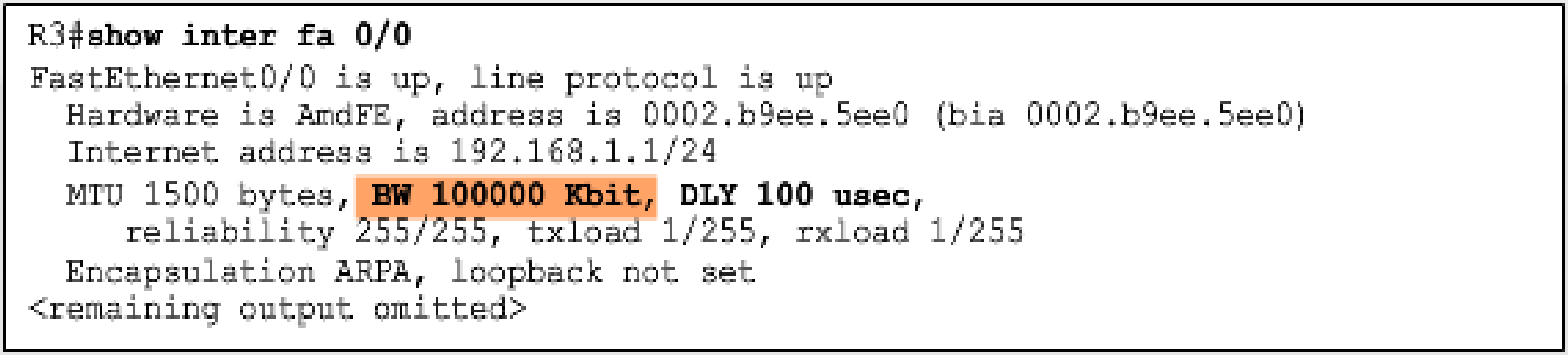
Metric calculation – bandwidth (1)

- Because EIGRP uses the slowest bandwidth in its metric calculation, we can find the slowest bandwidth by examining each interface between R2 and the destination network 192.168.1.0.
- The Serial 0/0/1 interface on R2 has a bandwidth of 1,024 Kbps or 1,024,000 bps.
- The FastEthernet 0/0 interface on R3 has a bandwidth of 100,000 Kbps or 100 Mbps.
- Therefore, the slowest bandwidth is 1024 Kbps and is used in the calculation of the metric.

Metric calculation – bandwidth (2)



```
R2#show inter ser 0/0/1
Serial0/0/1 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 192.168.10.9/30
  MTU 1500 bytes, BW 1024 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
<remaining output omitted>
```



```
R3#show inter fa 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0002.b9ee.5ee0 (bia 0002.b9ee.5ee0)
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
<remaining output omitted>
```

bandwidth = (10,000,000/1024) = 9765 * 256 = 2499840

Metric calculation – bandwidth (3)

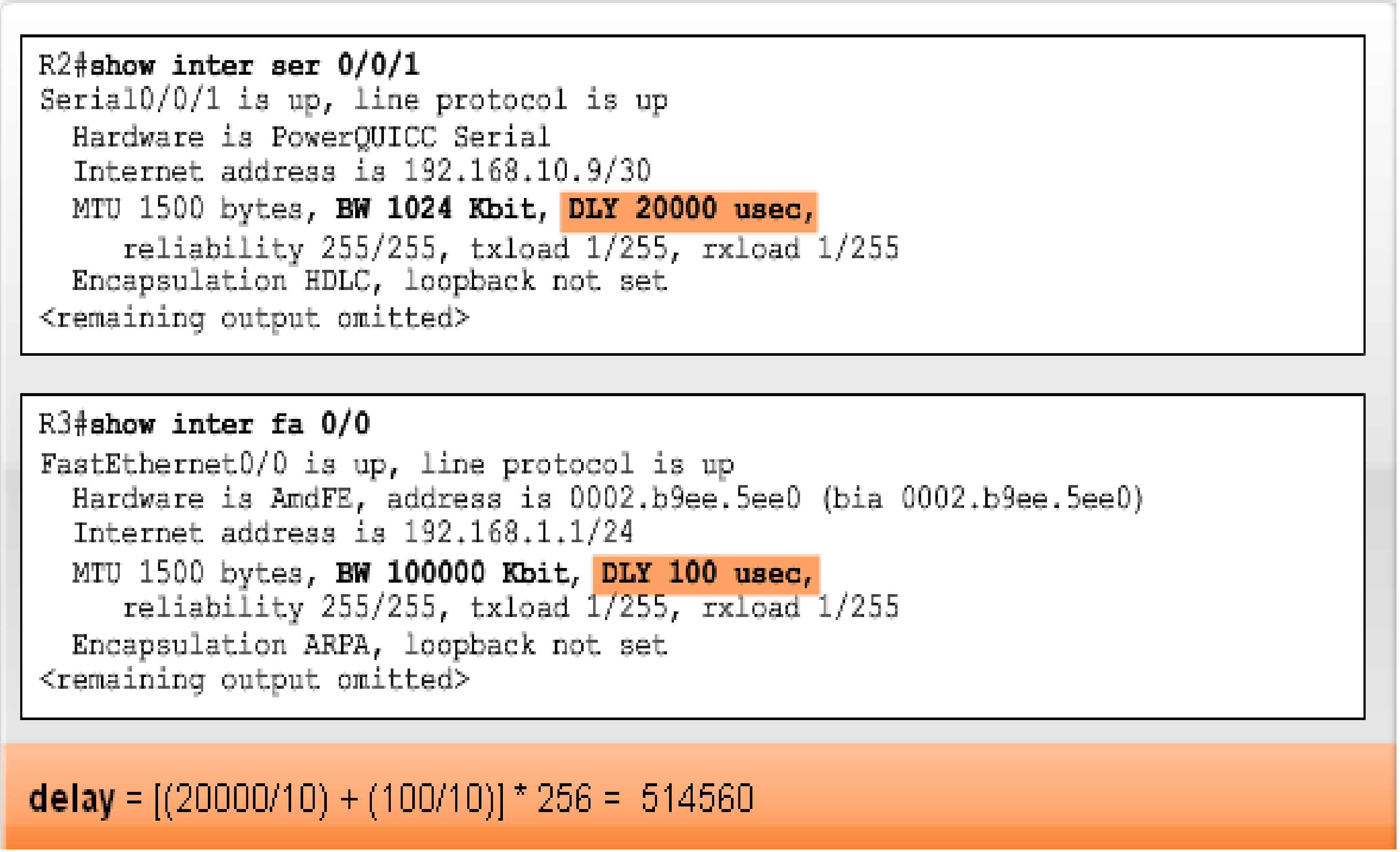
- EIGRP takes the bandwidth value in kbps and divides it by a reference bandwidth value of 10,000,000.
- This will result in higher bandwidth values receiving a lower metric and lower bandwidth values receiving a higher metric.

- In this case, 10,000,000 divided by 1024 equals 9765.625.
- The .625 is dropped before multiplying by 256. The bandwidth portion of the composite metric is 2,499,840.

Metric calculation – delay (1)

- EIGRP uses the cumulative sum of delay metrics of all of the outgoing interfaces.
  - The Serial 0/0/1 interface on R2 has a delay of 20000 microseconds.
  - The FastEthernet 0/0 interface on R3 has a delay of 100 microseconds.
- Each delay value is divided by 10 and then summed.
  - 20,000/10 + 100/10 results in a value of 2,010.
  - This result is then multiplied by 256.
  - The delay portion of the composite metric is 514,560.

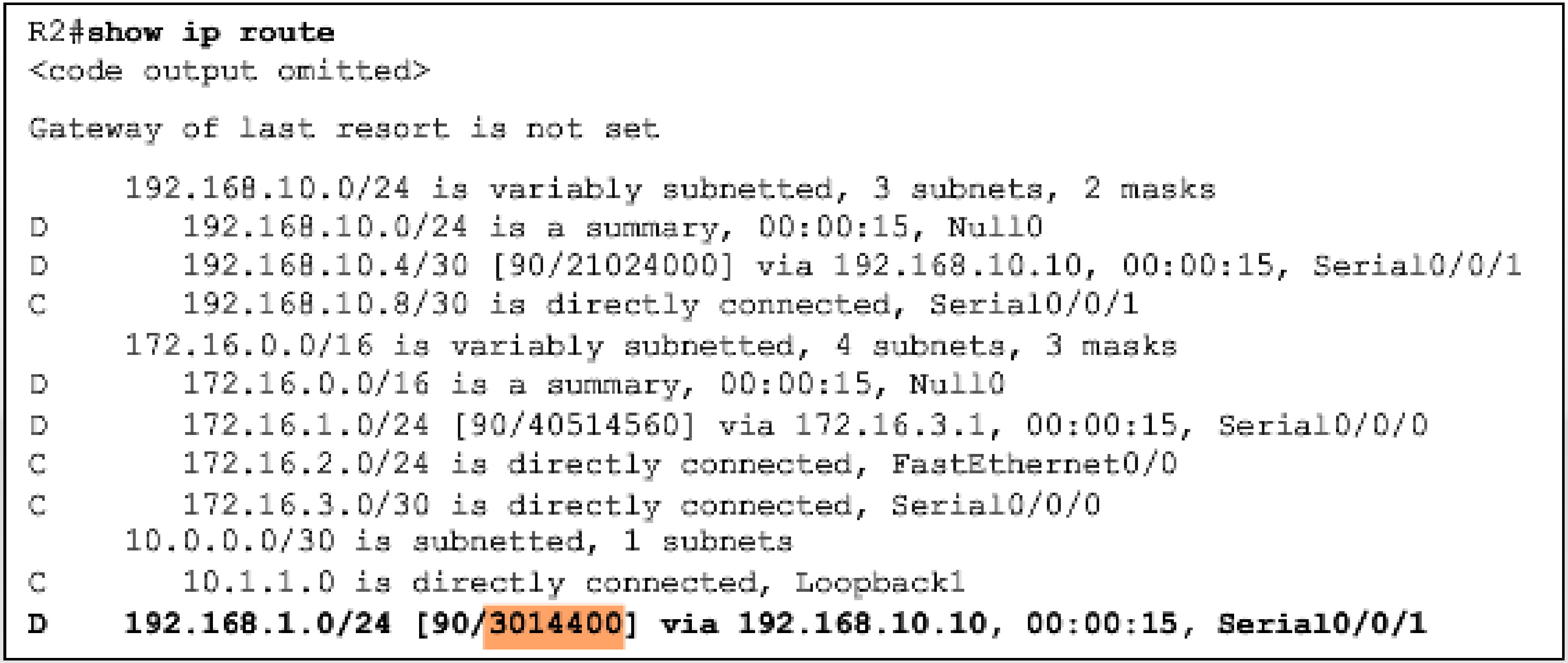
Metric calculation – delay (2)



```
R2#show inter ser 0/0/1
Serial0/0/1 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 192.168.10.9/30
  MTU 1500 bytes, BW 1024 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
<remaining output omitted>
```

```
R3#show inter fa 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0002.b9ee.5ee0 (bia 0002.b9ee.5ee0)
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
<remaining output omitted>
```

delay = [(20000/10) + (100/10)] * 256 = 514560



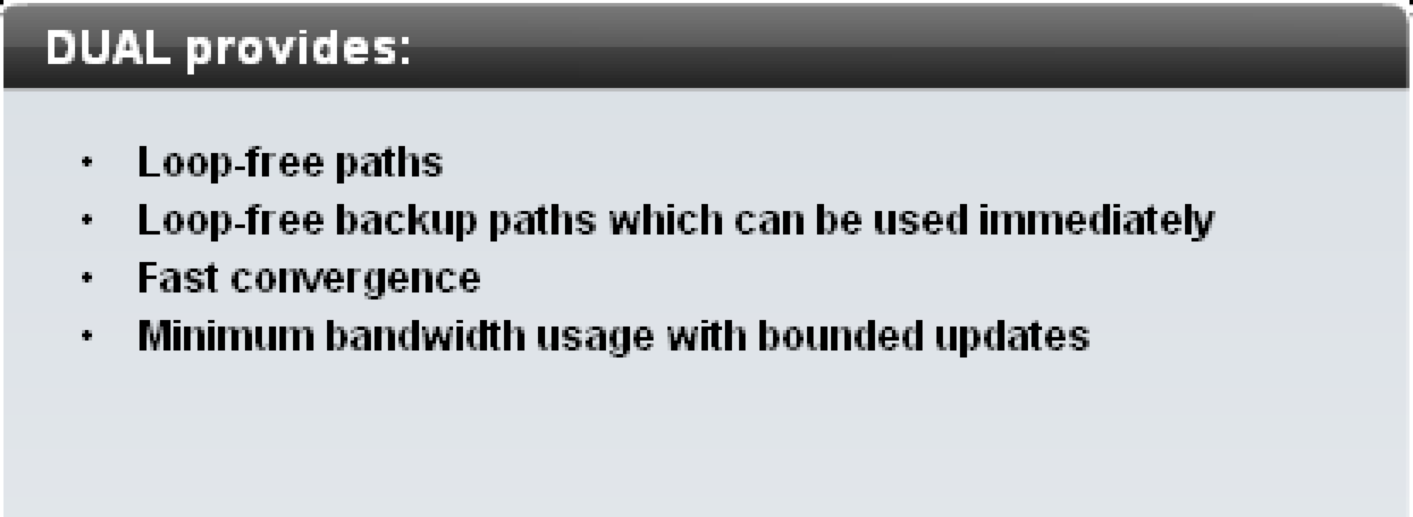
```
R2#show ip route
<code output omitted>

Gateway of last resort is not set

     192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
D       192.168.10.0/24 is a summary, 00:00:15, Null0
D       192.168.10.4/30 [90/21024000] via 192.168.10.10, 00:00:15, Serial0/0/1
C       192.168.10.8/30 is directly connected, Serial0/0/1
     172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
D       172.16.0.0/16 is a summary, 00:00:15, Null0
D       172.16.1.0/24 [90/40514560] via 172.16.3.1, 00:00:15, Serial0/0/0
C       172.16.2.0/24 is directly connected, FastEthernet0/0
C       172.16.3.0/30 is directly connected, Serial0/0/0
     10.0.0.0/30 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Loopback1
D    192.168.1.0/24 [90/3014400] via 192.168.10.10, 00:00:15, Serial0/0/1
```

EIGRP Metric = bandwidth + delay = 2499840 + 514560 = 3014400

- Simply add the two values together, 2,499,840 + 514,560, to obtain the EIGRP metric of 3,014,400.
- This is a result of the slowest bandwidth and the sum of the delays

EIGRP
(Enhanced Interior Gateway Routing Protocol) – Part II

DUAL concepts



DUAL provides:

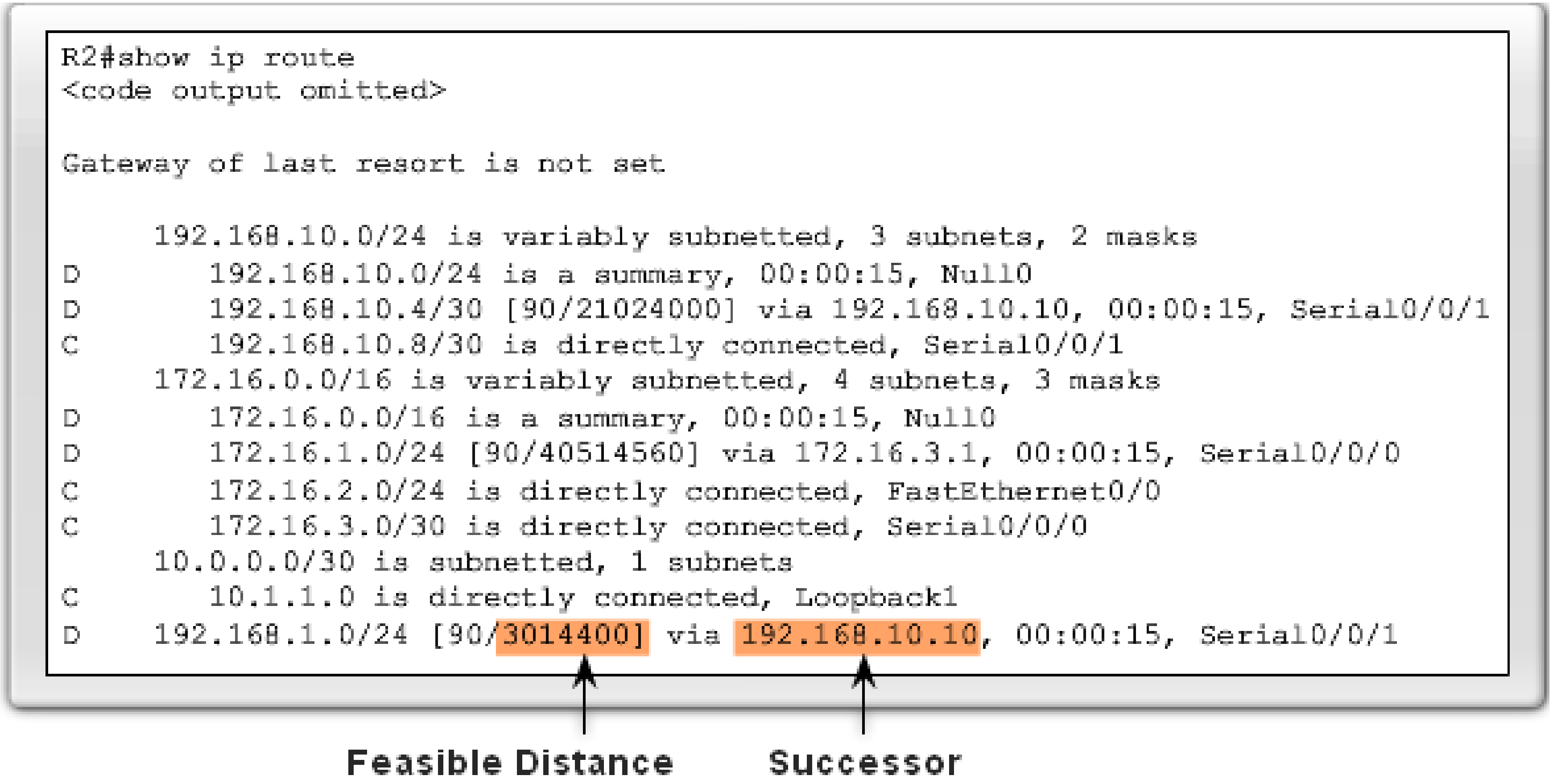
- Loop-free paths
- Loop-free backup paths which can be used immediately
- Fast convergence
- Minimum bandwidth usage with bounded updates

- DUAL uses several terms which will be discussed in more detail:
- Successor
- Feasible Distance (FD)
- Feasible Successor (FS)
- Reported Distance (RD) or Advertised Distance (AD)
- Feasible Condition or Feasibility Condition (FC)

Successor and Feasible Distance (1)

- A successor is a neighboring router that is used for packet forwarding and is the least-cost route to the destination network.
  - The IP address of a successor is shown in a routing table entry right after the word via.
- Feasible distance (FD) is the lowest calculated metric to reach the destination network.
  - FD is the metric listed in the routing table entry as the second number inside the brackets.
  - As with other routing protocols this is also known as the metric for the route.

Successor and Feasible Distance (2)

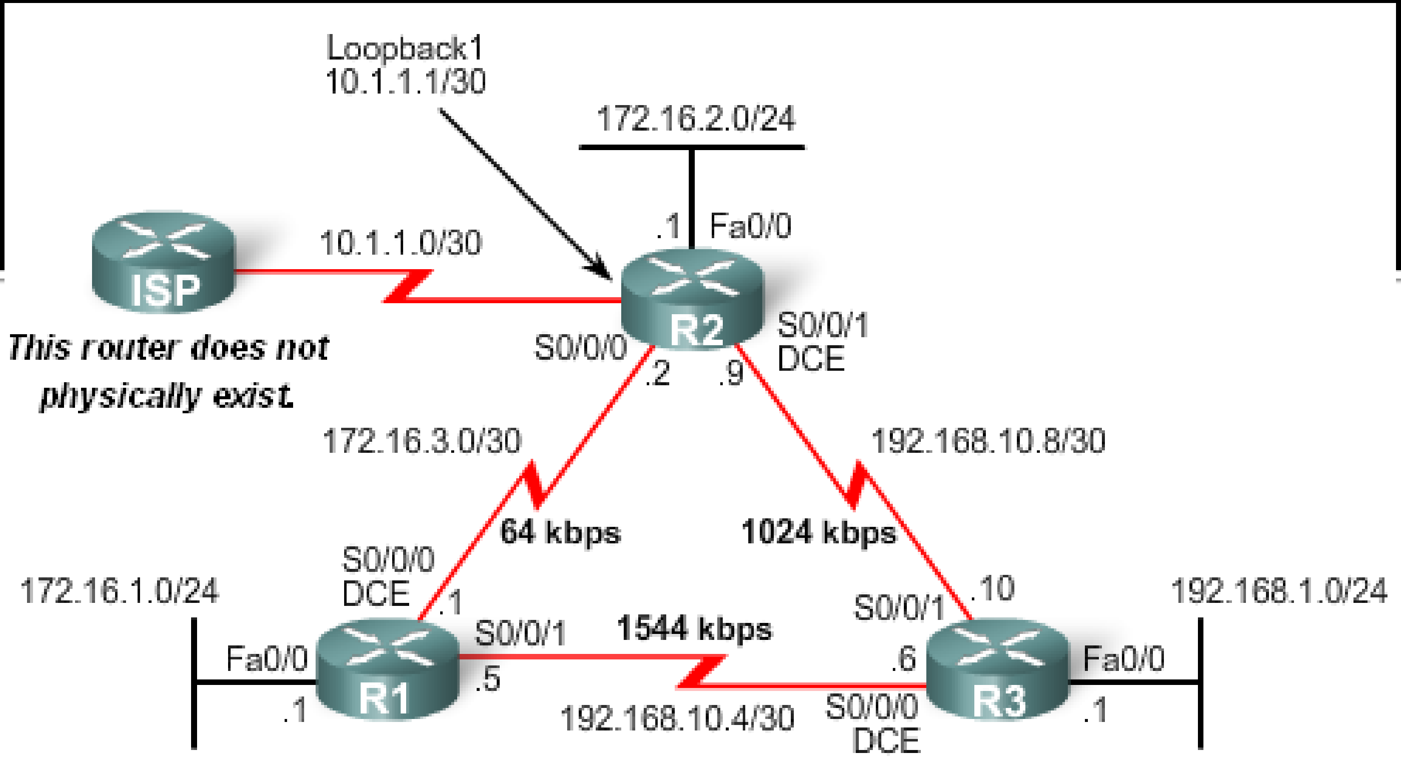


```
R2#show ip route
<code output omitted>

Gateway of last resort is not set

     192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
D       192.168.10.0/24 is a summary, 00:00:15, Null0
D       192.168.10.4/30 [90/21024000] via 192.168.10.10, 00:00:15, Serial0/0/1
C       192.168.10.8/30 is directly connected, Serial0/0/1
     172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
D       172.16.0.0/16 is a summary, 00:00:15, Null0
D       172.16.1.0/24 [90/40514560] via 172.16.3.1, 00:00:15, Serial0/0/0
C       172.16.2.0/24 is directly connected, FastEthernet0/0
C       172.16.3.0/30 is directly connected, Serial0/0/0
     10.0.0.0/30 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Loopback1
D    192.168.1.0/24 [90/3014400] via 192.168.10.10, 00:00:15, Serial0/0/1
```

R3 at 192.168.10.10 is the successor for network 192.168.1.0/24. This route has a feasible distance of 3014400.

Feasible Successor

- One of the reasons DUAL can converge quickly after a change in the topology is because it can use backup paths to other routers known as feasible successors without having to recompute DUAL.
- A feasible successor (FS) is a neighbor who has a loop-free backup path to the same network as the successor by satisfying the feasibility condition.



- In our topology, would R2 consider R1 to be a feasible successor to network 192.168.1.0/24?
- In order to be a feasible successor, R1 must satisfy the feasibility condition (FC).

Feasibility condition (FC)

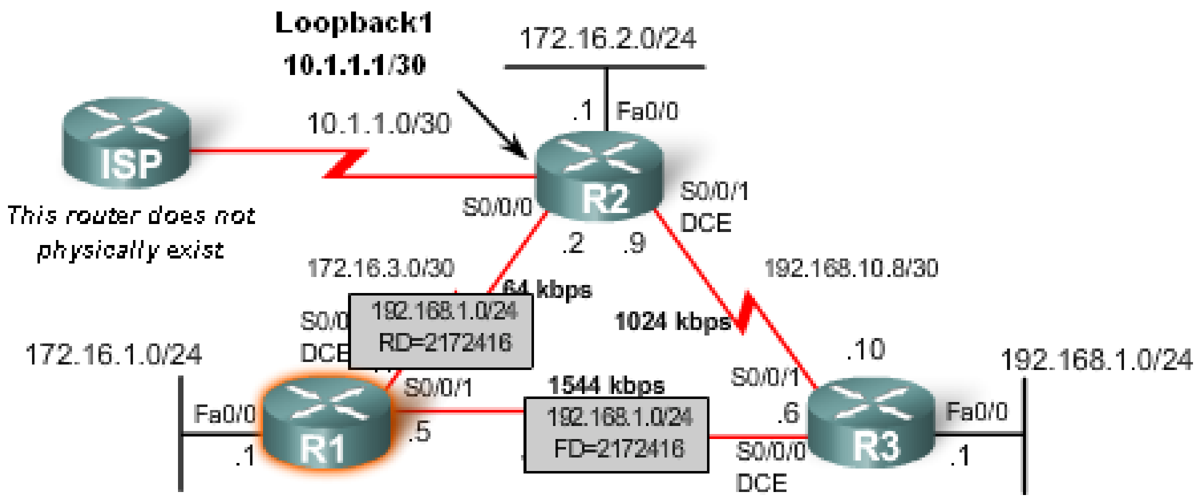
- The feasibility condition (FC) is met when a neighbor's reported distance (RD) to a network is less than the local router's feasible distance to the same destination network.
- The reported distance or advertised distance is simply an EIGRP neighbor's feasible distance to the same destination network.

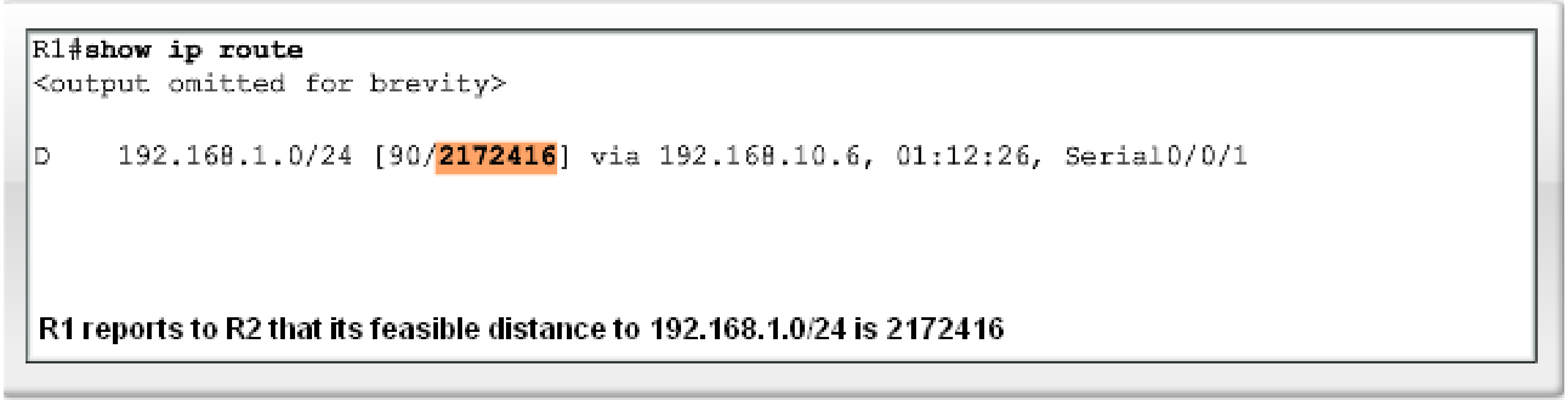
Reported distance (RD) (1)

- The reported distance is the metric that a router reports to a neighbor about its own cost to that network.
  - In the figure, R1 is reporting to R2 that its feasible distance to 192.168.1.0/24 is 2172416.
  - From R2's perspective, 2172416 is R1's reported distance. From R1's perspective, 2172416 is its feasible distance.

Reported distance (RD) (2)

Does R1 satisfy the feasibility condition?

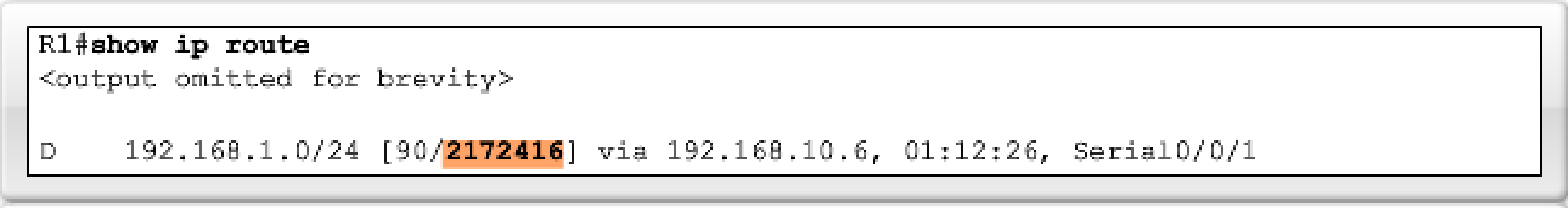




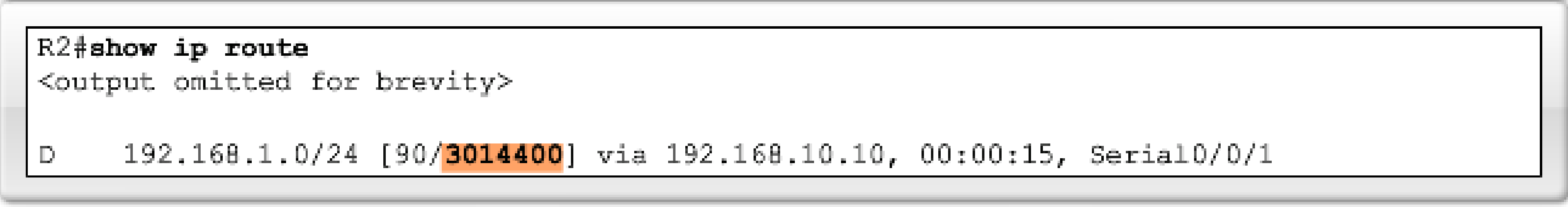
```
R1#show ip route
<output omitted for brevity>

D    192.168.1.0/24 [90/2172416] via 192.168.10.6, 01:12:26, Serial0/0/1
```

R1 reports to R2 that its feasible distance to 192.168.1.0/24 is 2172416

- R2 examines the reported distance (RD) of 2172416 from R1.
- Because the reported distance (RD) of R1 is less than R2's own feasible distance (FD) of 3014400, R1 meets the feasibility condition.
- R1 is now a feasible successor for R2 to the 192.168.1.0/24 network.



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

D    192.168.1.0/24 [90/2172416] via 192.168.10.6, 01:12:26, Serial0/0/1
```

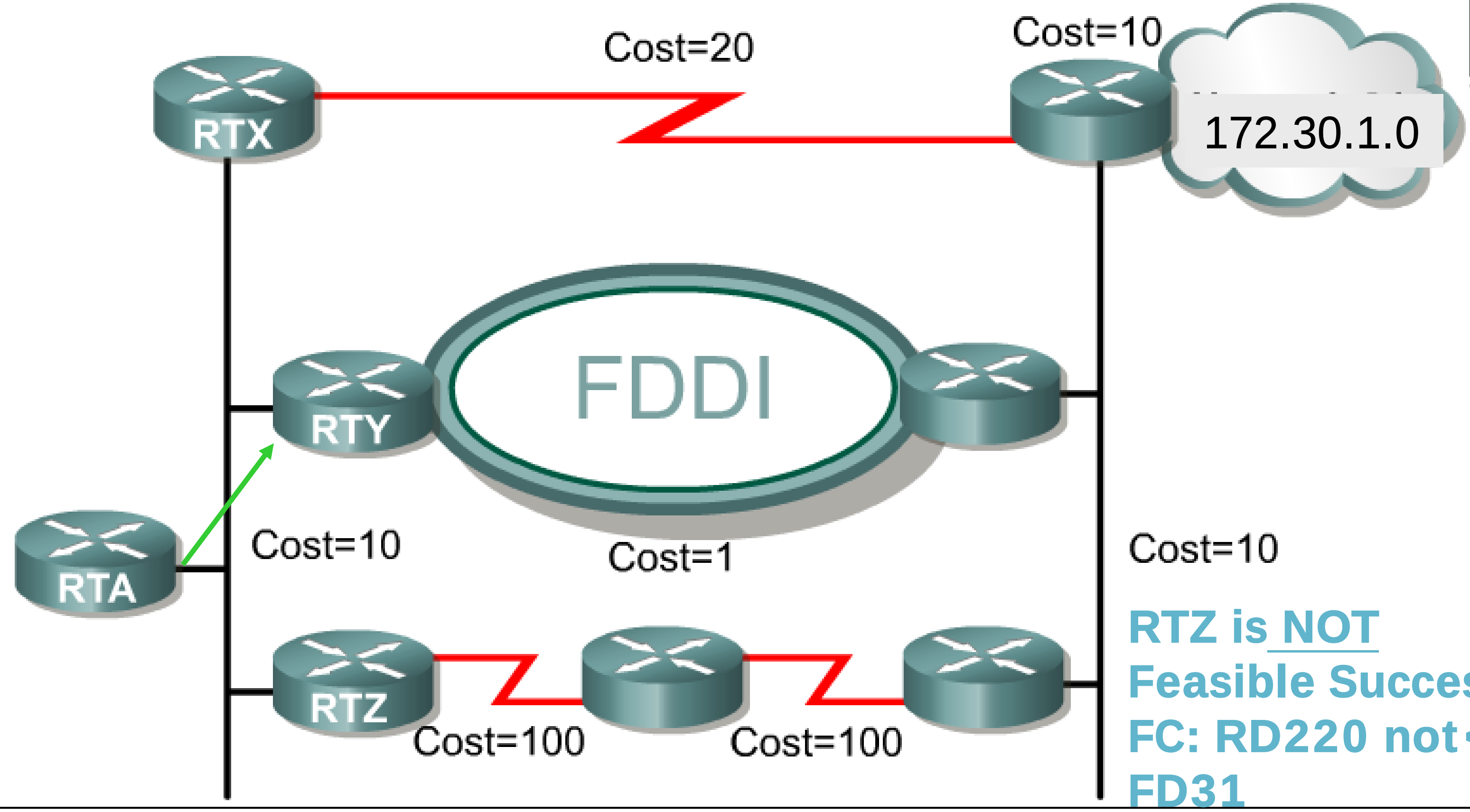


```
R2#show ip route
<output omitted for brevity>

D    192.168.1.0/24 [90/3014400] via 192.168.10.10, 00:00:15, Serial0/0/1
```

Why isn't R1 the successor if its reported distance (RD) is less than R2's feasible distance (FD) to 192.168.1.0/24?

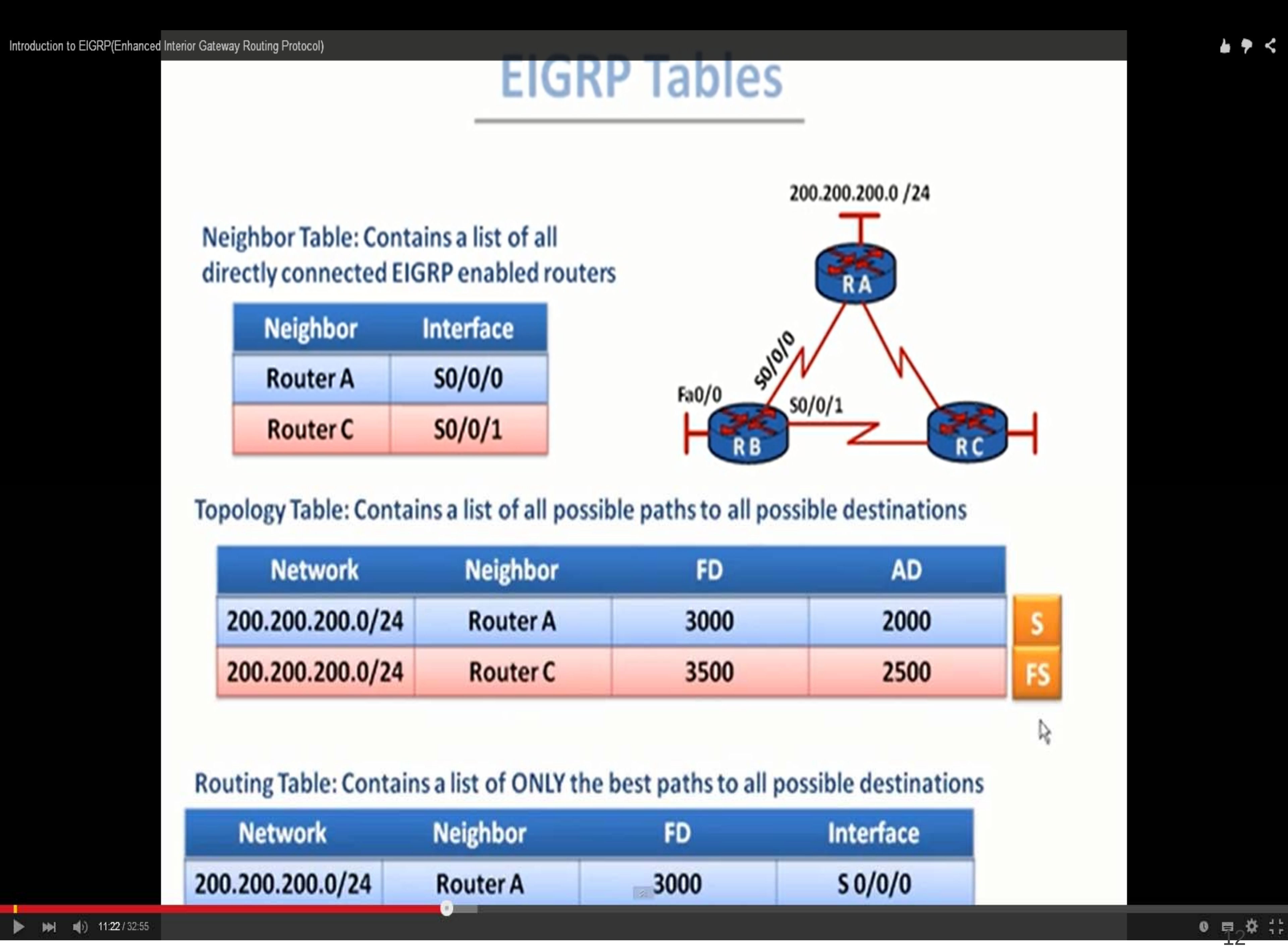
EIGRP technologies (cont.)

Feasible Successor, FC: RD30 < FD31



RTZ is NOT Feasible Successor, FC: RD220 not< FD31

Destination	Feasible Dist.	Advertised or Reported. Dist.	Neighbor	
172.30.1.0	40	30	X	In Topology Table
172.30.1.0	**31**	21	Y	In Routing Table
172.30.1.0	230	220	Z	Not in Topology Table


Introduction to EIGRP(Enhanced Interior Gateway Routing Protocol)
200.200.200.0 /24
RA
S0/0/0
Fa0/0
S0/0/1
RB
RC
11:22 / 32:55

EIGRP Tables

Neighbor Table: Contains a list of all directly connected EIGRP enabled routers

Neighbor	Interface
Router A	S0/0/0
Router C	S0/0/1

Topology Table: Contains a list of all possible paths to all possible destinations

Network	Neighbor	FD	AD	
200.200.200.0/24	Router A	3000	2000	S
200.200.200.0/24	Router C	3500	2500	FS

Routing Table: Contains a list of ONLY the best paths to all possible destinations

Network	Neighbor	FD	Interface
200.200.200.0/24	Router A	3000	S 0/0/0

Show ip eigrp neighbors

<table>
<tr><th colspan="9">IP EIGRP neighbors for process 1</th></tr>
<tr><th>H</th><th>Address</th><th>Interface</th><th>Hold sec</th><th>Uptime</th><th>SRTT (ms)</th><th>RTP</th><th>Q cnt</th><th>Seq type num</th></tr>
<tr><td>1</td><td>192.168.1.1</td><td>Se0/0</td><td>10</td><td>00:01:41</td><td>20</td><td>200</td><td>0</td><td>7</td></tr>
<tr><td>0</td><td>172.16.1.1</td><td>Se0/1</td><td>10</td><td>00:08:24</td><td>25</td><td>200</td><td>0</td><td>28</td></tr>
</table>

Order in which neighbours were learned

Show ip eigrp neighbors

IP EIGRP neighbors for process 1

H	Address	Interface	Hold sec	Uptime	SRTT (ms)	RTP	Q cnt	Seq type num
1	192.168.1.1	Se0/0	10	00:01: 41	20	200	0	7
0	172.16.1.1	Se0/1	10	00:08: 24	25	200	0	28

Address of neighbour

Show ip eigrp neighbors

IP EIGRP neighbors for process 1								
H	Address	Interface	Hold sec	Uptime	SRTT (ms)	RTP	Q cnt	Seq type num
1	192.168.1.1	Se0/0	10	00:01:41	20	200	0	7
0	172.16.1.1	Se0/1	10	00:08:24	25	200	0	28

Interface that connects to neighbour

Show ip eigrp neighbors

IP EIGRP neighbors for process 1								
H	Address	Interface	Hold sec	Uptime	SRTT (ms)	RTP	Q cnt	Seq type num
1	192.168.1.1	Se0/0	10	00:01:41	20	200	0	7
0	172.16.1.1	Se0/1	10	00:08:24	25	200	0	28

Time remaining before neighbour is considered down. Set to maximum when Hello arrives.

Show ip eigrp neighbors

IP EIGRP neighbors for process 1

H	Address	Interface	Hold sec	Uptime	SRTT (ms)	RTP	Q cnt	Seq type num
1	192.168.1.1	Se0/0	10	00:01:41	20	200	0	7
0	172.16.1.1	Se0/1	10	00:08:24	25	200	0	28

How long neighbour has been adjacent.

Show ip eigrp neighbor

IP EIGRP neighbors for process 1

H	Address	Interface	Hold sec	Uptime	SRTT (ms)	RTO	Q cnt	Seq type num
1	192.168.1.1	Se0/0	10	00:01:41	20	200	0	7
0	172.16.1.1	Se0/1	10	00:08:24	25	200	0	28

Used in reliable transport

Tracks updates, queries etc

Neighbor Table

```
RouterC#show ip eigrp neighbors
IP-EIGRP neighbors for process 44
H   Address         Interface    Hold Uptime    SRTT   RTO  Q   Seq
                                 (sec)          (ms)        Cnt Num
0   192.168.0.1     Se0            11 00:03:09 1138   5000  0   6
1   192.168.1.2     Et0            12 00:34:46    4    200  0   4
```

- *Neighbor address* The network-layer address of the neighbor router(s).
- *Queue count* The number of packets waiting in queue to be sent. If this value is constantly higher than zero, then there may be a congestion problem at the router. A zero means that there are no EIGRP packets in the queue.

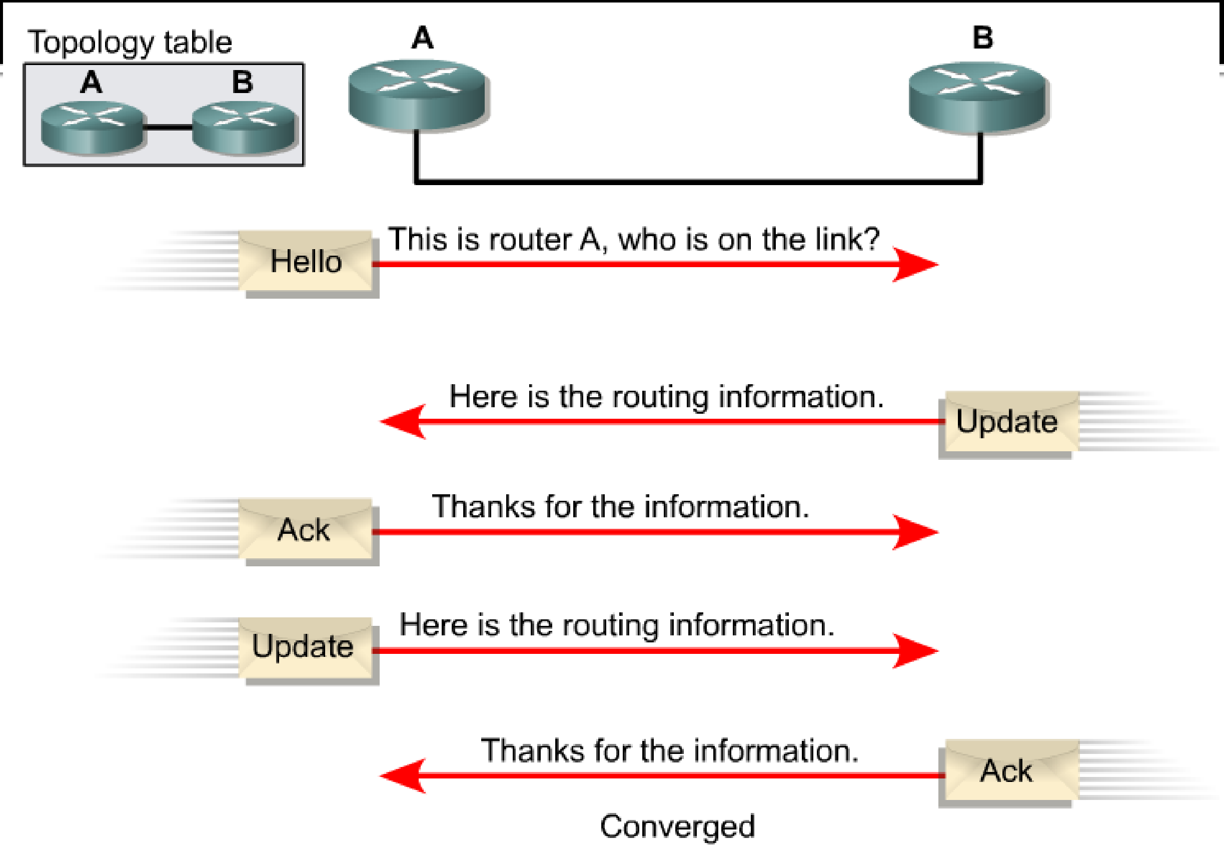
Neighbor Table

```
RouterC#show ip eigrp neighbors
IP-EIGRP neighbors for process 44
H   Address         Interface     Hold Uptime   SRTT   RTO  Q   Seq
                                  (sec)         (ms)        Cnt Num
0   192.168.0.1     Se0             11 00:03:09 1138   5000  0   6
1   192.168.1.2     Et0             12 00:34:46    4    200  0   4
```

- ***Smooth Round Trip Timer (SRTT)*** The average time it takes to send and receive packets from a neighbor.
    - This timer is used to determine the retransmit interval (RTO)

- ***Hold Time*** The interval to wait without receiving anything from a neighbor before considering the link unavailable.
    - Originally, the expected packet was a hello packet, but in current Cisco IOS software releases, **any EIGRP packets received after the first hello will reset the timer**.

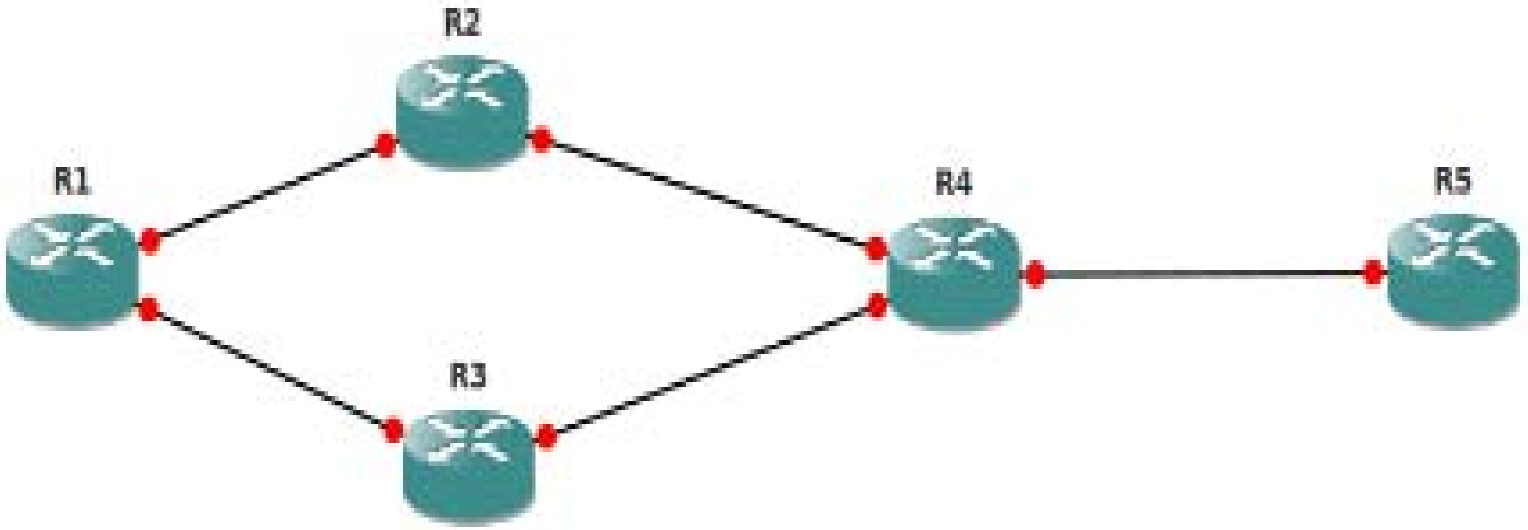
Establishing Adjacencies with Neighbors (Revisited)


Topology table
A
B
A
B
Hello
This is router A, who is on the link?
Here is the routing information.
Update
Ack
Thanks for the information.
Update
Here is the routing information.
Thanks for the information.
Ack
Converged

By forming adjacencies, EIGRP routers do the following:

- Dynamically learn of new routes that join their network
- Identify routers that become either unreachable or inoperable
- Rediscover routers that had previously been unreachable

Example of a loop: Consider the following topology.



❑ R1 is our router which has a route to R5.

❑ Suppose the path through R2 is better (the lowest FD), so R2 is becoming our Successor for that route.

❑ Now R1 has to decide if R3 will become a Feasible Successor or not. Let's split this into 2 cases.

Example of a loop (Cont.):

1) Suppose R1's distance to R5 (which is FD) is 100, whereas R3's distance to R5 through R4 is 120 (which is RD). Let's violate the feasibility condition and choose R3 as a Feasible Successor.

- What happens when we configure unequal load balancing? R1 sends some IP packets to R5 through R3. When a packet comes to the next hop, R3 thinks "I need to send it towards R5, so let's consult my routing table". And guess what? The routing table will tell R3 to send the packet back to R1 because R1 is the Successor for that route, in R3's point of view (remember that distance on the left is 100+X, and 120 on the right; X is distance between R1 and R3 which may easily be less than 20). So we have a loop, as the packet can bounce between R1 and R3 never going to other routers.

2) Suppose R1's distance to R5 (which is FD) is 100, whereas R3's distance to R5 through R4 is 95 (which is RD). Now whatever distance X is between R1 and R3, 100+X always greater than 95, so R3 will never route through R1. This is why it's a loop free path.

What if the successor fails?

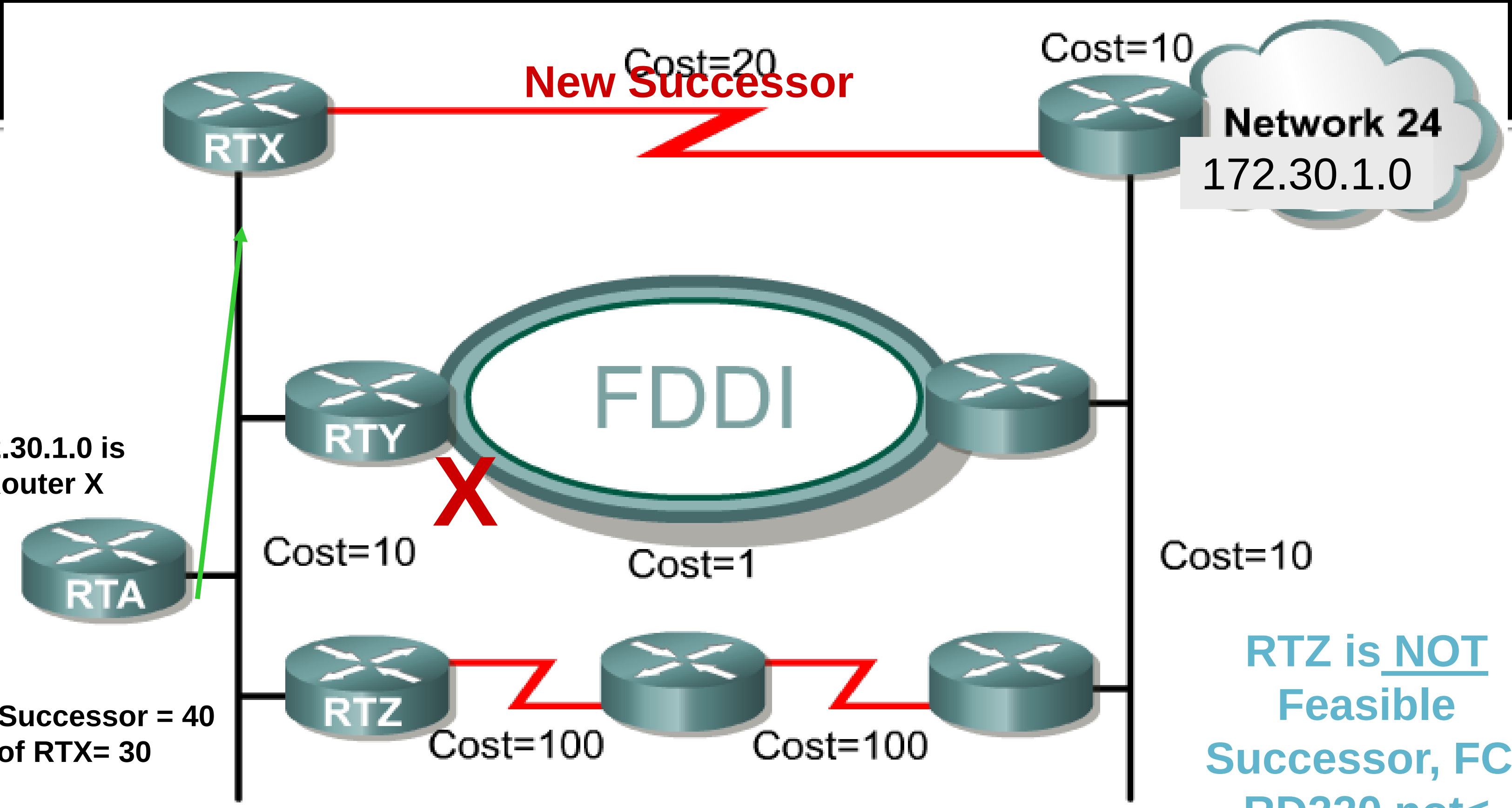
Feasible Successor exists:

- If current successor route fails, feasible successor becomes the current successor, i.e. the current route.
- Routing of packets continue with little delay.

No Feasible Successor exists:

- This may be because the Reported Distance is greater than the Feasible Distance.
- Before this route can be installed, it must be placed in the *active state* and recomputed (utilizing query and reply packets).
- Routing of packets continue but with more of a delay.

Successors and Feasible Successors (Cont.)



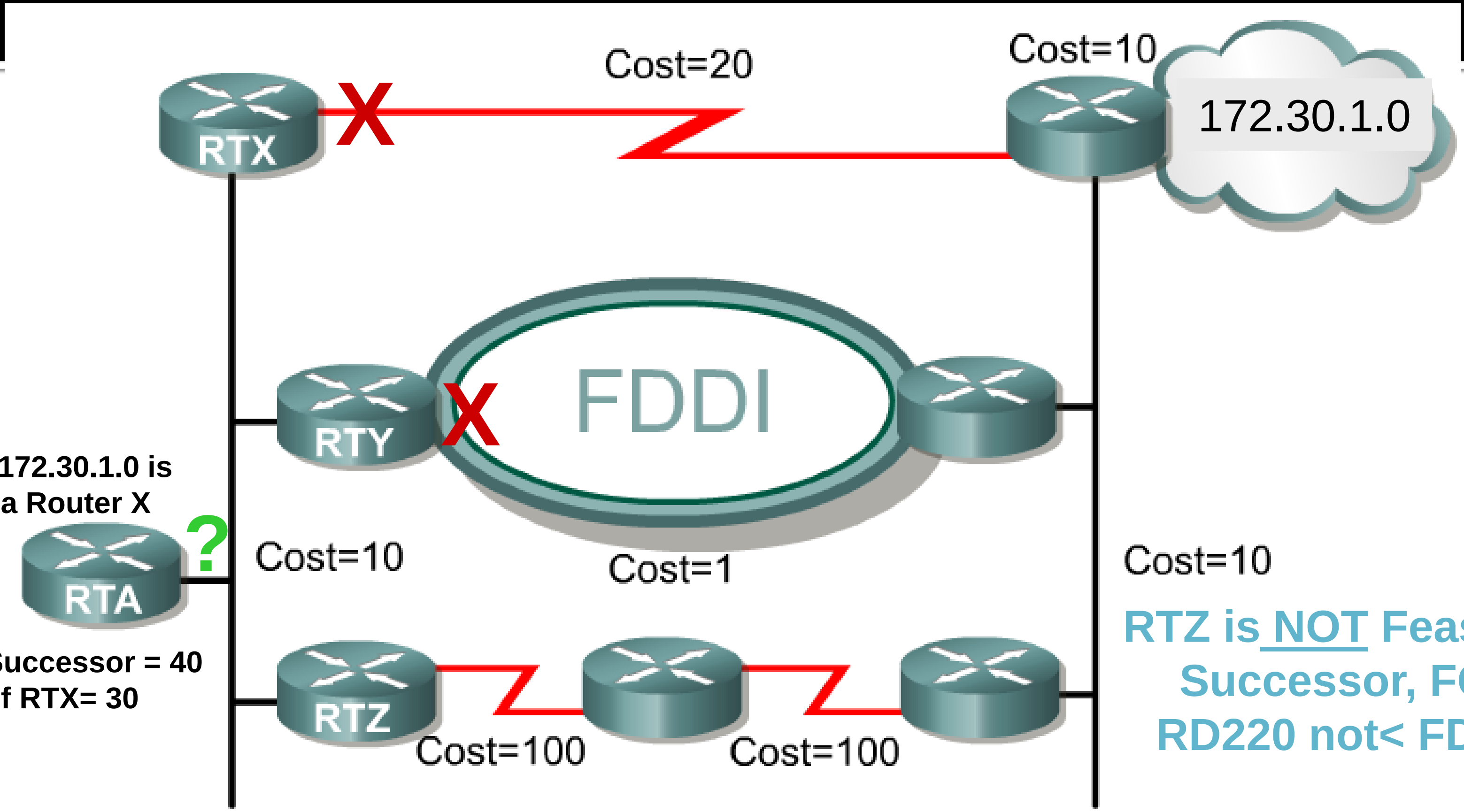
FD to 172.30.1.0 is 40 via Router X

Current Successor = 40
RD of RTX= 30

RTZ is NOT Feasible Successor, FC: RD220 not< FD31

- Since RTX is the feasible successor, and becomes the successor.
- RTX is immediately installed from the topology table into the routing table (no recomputation of DUAL).
- RTA’s new FD via RTX is 40.
- RTZ is not a feasible successor, because it’s RD (220) is still greater than the new FD (40) for 172.30.1.0/24.

Successors and Feasible Successors (Cont.)



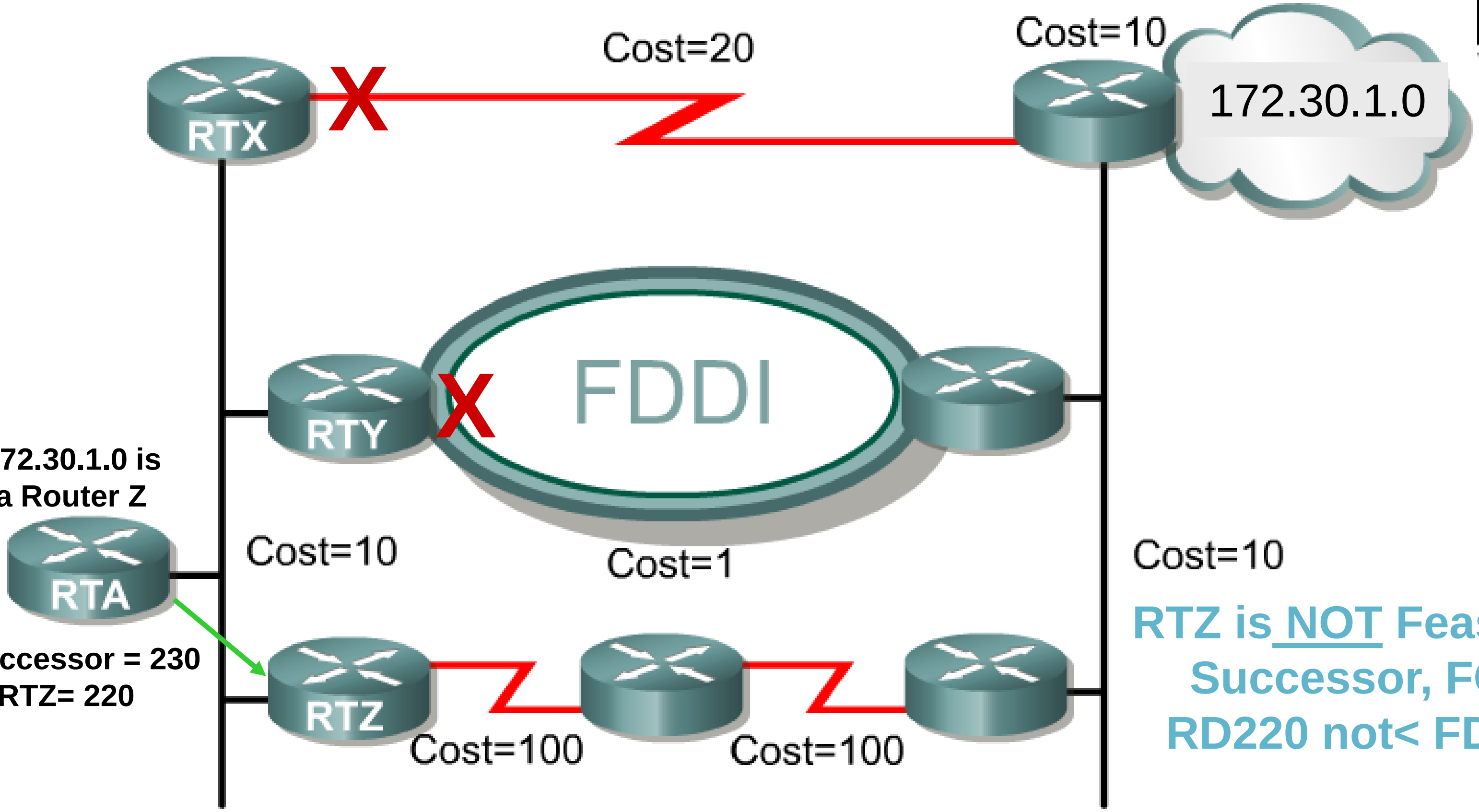
FD to 172.30.1.0 is
40 via Router X

Current Successor = 40
RD of RTX= 30

RTZ is <u>NOT</u> Feasible Successor, FC: RD220 not< FD40

- RTZ is not a feasible successor.
- It’s RD (220) is greater than the previous FD (40) for 172.30.1.0/24.
- Before this route can be installed, the route to net 24 must be placed in the *active state* and recomputed.

Successors and Feasible Successors (Cont.)



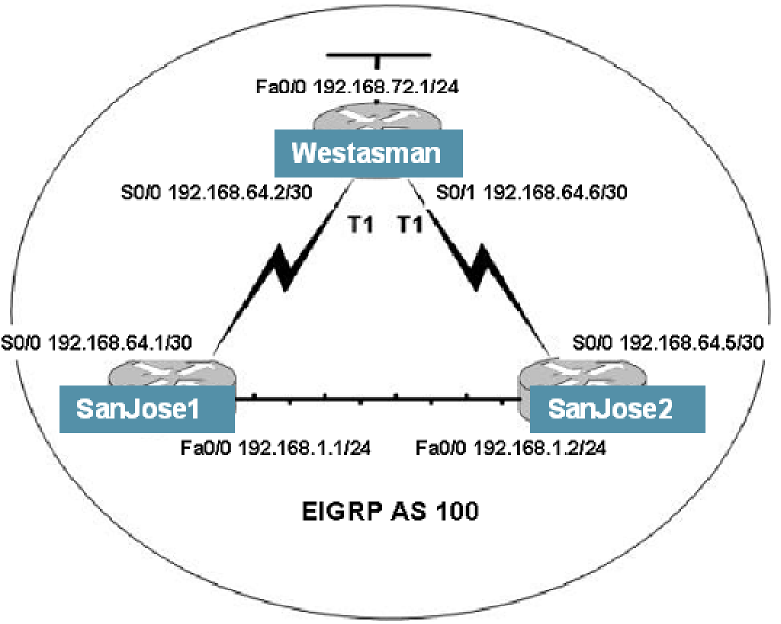
FD to 172.30.1.0 is 230 via Router Z

Current Successor = 230
RD of RTZ= 220

RTZ is NOT Feasible Successor, FC: RD220 not< FD40

- After a series of EIGRP Queries and Replies, and a recomputation of DUAL, RTZ becomes the successor.
- There is nothing better to prohibit it from being the successor.

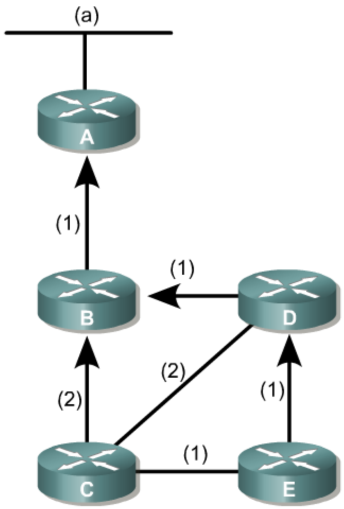
Understanding the Topology Table



```
SanJose2#show ip eigrp top all
P 192.168.72.0/24, 1 successors, FD is 2172416, serno 93
         via 192.168.64.6 (2172416/28160), Serial0
         via 192.168.1.1 (2174976/2172416), FastEthernet0
P 192.168.64.0/30, 1 successors, FD is 2172416, serno 91
         via 192.168.1.1 (2172416/2169856), FastEthernet0
         via 192.168.64.6 (2681856/2169856), Serial0
P 192.168.64.4/30, 1 successors, FD is 2169856, serno 72
         via Connected, Serial0
P 192.168.1.0/24, 1 successors, FD is 28160, serno 1
         via Connected, FastEthernet0
```

Utilizing Query and Reply Packets (Example):

1

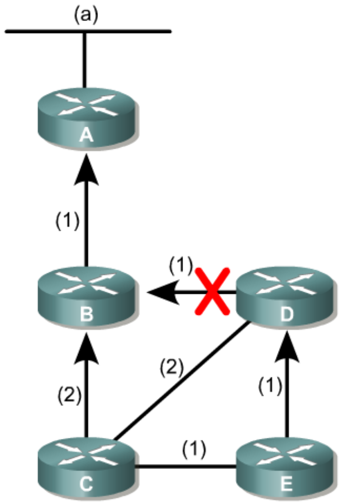


C EIGRP	FD	AD	Topology
(a)	3		(fd)
via B	3	1	(Successor)
via D	4	2	(fs)
via E	4	3	

D EIGRP	FD	AD	Topology
(a)	2		(fd)
via B	2	1	(Successor)
via C	5	3	

E EIGRP	FD	AD	Topology
(a)	3		(fd)
via D	3	2	(Successor)
via C	4	3	

2

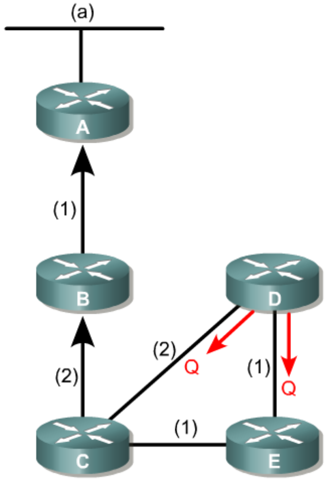


C EIGRP	FD	AD	Topology
(a)	3		(fd)
via B	3	1	(Successor)
via D	4	2	(fs)
via E	4	3	

D EIGRP	FD	AD	Topology
(a)	2		(fd)
~~via B~~	~~2~~	~~1~~	~~(Successor)~~
via C	5	3	

E EIGRP	FD	AD	Topology
(a)	3		(fd)
via D	3	2	(Successor)
via C	4	3	

3

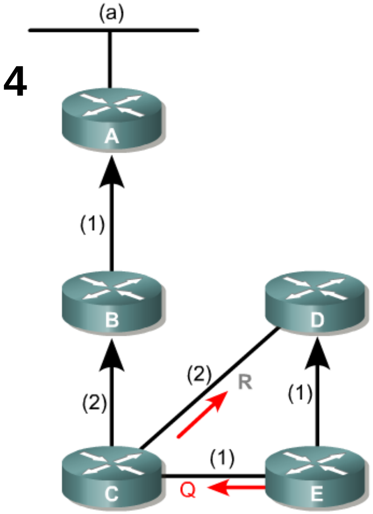


C EIGRP	FD	AD	Topology
(a)	3		(fd)
via B	3	1	(Successor)
via D			
via E	4	3	

D EIGRP	FD	AD	Topology
(a) **ACTIVE**	-1		(fd)
via B			(q)
via C	5	3	(q)

E EIGRP	FD	AD	Topology
(a)	3		(fd)
~~via D~~	~~3~~	~~2~~	~~(Successor)~~
via C	4	3	

4



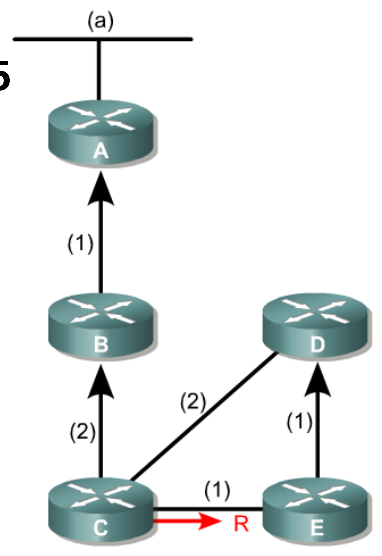
C EIGRP	FD	AD	Topology
(a)	3		(fd)
via B	3	1	(Successor)
via D	4	2	(fs)
via E	4	3	

D EIGRP	FD	AD	Topology
(a) **ACTIVE**	-1		(fd)
via B			(q)
via C	5	3	(q)

D EIGRP	FD	AD	Topology
(a) **ACTIVE**	-1		(fd)
via D			
via C	4	3	(q)

Example (Cont.):

5

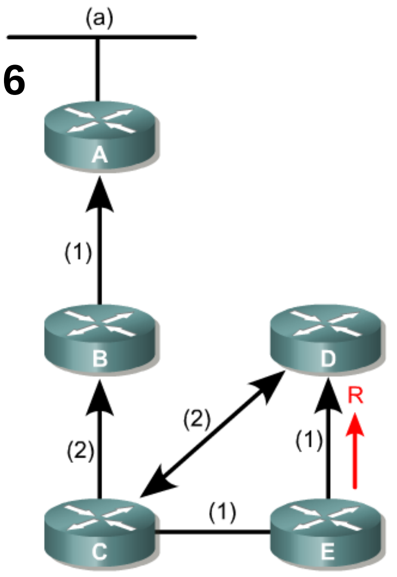


C	EIGRP	FD	AD	Topology
(a)		3		(fd)
	via B	3	1	(Successor)
	via D			
	via E			

D	EIGRP	FD	AD	Topology
(a)	**ACTIVE**	-1		(fd)
	via E			(q)
	via C	5	3	

D	EIGRP	FD	AD	Topology
(a)		4		(fd)
	via C	4	3	(Successor)
	via D			

6

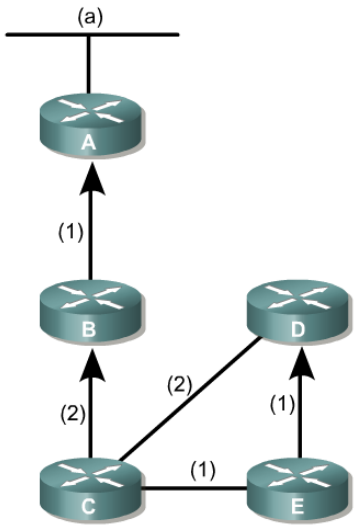


C	EIGRP	FD	AD	Topology
(a)		3		(fd)
	via B	3	1	(Successor)
	via D			
	via E			

D	EIGRP	FD	AD	Topology
(a)		5		(fd)
	via E	5	3	(Successor)
	via C	5	4	

D	EIGRP	FD	AD	Topology
(a)		4		(fd)
	via C	4	3	(Successor)
	via D			

7



C	EIGRP	FD	AD	Topology
(a)		3		(fd)
	via B	3	1	(Successor)
	via D			
	via E			

D	EIGRP	FD	AD	Topology
(a)		5		(fd)
	via C	5	3	(Successor)
	via E	5	4	(Successor)

E	EIGRP	FD	AD	Topology
(a)		4		(fd)
	via C	4	3	(Successor)
	via D			

Stuck in Active (SIA) (Cont.)

- Typically, SIAs results when a router cannot answer a query because:
  - the router is too busy to answer the query (generally high cpu utilization)
  - the router cannot allocate the memory to process the query or build the reply packet
  - the circuit between the two routers is not good (packet loss)
  - unidirectional links (a link on which traffic can only flow in one direction due to a failure)
- When this happens, the router that issued the query gives up and resets its neighbor relationship with the router that didn’t answer.

Summary

- **Background & History**
  - EIGRP is a derivative of IGRP
    - EIGRP is a Cisco proprietary distance vector routing protocol released in 1994
- **EIGRP terms and characteristics**
  - EIGPR uses RTP to transmit & receive EIGRP packets
  - EIGRP has 5 packet type:
    - Hello packets
    - Update packets
    - Acknowledgement packets
    - Query packets
    - Reply packets
  - Supports VLSM & CIDR

Summary

- **EIGRP terms and characteristics**
  - EIGRP uses a hello protocol
    - Purpose of hello protocol is to discover & establish adjacencies
  - EIGRP routing updates
    - A periodic
    - Partial and bounded
    - Fast convergence

Summary

- **EIGRP metrics include**
  - Bandwidth (default)
  - Delay (default)
  - Reliability
  - Load

Summary

- **DUAL**
  - Purpose of DUAL
    - To prevent routing loops
  - Successor
    - Primary route to a destination
  - Feasible successor
    - Backup route to a destination
  - Feasible distance
    - Lowest calculated metric to a destination
  - Reported distance
    - The distance towards a destination as advertised by an upstream neighbor

Summary

- **Choosing the best route**
  - After router has received all updates from directly connected neighbors, it can calculate its DUAL
    - 1st metric is calculated for each route
    - 2nd route with lowest metric is designated successor & is placed in routing table
    - 3rd feasible successor is found
      - Criteria for feasible successor: it must have lower reported distance to the destination than the installed route’s feasible distance
      - Feasible routes are maintained in topology table

Questions?

TCP Congestion Control

- TCP has a mechanism for congestion control. The mechanism is implemented at the sender

- The window size at the sender is set as follows:

Send Window = MIN (flow control window, congestion window)

where

- flow control window is advertised by the receiver
- congestion window is adjusted based on feedback from the network

What Are TCP Variations?

- Implementations of TCP that use different algorithms to achieve end-to-end congestion control.
  - Tahoe
  - Reno
  - NewReno

TCP Variation: *TCP Tahoe*

- 1st improvement was TCP Tahoe (1988)
  - – Adjusts sending window as congestion increases or decreases (AIMD congestion avoidance & slow-start)
  - – Improved retransmission policy (Fast Retransmit)

TCP Tahoe Window Control

- *TCP sender* maintains two new variables:
  - cwnd – congestion window

    cwnd is inferred from the level of congestion in the network.
  - ssthresh – slow-start threshold

    ssthresh can be thought of as an estimate of the level below which congestion is not expected.

Slow Start Phase

(cwnd < ssthresh)

- Initially:
  - cwnd = 1*MSS (Maximum Segment Size)
  - ssthresh is very large.
- If no loss:
  - cwnd += 1*MSS (after each new ACK)
  - *(This gives exponential growth of cwnd)*
- If loss (timeout):
  - ssthresh = max(flight size/2, 2*MSS)
  - cwnd = 1*MSS

Congestion Avoidance Phase (cwnd > ssthresh)

- If no loss:
  - increase cwnd *at most* 1*MSS per RTT (additive increase)
  - cwnd += (MSS*MSS / cwnd) on every ACK (*approximation* to increasing cwnd by 1*MSS per RTT)
- If loss:
  - ssthresh = max (flight size/2, 2*MSS) (multiplicative decrease)
  - cwnd = 1*MSS.

Slow Start & Congestion Avoidance

- Initally:

  - cwnd = 1*MSS

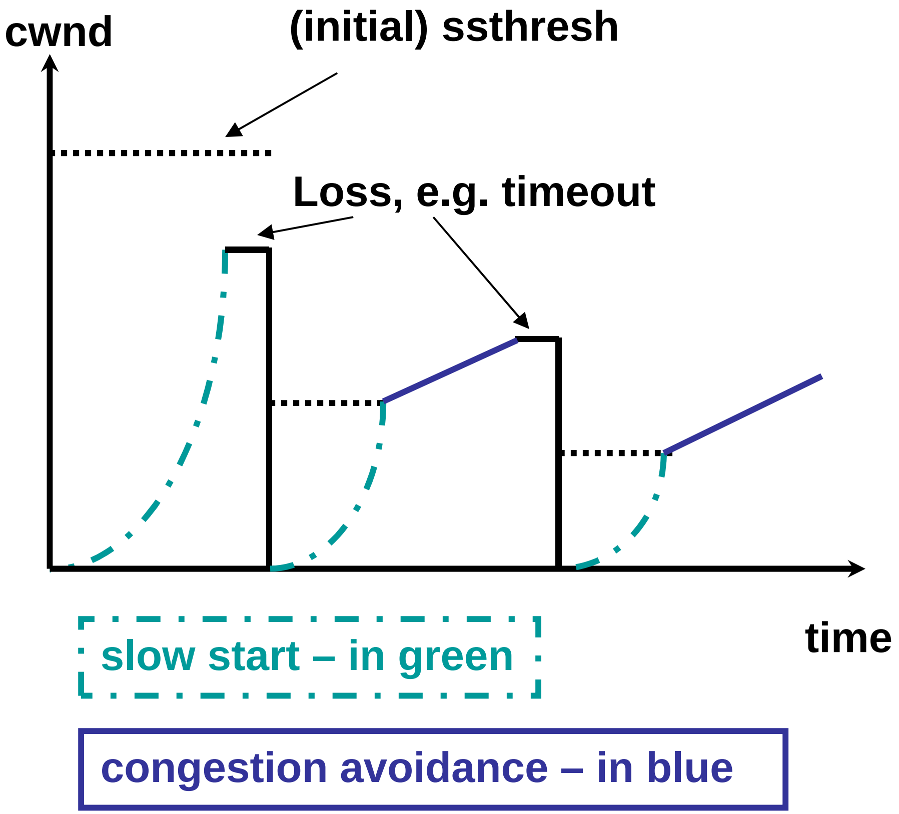
  - ssthresh = very high (65535)

- If a new ACK comes:
  - if cwnd < ssthresh → update cwnd according to slow start
  - if cwnd > ssthresh → update cwnd according to congestion avoidance
  - If cwnd = ssthresh → either

- If timeout (i.e. loss) :

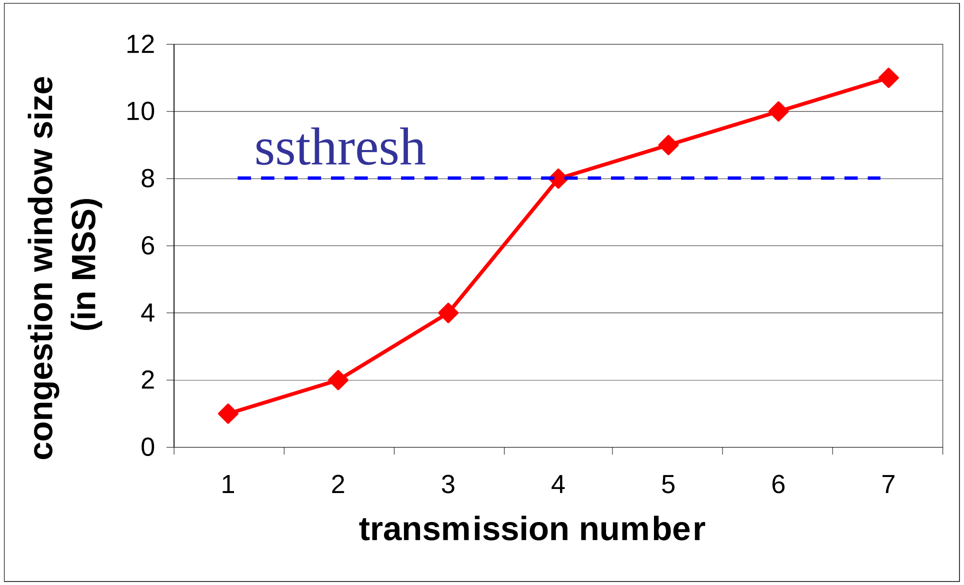
  - ssthresh = flight size/2;

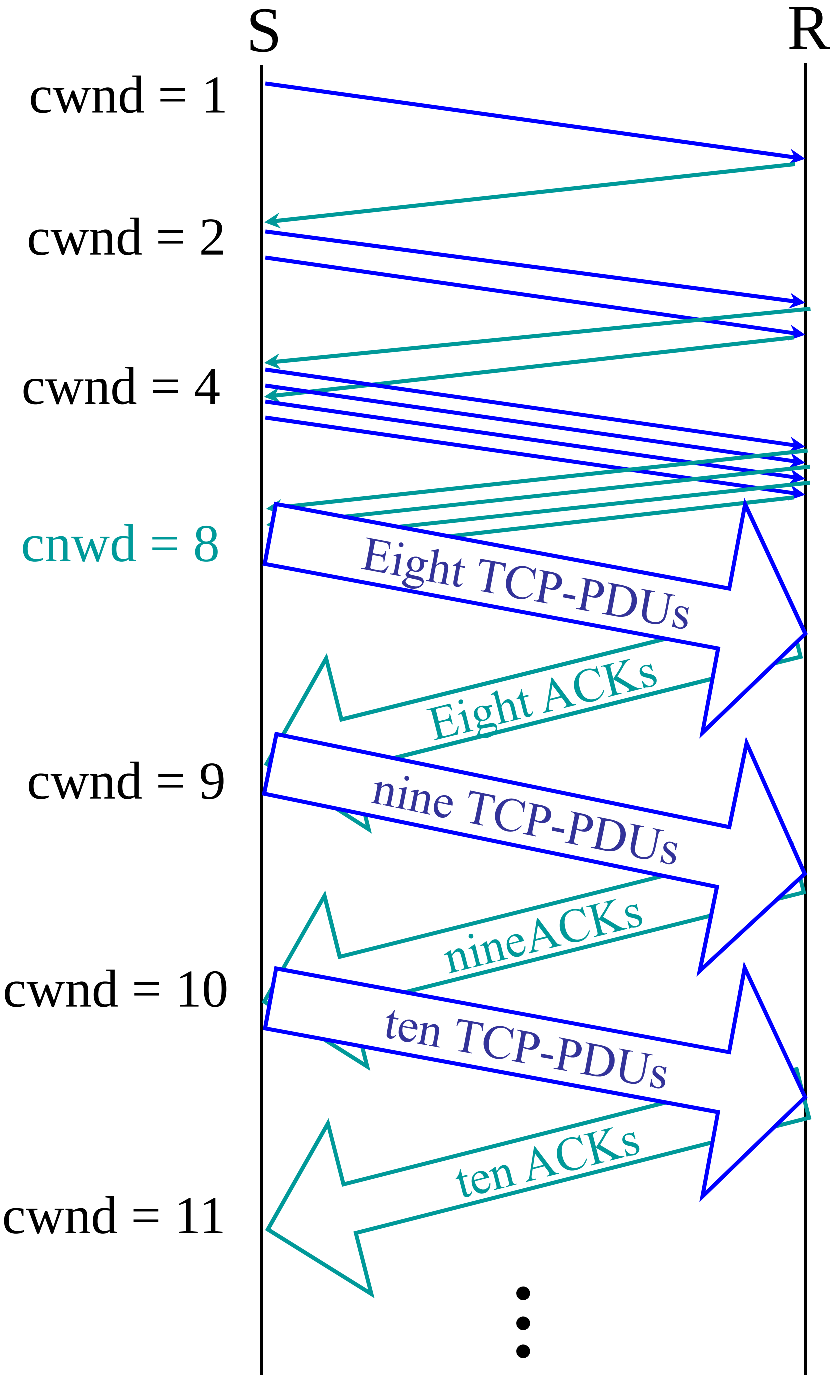
  - cwnd = 1*MSS



Example: Slow Start/Congestion Avoidance

assume ssthresh = 8*MSS





S

R

cwnd = 1

cwnd = 2

cwnd = 4

cnwd = 8

Eight TCP-PDUs

Eight ACKs

cwnd = 9

nine TCP-PDUs

nineACKs

cwnd = 10

ten TCP-PDUs

ten ACKs

cwnd = 11

⋮

TCP Tahoe’s Retransmission Policy

- When a segment is lost, original TCP waits for an ACK that’s not coming and eventually times-out.

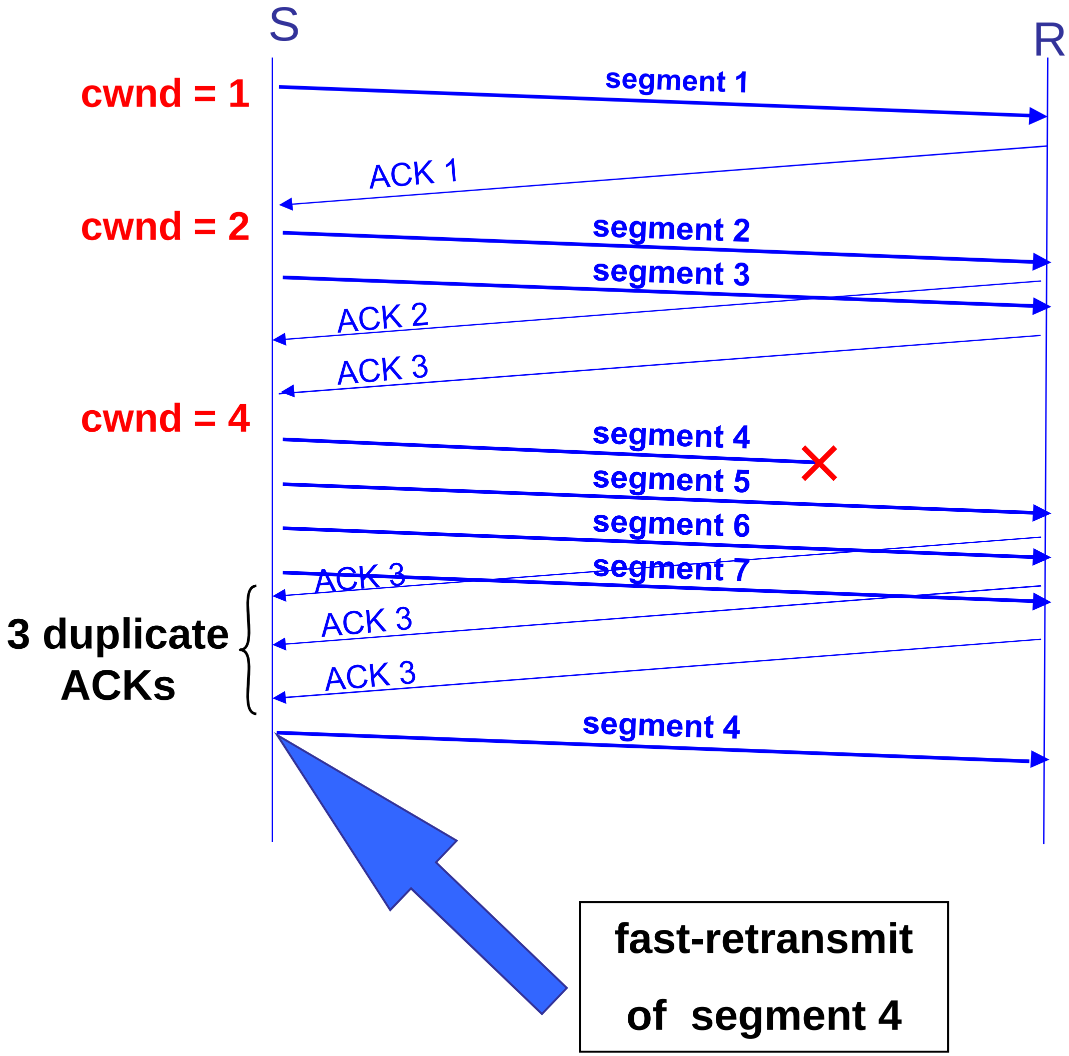
- Often, many, if not all, of the segments sent after the lost segment arrive at the receiver.

- For each segment received, the receiver sends a duplicate ACK, notifying the sender that the receiver is waiting for the missing segment.

- TCP Tahoe interprets duplicate ACK’s as an indication that a segment was lost.

TCP Tahoe’s Fast Retransmit

1. Sender receives 3 dupACKS.
2. Sender infers that the segment is lost.
3. Sender re-sends the segment immediately!
4. Sender returns to slow-start.



Fast Retransmit (Cont.)

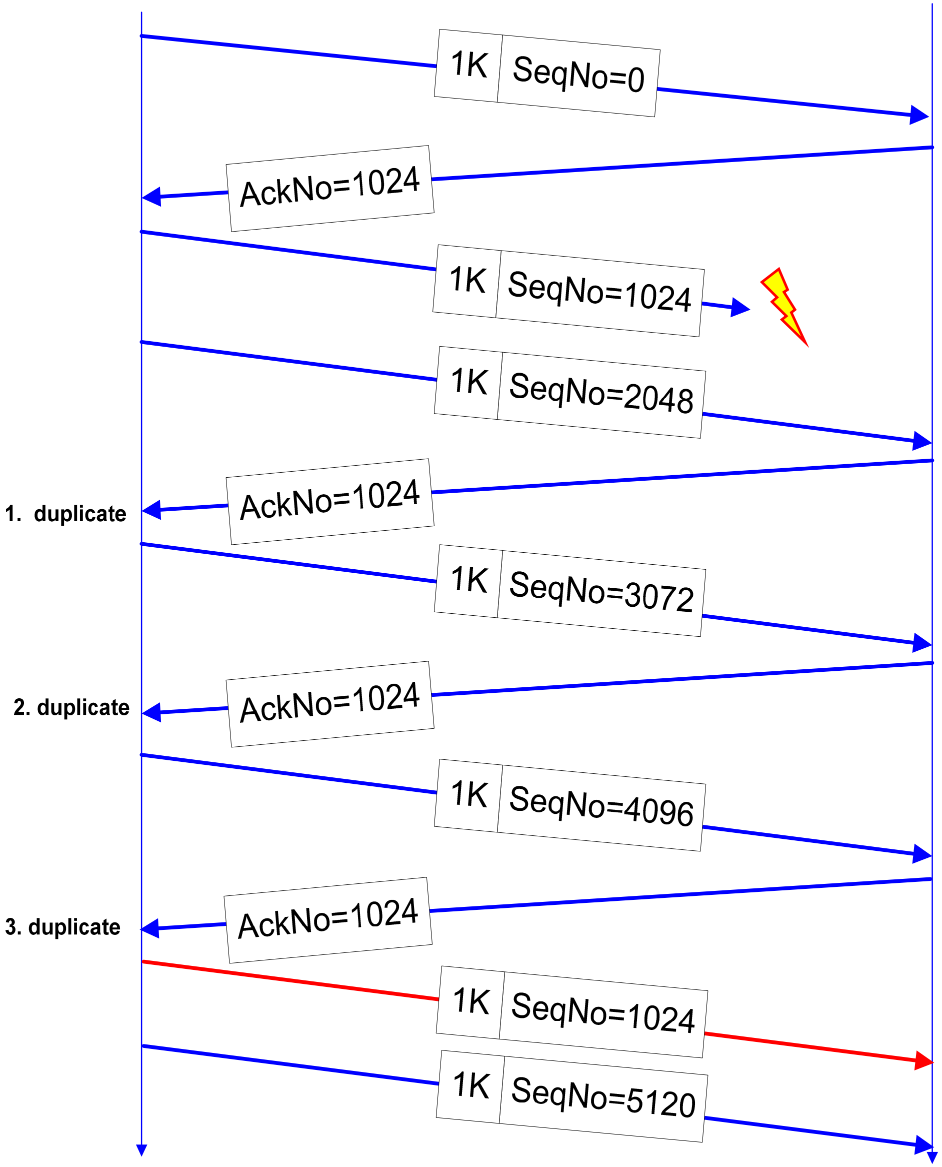
- If three or more duplicate ACKs are received in a row, the TCP sender believes that a segment has been lost.

- Then TCP performs a retransmission of what seems to be the missing segment, without waiting for a timeout to happen.

- Enter slow start:

  ssthresh = cwnd/2

  cwnd = 1



TCP Variation: *TCP Reno*

- 2nd Improvement was TCP Reno (1990)
  - From Tahoe:
    - AIMD congestion avoidance with slow-start
    - Fast retransmit
  - New to Reno:
    - Fast recovery

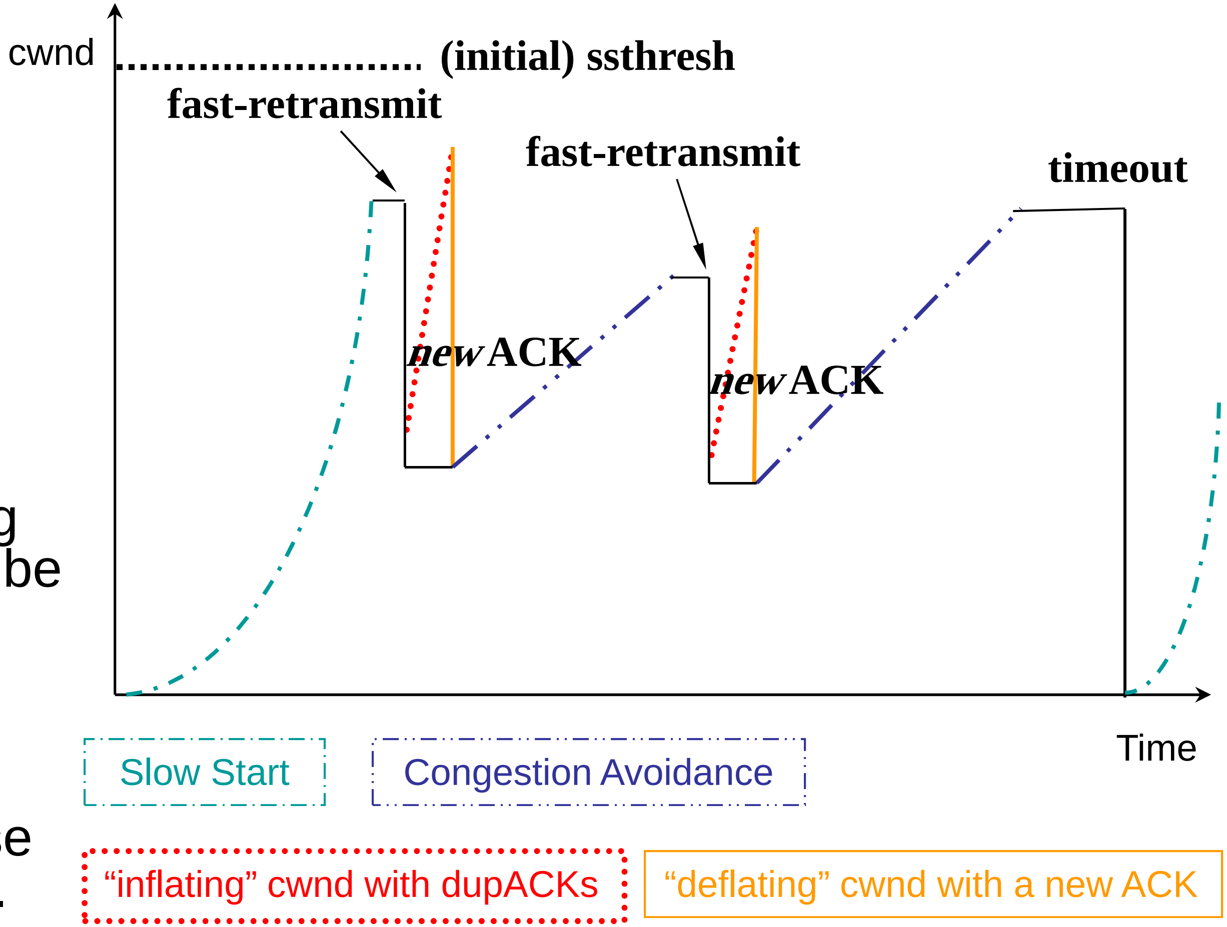
Fast Recovery

Concept:

- After fast retransmit, reduce cwnd by half, and continue sending segments at this reduced level.

Observations:

- Receiver is still getting T-PDUs. There can't be overwhelming congestion.
- How does sender transmit T-PDUs on a dupACK? Need to use a "trick" - inflate cwnd.



Fast Recovery (Cont.)

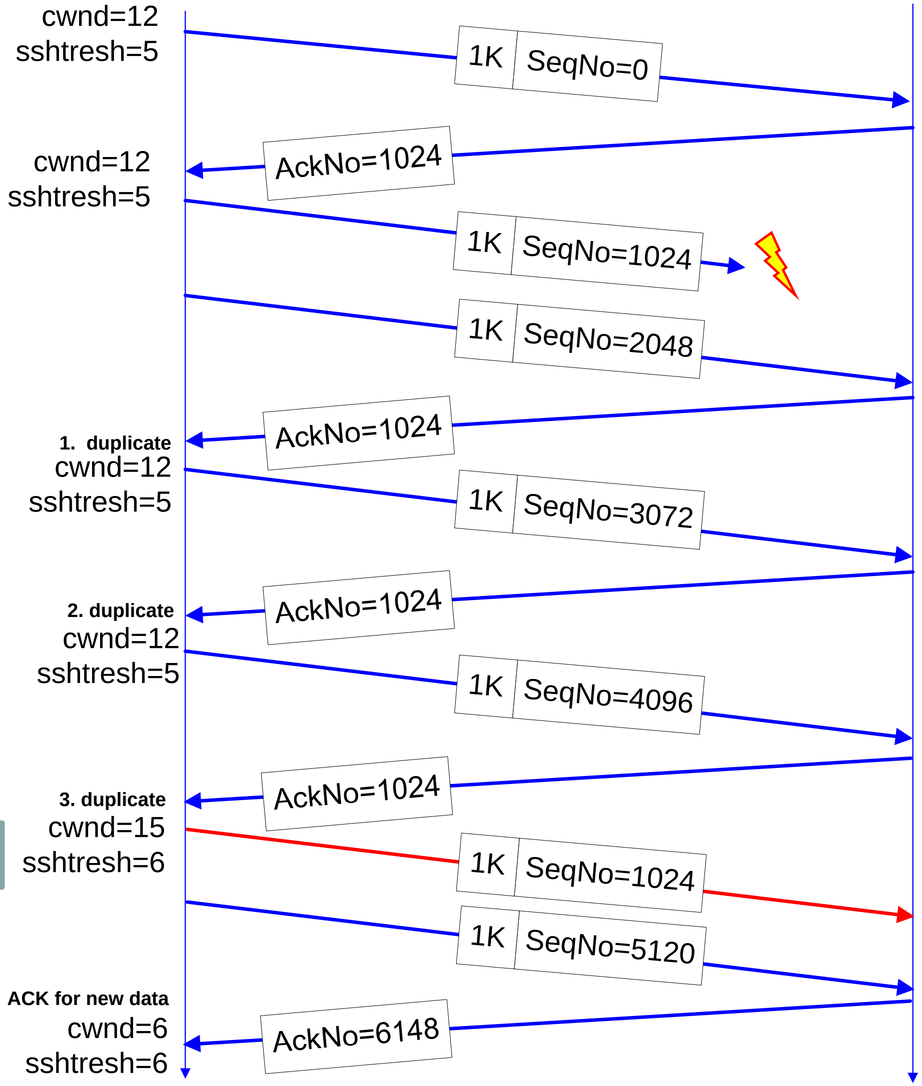
- Fast recovery avoids slow start after a fast retransmit
- **Intuition:** Duplicate ACKs indicate that data is getting through
- After three duplicate ACKs set:
  - Retransmit packet that is presumed lost
  - ssthresh = cwnd/2
  - cwnd = (cwnd/2)+3
  - (note the order of operations)
  - Increment cwnd by one for each additional duplicate ACK

cwnd=9 (not 15)

- When ACK arrives that acknowledges "new data" (here: AckNo=6148), set:

  cwnd=ssthresh

  enter congestion avoidance



TCP Reno (Revisited)

- Duplicate ACKs:
    - Fast retransmit
    - Fast recovery

  → Fast Recovery avoids slow start

- Timeout:
    - Retransmit
    - Slow Start

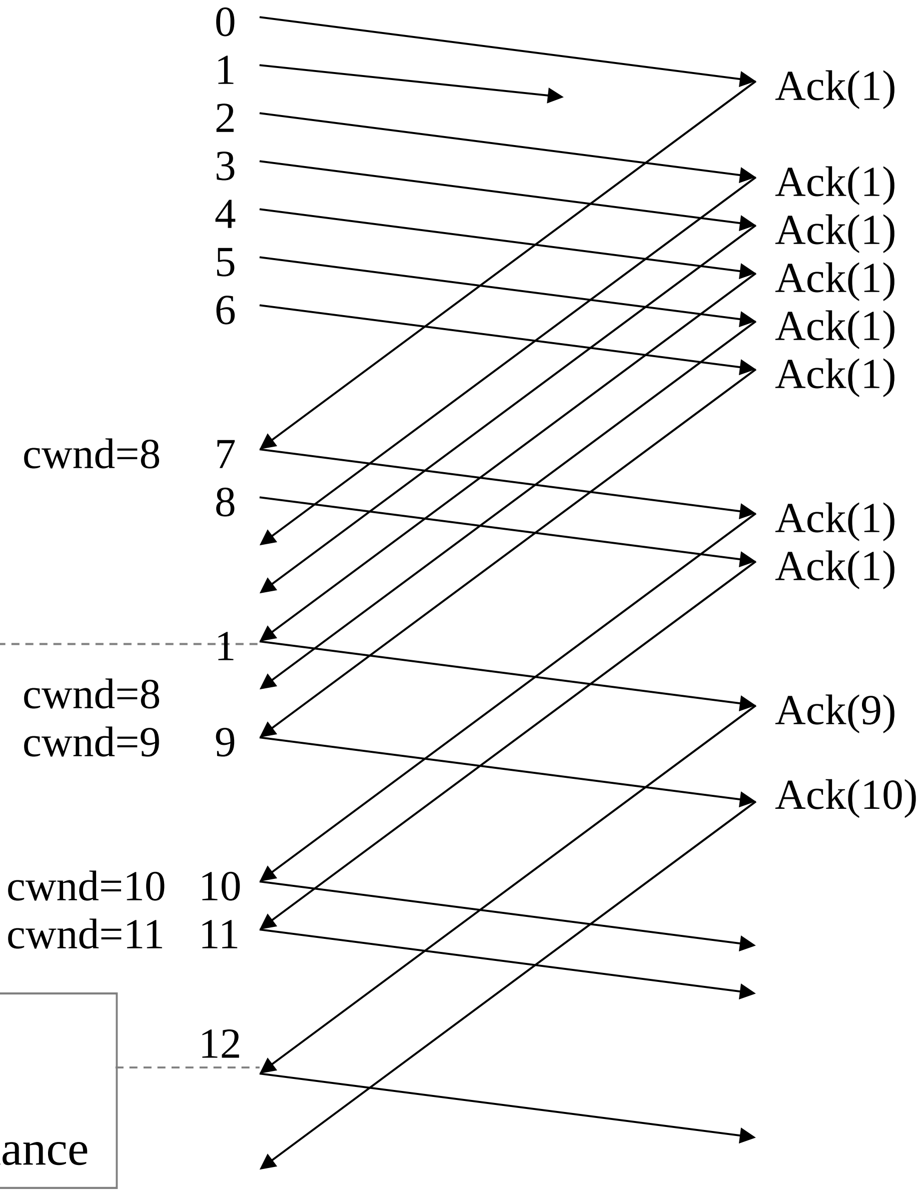
- TCP Reno improves upon TCP Tahoe when a single packet is dropped in a round-trip time.

Fast Retransmit / Fast Recovery in TCP Reno

- A sender uses fast retransmit / fast recovery algorithm to improve throughput of TCP
  - "Fast" – because it doesn't wait for *time out* when not getting an ACK for a segment
- **Fast Retransmit** – after 3 "**duplicative ACKs**", the sender assumes that the segment was lost, retransmits the segment and moves to Fast Recovery phase
- **Fast Recovery** – the sender decreases Congestion Window (cwnd) twice of its original size, adds 3 (3 packets have left the network and buffered by the receiver) and continue to send new segments (if allowed by the cwnd value) until receiving new different ACK, which should acknowledge receiving all segments sent till moving to Fast Recovery phase (assuming that no more segments were lost).
  - For each additional duplicated ACK received, increment cwnd by 1

Example

Initial state
cwnd=7
Slow start



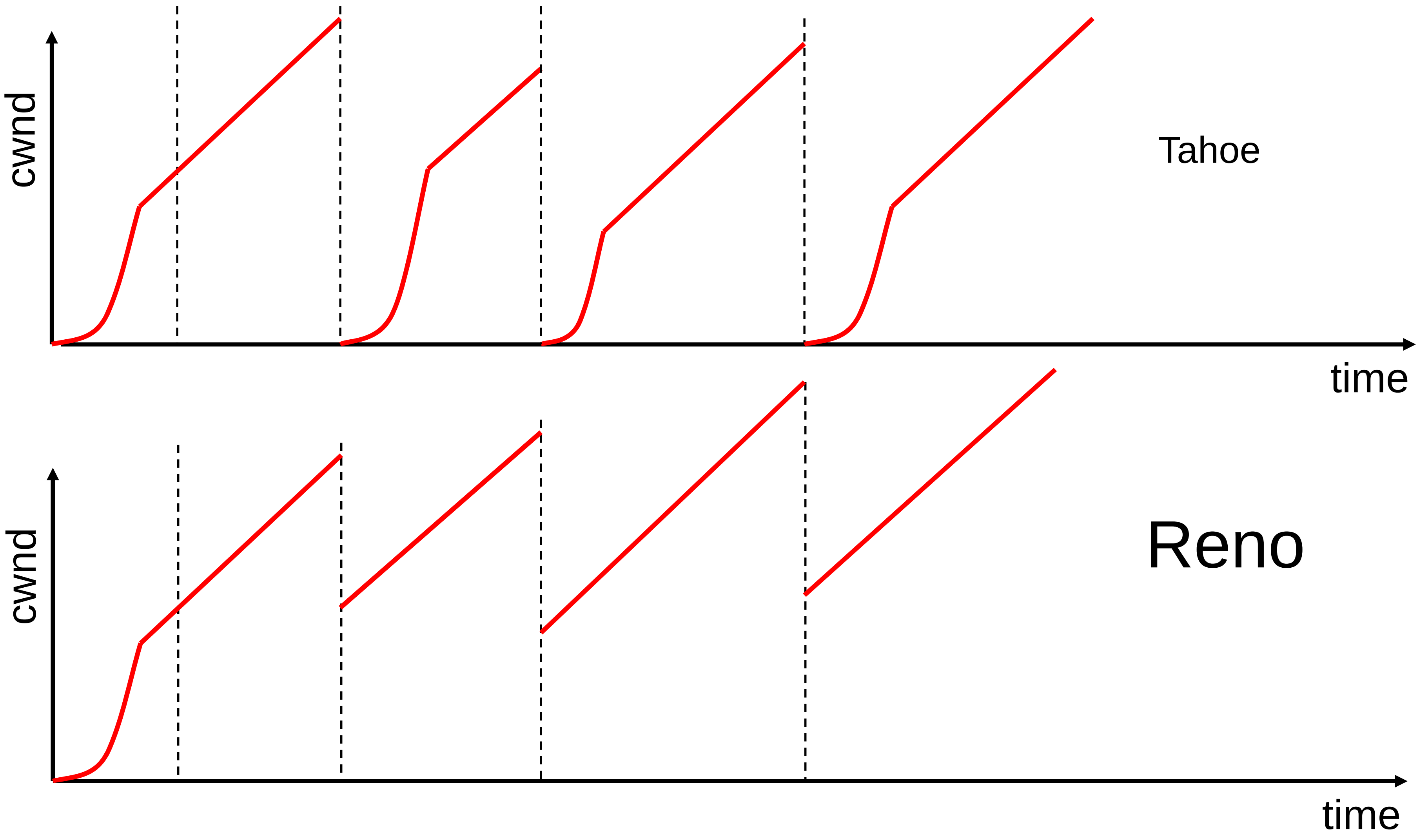
0
1
2
3
4
5
6
cwnd=8 7
8
1
cwnd=8
cwnd=9 9
cwnd=10 10
cwnd=11 11
12

Ack(1)
Ack(1)
Ack(1)
Ack(1)
Ack(1)
Ack(1)
Ack(1)
Ack(1)
Ack(9)
Ack(10)

Fast Retransmit
cwnd=8/2+3=7
ssthresh=8/2=4
=> Fast Recovery

Exit Fast Recovery
cwnd=ssthresh=4
=> Congestion Avoidance

TCP Tahoe and TCP Reno
(for single segment losses)



Limitation of TCP Reno algorithm

- If **cwnd** size is too small (smaller than 4 packets) then it’s not possible to get 3 duplicate acks and run the algorithm
- The algorithm can not manage a loss of multiple packets from a single window of data
  - It will cause a use of retransmission time out
- The algorithm doesn’t manage a loss of packets during the Fast Recovery stage
  - Not a loss of the retransmitted packet
  - There is no recursive run of the Fast Retransmit

Example

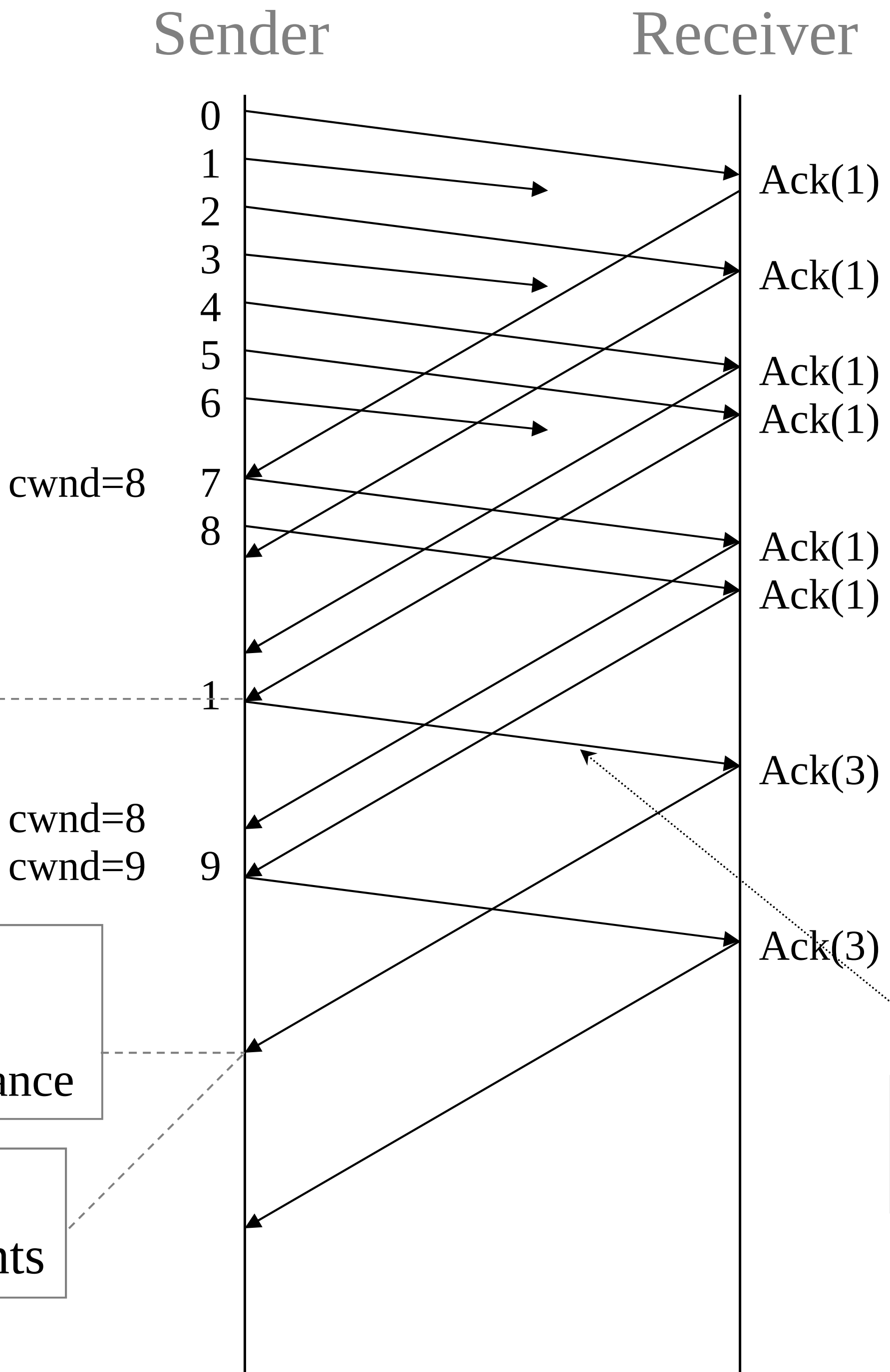
Initial State
cwnd=7
Slow Start

Flight Size =No. of Unacknowledged segments

Fast Retransmit
cwnd=8/2+3=7
ssthresh=8/2=4
=> Fast Recovery

Exit Fast Recovery
cwnd=ssthresh=4
=> Congestion Avoidance

Flight Size > cwnd
=> No new segments



The algorithm doesn’t know which segments were acknowledged

What was happen
if this packet was lost?

TCP New Reno

- The Idea: If the sender remembers the number of the last segment that was sent before entering the Fast Retransmit phase
  - Then it can deal with a situation when a “new” ACK (which is not **duplicate ACK**) does not cover the last remembered segment (**“partial ACK”**)
  - This is a situation when more packets were lost before entering the Fast Retransmit.
- After discovering such situation the sender will retransmit the new lost packet too and will stay at the Fast Recovery stage
- The sender will finish the Fast Recovery stage when it will get ACK that covers last segment sent before the Fast Retransmit

TCP New Reno – Retransmission Process (I)

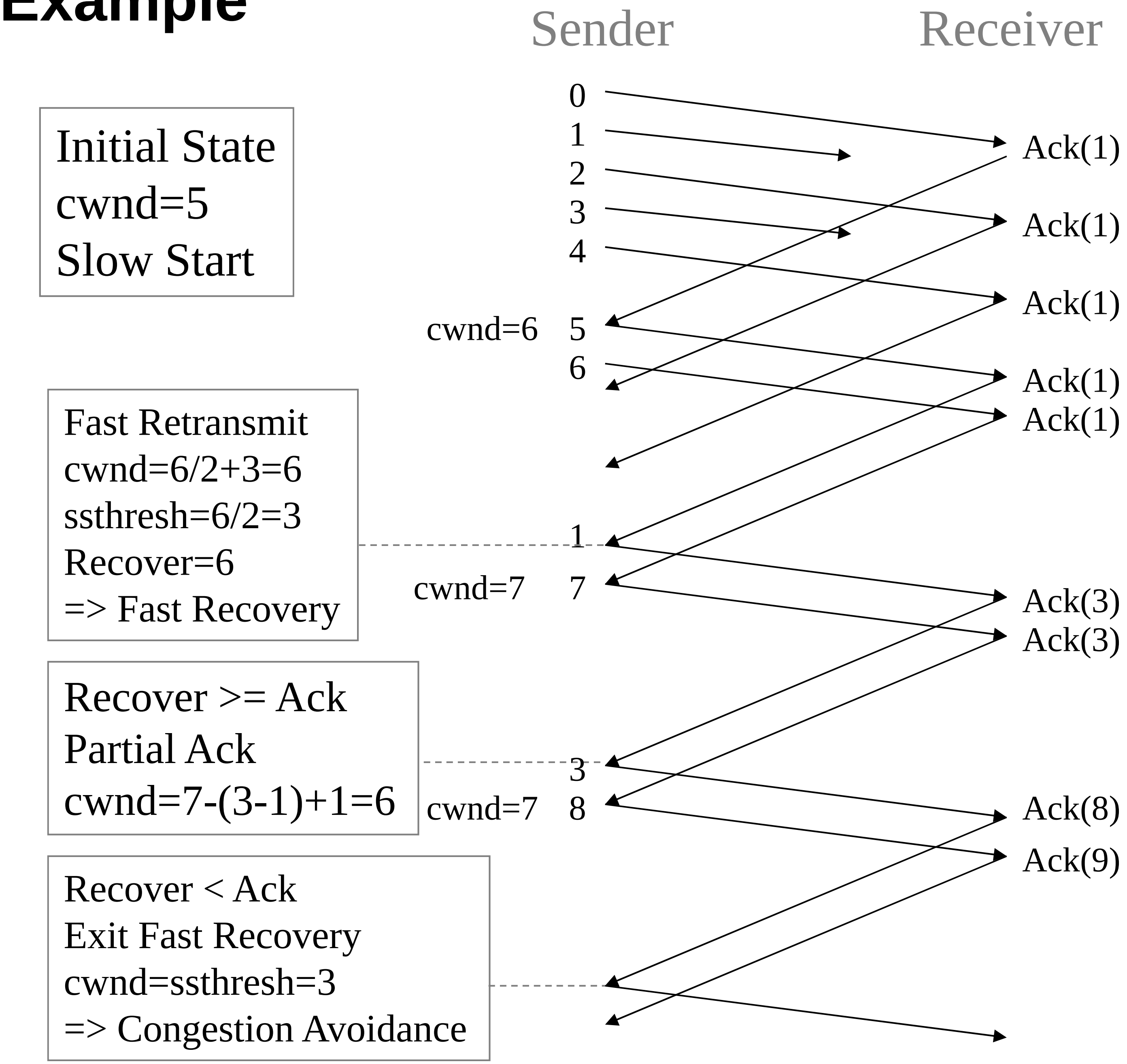
- Set ssthresh to max (FlightSize / 2, 2*MSS)
- Record to “Recovery” variable the highest sequence number transmitted
- Retransmit the lost segment and set cwnd to ssthresh + 3*MSS.
  - The congestion window is increased by the number of segments (three) that were sent and buffered by the receiver.
- For each additional duplicate ACK received, increment cwnd by MSS.
  - Thus, the congestion window reflects the additional segment that has left the network.
- Transmit a segment, if allowed by the new value of cwnd and the receiver's advertised window.

TCP New Reno – Retransmission Process (II)

- When a partial ACK is received
  - retransmit the first unacknowledged segment
  - deflate the congestion window by the amount of new acknowledged data, then add back one MSS
  - send a new segment if permitted by the new value of cwnd

- When an acknowledge of all of the data up to and including "recover“ arrives:
  - In our example: Set cwnd to ssthresh

Example



Initial State
cwnd=5
Slow Start

Fast Retransmit
cwnd=6/2+3=6
ssthresh=6/2=3
Recover=6
=> Fast Recovery

Recover >= Ack
Partial Ack
cwnd=7-(3-1)+1=6

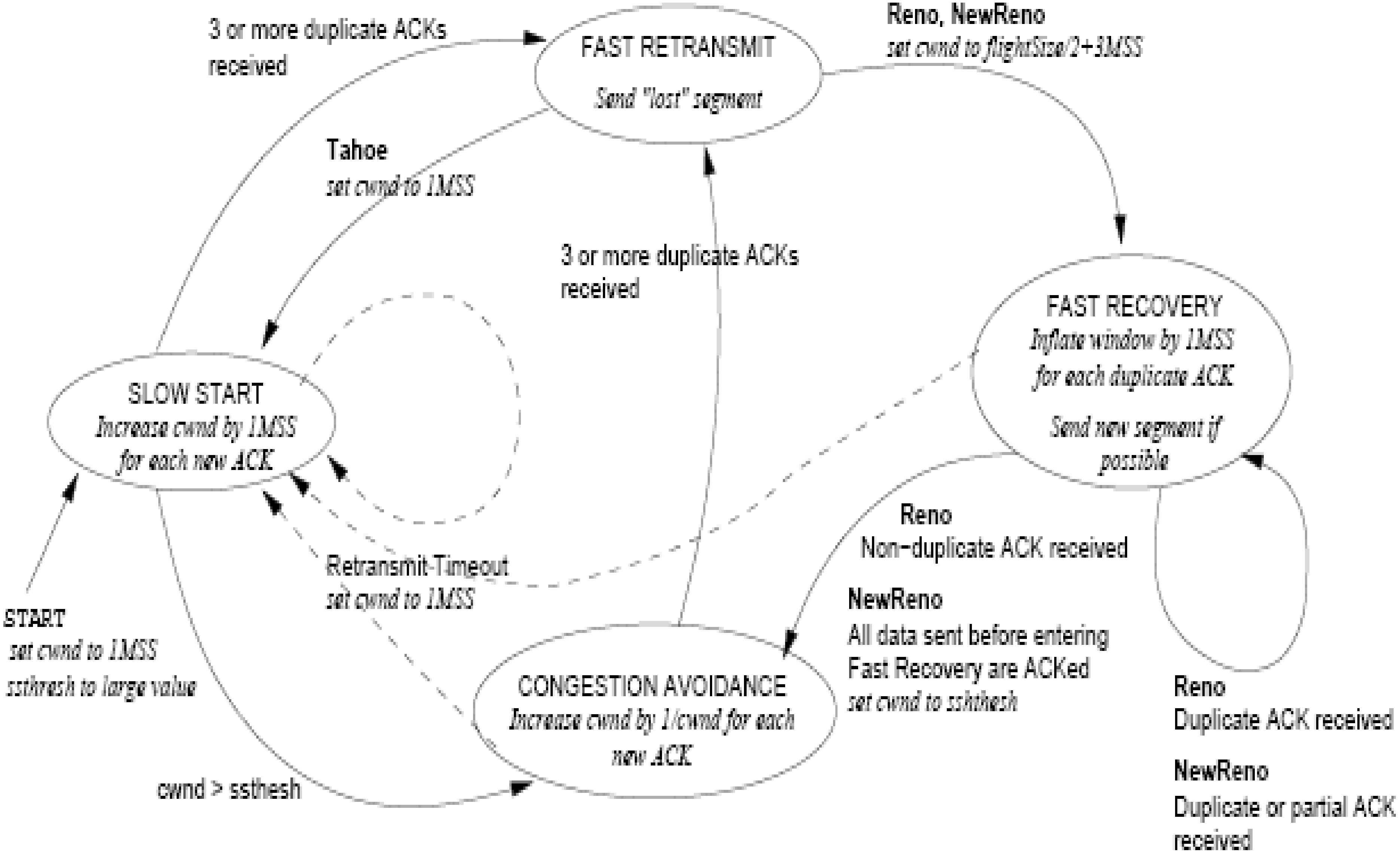
Recover < Ack
Exit Fast Recovery
cwnd=ssthresh=3
=> Congestion Avoidance

TCP New Reno (Summary)

- When multiple packets are dropped, Reno has problems
- Partial ACK:
  - Occurs when multiple packets are lost
  - A partial ACK acknowledges some, but not all packets that are outstanding at the start of a fast recovery, takes sender out of fast recovery

  → Sender has to wait until timeout occurs
- **New Reno:**
  - Partial ACK does not take sender out of fast recovery
  - Partial ACK causes retransmission of the segment following the acknowledged segment
- New Reno can deal with multiple lost segments without going to slow start

State Transitions for Tahoe, Reno & New Reno


FAST RETRANSMIT
Send "lost" segment
3 or more duplicate ACKs received
Reno, NewReno
set cwnd to flightSize/2+3MSS
Tahoe
set cwnd to 1MSS
3 or more duplicate ACKs received
FAST RECOVERY
Inflate window by 1MSS for each duplicate ACK
Send new segment if possible
SLOW START
Increase cwnd by 1MSS for each new ACK
Reno
Non−duplicate ACK received
NewReno
All data sent before entering Fast Recovery are ACKed
set cwnd to ssthesh
Retransmit-Timeout
set cwnd to 1MSS
START
set cwnd to 1MSS
ssthresh to large value
CONGESTION AVOIDANCE
Increase cwnd by 1/cwnd for each new ACK
cwnd > ssthesh
Reno
Duplicate ACK received
NewReno
Duplicate or partial ACK received

Summary of TCP Behavior

TCP Variation	Response to 3 dupACK's	Response to Partial ACK of Fast Retransmission	Response to "full" ACK of Fast Retransmission
Tahoe	Do fast retransmit, enter slow start	++cwnd	++cwnd
Reno	Do fast retransmit, enter fast recovery	Exit fast recovery, deflate window, enter congestion avoidance	Exit fast recovery, deflate window, enter congestion avoidance
NewReno	Do fast retransmit, enter modified fast recovery	Fast retransmit and deflate window – remain in modified fast recovery	Exit modified fast recovery, deflate window, enter congestion avoidance

- When entering slow start, if connection is new,
  ssthresh = arbitrarily large value
  cwnd = 1.
  else,
  ssthresh = max(flight size/2, 2*MSS)
  cwnd = 1.
- In slow start ++cwnd on new ACK

- When entering either fast recovery or modified fast recovery,
  ssthresh = max(flight size/2, 2*MSS)
  cwnd = ssthresh.
- In congestion avoidance
  cwnd += 1*MSS per RTT

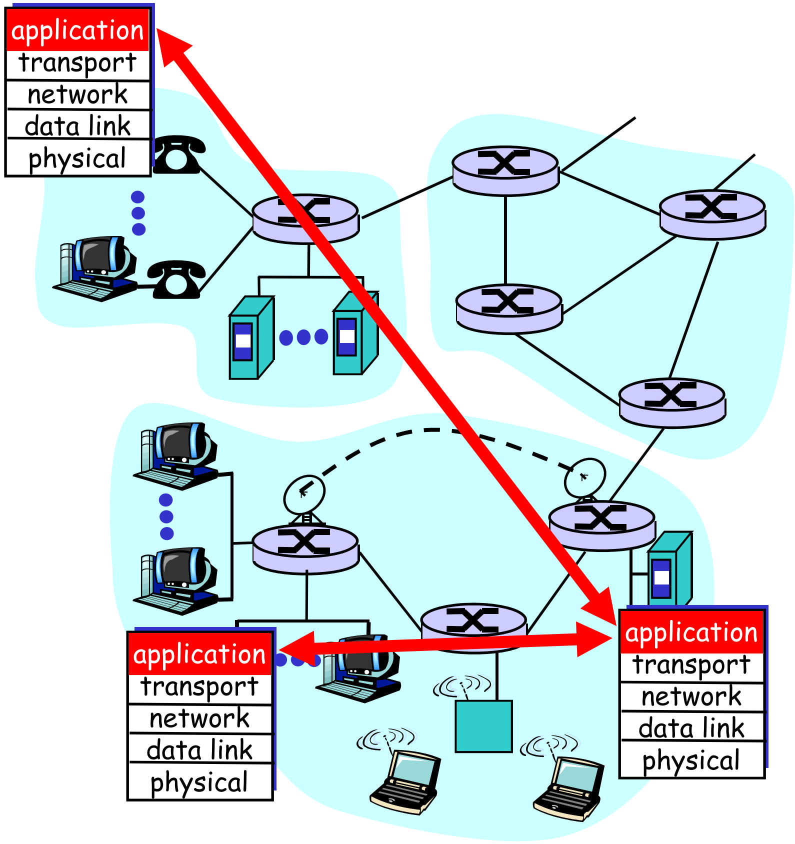
Computer Networks

Transport Layer Protocols

Application-layer Protocols

Application-layer protocols

- one “piece” of an app
- define messages exchanged by apps and actions taken
- use communication services provided by lower layer protocols (TCP, UDP)



Jargons

Process: program running within a host.

- within same host, two processes communicate using inter-process communication

- processes running in different hosts communicate with an application-layer protocol

Client-Server Paradigm

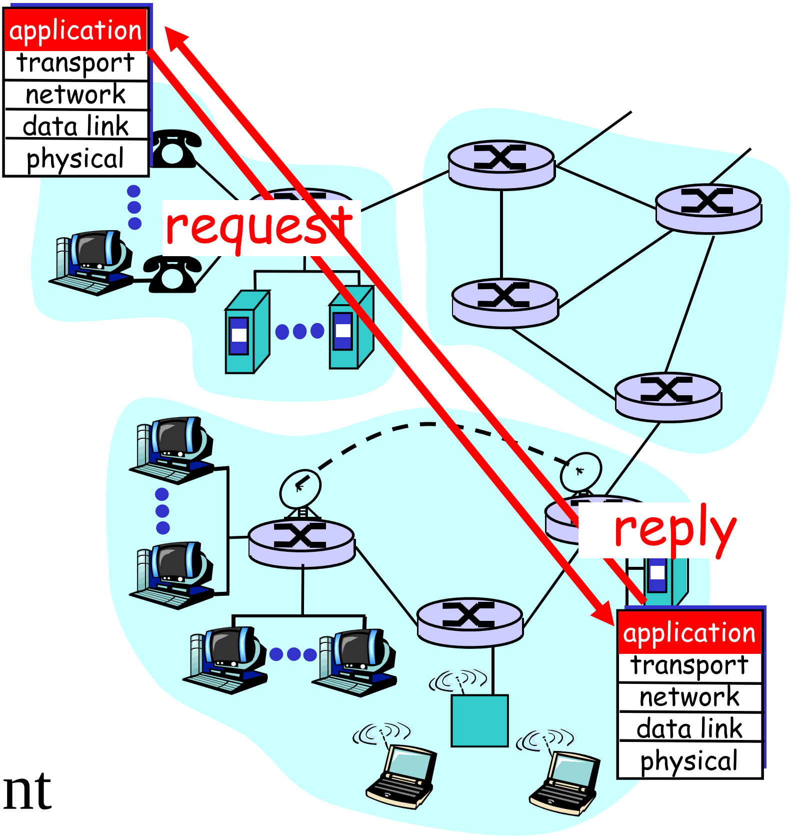
Typical network app has two pieces: *client* and *server*

Client:

- initiates contact with server
- typically requests service from server

Server:

- provides requested service to client



Application-layer Protocols

Q: how does a process “identify” the other process with which it wants to communicate?

– IP address

of host running other process

– Port number

allows receiving host to determine to which local process the message should be delivered

Transport Services

Data loss

- some apps can tolerate some loss
- other apps require 100% reliable data transfer

Timing

- some apps require low delay to be "effective"

Bandwidth

- some apps require minimum amount of bandwidth to be "effective"
- other apps ("elastic apps") make use of whatever bandwidth they get

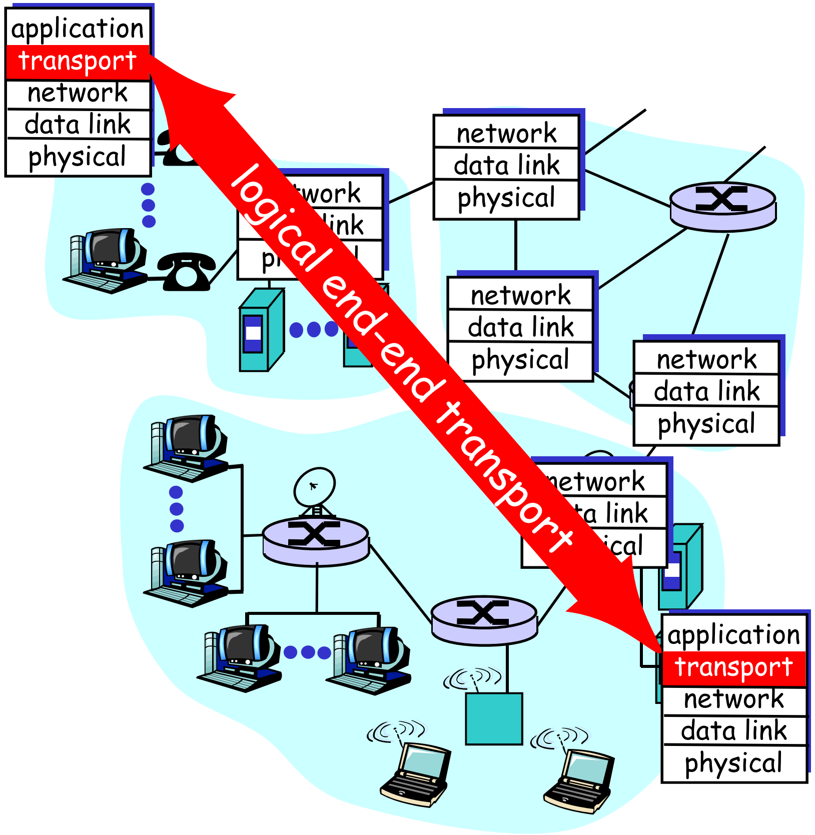
Transport Service Requirements

Application	Data loss	Bandwidth	Time Sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	loss-tolerant	elastic	no
real-time audio/video	loss-tolerant	audio: 5Kb-1Mb video:10Kb-5Mb	yes, 100's msec
stored audio/video	loss-tolerant	same as above	yes, few secs
interactive games	loss-tolerant	few Kbps up	yes, 100's msec
financial apps	no loss	elastic	

Transport-Layer Protocols

Services:

- Reliable, in-order delivery (TCP)
- Unreliable (“best-effort”), unordered delivery: UDP
- Services not available:
  - real-time
  - bandwidth guarantees



TCP Services

- *Connection-oriented:* setup required between client and server
- *Reliable transport* between sending and receiving processes
- *Flow control:* sender won’t overwhelm receiver
- *Congestion control:* throttle sender when network overloaded
- *No*
  - timing
  - minimum bandwidth guarantees

UDP Services

- unreliable data transfer between sending and receiving processes
- Does not provide
  - connection setup
  - reliability
  - flow control
  - congestion control
  - timing guarantee
  - bandwidth guarantee

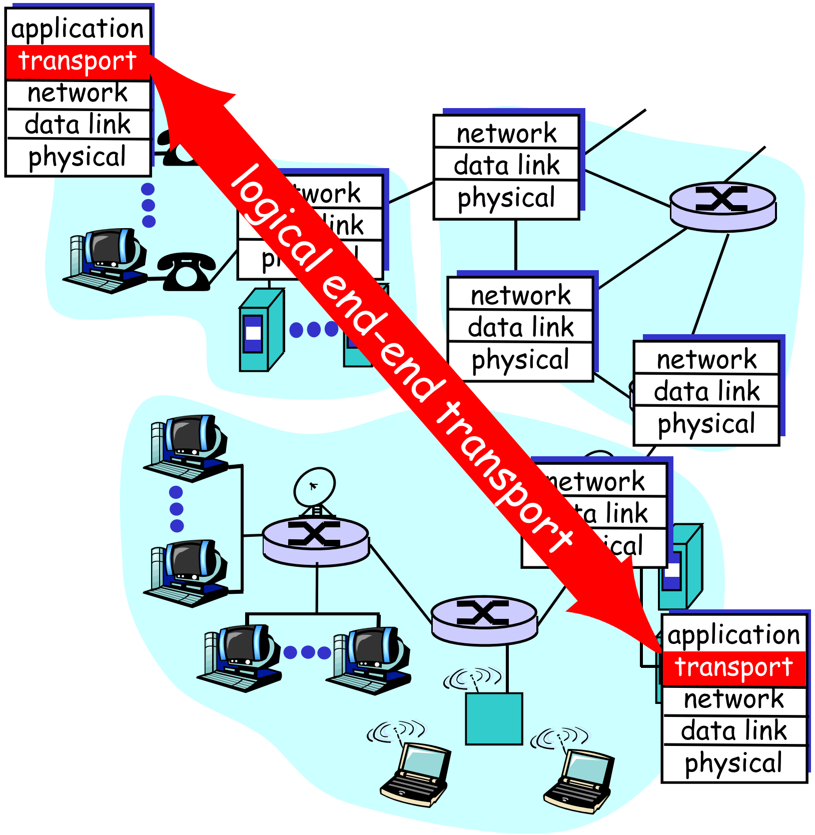
Q: Why is there a UDP?

Internet Application and Transport Protocols

Application	Application layer protocol	Underlying transport protocol
e-mail	smtp [RFC 821]	TCP
remote terminal access	telnet [RFC 854]	TCP
Web	http [RFC 2068]	TCP
file transfer	ftp [RFC 959]	TCP
streaming multimedia	proprietary (e.g. RealNetworks)	TCP or UDP
Internet telephony	proprietary	typically UDP

Transport Services and Protocols

- Provide *logical communication* between processes running on different hosts
- Transport protocols run in end systems
- *Network layer*
  - data transfer between end systems
- *Transport layer*
  - data transfer between processes

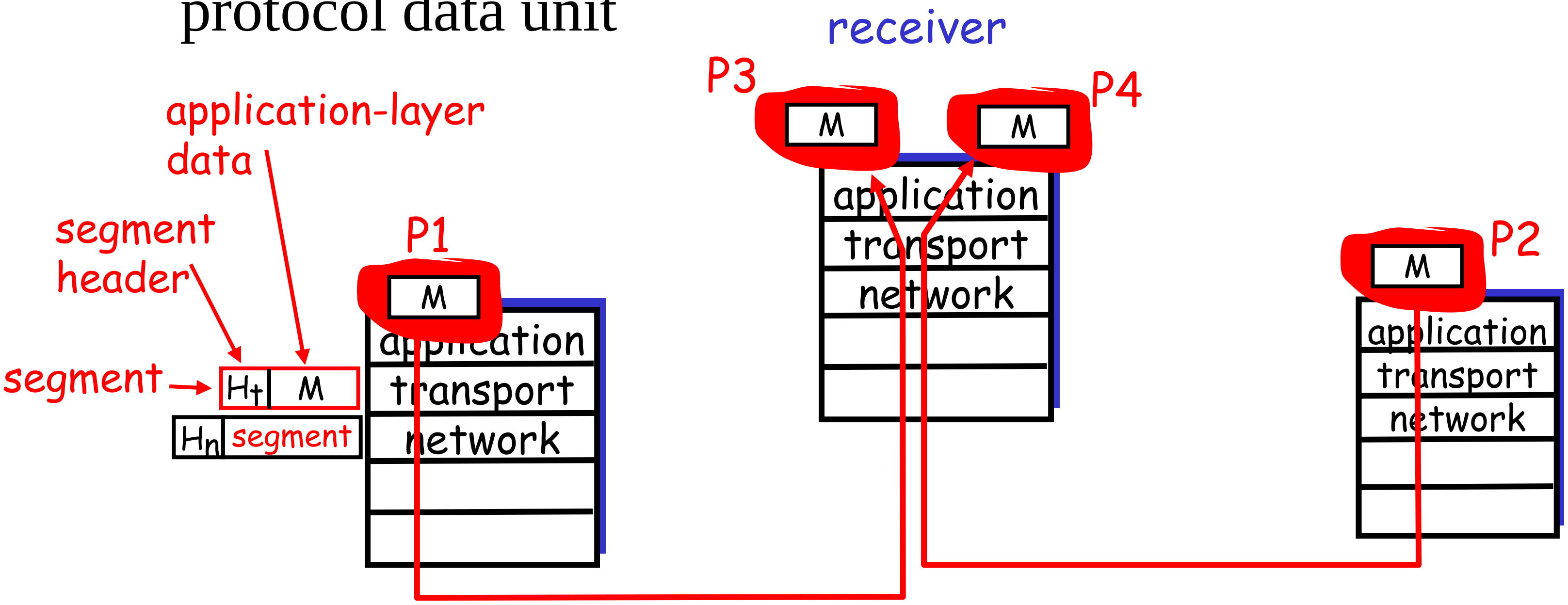


Demultiplexing

Recall: *segment* - unit of data exchanged between transport layer entities

- aka TPDU: transport protocol data unit

Demultiplexing: delivering received segments to correct app layer processes



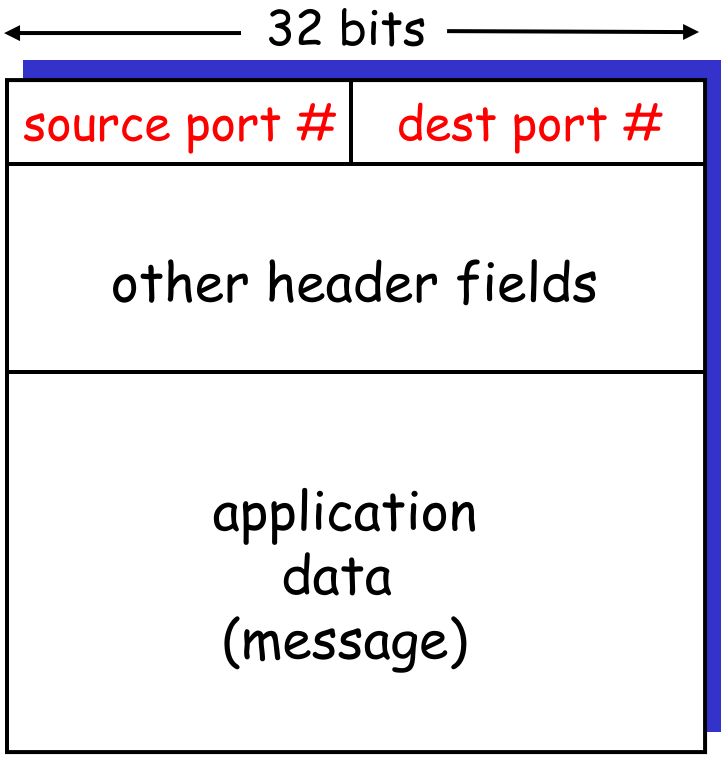
Multiplexing

Multiplexing:

gathering data from multiple app processes, enveloping data with header (later used for demultiplexing)

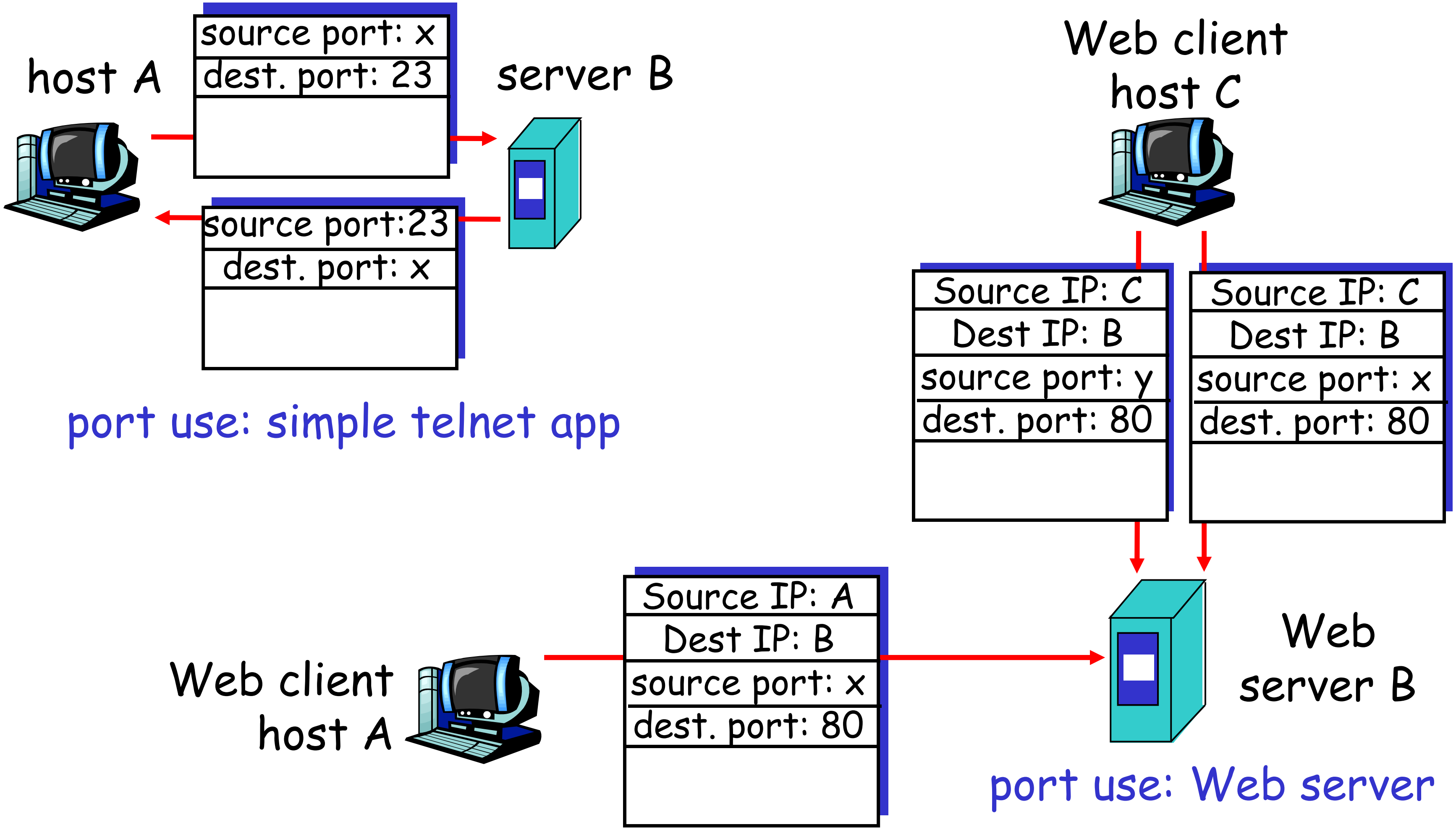
- Well-know port numbers defined in RFC 1700, e.g.,

HTTP:	80
FTP:	21
Telnet:	23



TCP/UDP segment format

Examples



host A

source port: x
dest. port: 23

server B

source port:23
dest. port: x

port use: simple telnet app

Web client
host C

Source IP: C
Dest IP: B
source port: y
dest. port: 80

Source IP: C
Dest IP: B
source port: x
dest. port: 80

Web client
host A

Source IP: A
Dest IP: B
source port: x
dest. port: 80

Web
server B

port use: Web server

UDP: User Datagram Protocol

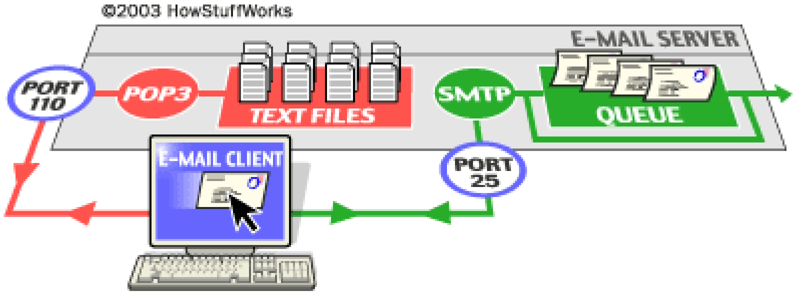
- "Bare Bones" Internet transport protocol
- "Best effort" service, UDP segments may be:
  - lost
  - delivered out of order to app
- *Connectionless:*
  - no handshaking between UDP sender, receiver
  - each UDP segment handled independently of others

Why is there a UDP?

- no connection establishment (which can add delay)
- simple: no connection state at sender, receiver
- Often used for streaming multimedia apps
  - loss tolerant
  - rate sensitive

The Real E-mail System

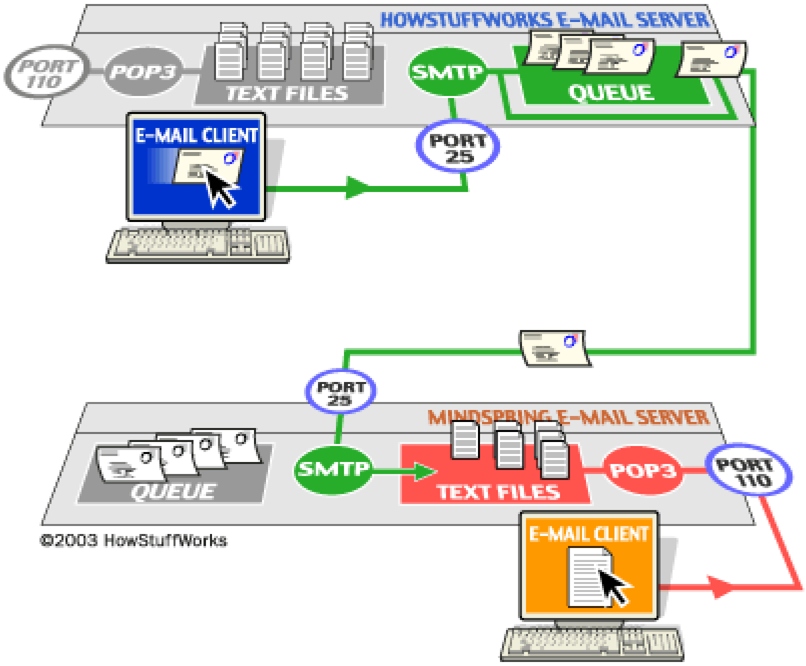
For the vast majority of people right now, the real e-mail system consists of two different servers running on a server machine. One is called the **SMTP server**, where SMTP stands for Simple Mail Transfer Protocol. The SMTP server handles outgoing mail. The other is either a **POP3 server** or an **IMAP server**, both of which handle incoming mail. POP stands for Post Office Protocol, and IMAP stands for Internet Mail Access Protocol. A typical e-mail server looks like this:



The SMTP server listens on well-known port number 25, POP3 listens on port 110 and IMAP uses port 143.

The SMTP Server

Whenever you send a piece of e-mail, your e-mail client interacts with the SMTP server to handle the sending. The SMTP server on your host may have conversations with other SMTP servers to actually deliver the e-mail.



Let's assume that I want to send a piece of e-mail. My e-mail ID is **brain**, and I have my account on **howstuffworks.com**. I want to send e-mail to **jsmith@mindspring.com**. I am using a stand-alone e-mail client like Outlook Express.

When I set up my account at howstuffworks, I told Outlook Express the name of the mail server -- **mail.howstuffworks.com**. When I compose a message and press the Send button, here is what happens:

1. Outlook Express connects to the SMTP server at mail.howstuffworks.com using **port 25**.
2. Outlook Express has a conversation with the SMTP server, telling the SMTP server the address of the sender and the address of the recipient, as well as the body of the message.
3. The SMTP server takes the "to" address (jsmith@mindspring.com) and breaks it into two parts:
   - The recipient name (jsmith)
   - The domain name (mindspring.com)

   If the "to" address had been another user at howstuffworks.com, the SMTP server would simply hand the message to the POP3 server for howstuffworks.com (using a little program called the **delivery agent**). Since the recipient is at another domain, SMTP needs to communicate with that domain.

4. The SMTP server has a conversation with a **Domain Name Server**, or **DNS**. It says, "Can you give me the IP address of the SMTP server for mindspring.com?" The DNS replies with the one or more IP addresses for the SMTP server(s) that Mindspring operates.
5. The SMTP server at howstuffworks.com connects with the SMTP server at Mindspring using port 25. It has the same simple text conversation that my e-mail client had with the SMTP server for HowStuffWorks, and gives the message to the Mindspring server. The Mindspring server recognizes that the domain name for jsmith is at Mindspring, so it hands the message to Mindspring's POP3 server, which puts the message in jsmith's mailbox.

Note: If, for some reason, the SMTP server at HowStuffWorks cannot connect with the SMTP server at Mindspring, then the message goes into a queue. The SMTP server on most machines uses a program called **sendmail** to do the actual sending, so this queue is called the **sendmail queue**. Sendmail will periodically try to resend the messages in its queue. For example, it might retry every 15 minutes. After four hours, it will usually send you a piece of mail that tells you there is some sort of problem. After five days, most sendmail configurations give up and return the mail to you undelivered.

The POP3 Server

In the simplest implementations of POP3, the server really does maintain a collection of text files -- one for each e-mail account. When a message arrives, the POP3 server simply appends it to the bottom of the recipient's file!

When you check your e-mail, your e-mail client connects to the POP3 server using **port 110**. The POP3 server requires an **account name** and a **password**. Once you have logged in, the POP3 server opens your text file and allows you to access it.

Domain Name Servers

If you spend any time on the Internet sending e-mail or browsing the Web, then you use **domain name servers** without even realizing it. Domain name servers, or DNS, are an incredibly important but completely hidden part of the Internet, and they are fascinating! The DNS system forms one of the largest and most active distributed databases on the planet. Without DNS, the Internet would shut down very quickly.

The Basics

When you use the Web or send an e-mail message, you use a **domain name** to do it. For example, the URL "http://www.howstuffworks.com" contains the domain name **howstuffworks.com**. So does the e-mail address "iknow@howstuffworks.com."

Human-readable names like "howstuffworks.com" are easy for people to remember, but they don't do machines any good. All of the machines use names called **IP addresses** to refer to one another. For example, the machine that humans refer to as "www.howstuffworks.com" has the IP address **216.183.103.150**. Every time you use a domain name, you use the Internet's domain name servers (DNS) to translate the human-readable domain name into the machine-readable IP address. During a day of browsing and e-mailing, you might access the domain name servers hundreds of times!

Domain name servers translate domain names to IP addresses.

Note: Every machine on the Internet has its own IP address. A server has a static IP address that does not change very often. A home machine that is dialing up through a modem often has an IP address that is assigned by the ISP when you dial in. That IP address is unique for your session and may be different the next time you dial in. In this way, an ISP only needs one IP address for each modem it supports, rather than for every customer.

Note: As far as the Internet's machines are concerned, an IP address is all that you need to talk to a server. For example, you can type in your browser the URL **http://216.183.103.150** and you will arrive at the machine that contains the Web server for HowStuffWorks. Domain names are strictly a human convenience.

Domain Names

If we had to remember the IP addresses of all of the Web sites we visit every day, we would all go nuts. Human beings just are not that good at remembering strings of numbers. We are good at remembering words, however, and that is where domain names come in. You probably have hundreds of domain names stored in your head. For example:

- www.howstuffworks.com - a typical name
- www.yahoo.com - the world's best-known name
- www.mit.edu - a popular EDU name
- encarta.msn.com - a Web server that does not start with www
- www.bbc.co.uk - a name using four parts rather than three
- ftp.microsoft.com - an FTP server rather than a Web server

The COM, EDU and UK portions of these domain names are called the **top-level domain** or **first-level domain**. There are several hundred top-level domain names, including COM, EDU, GOV, MIL, NET, ORG and INT, as well as unique two-letter combinations for every country.

Within every top-level domain there is a huge list of **second-level domains**. For example, in the COM first-level domain, you've got:

- howstuffworks
- yahoo
- msn
- microsoft
- plus millions of others...

Educational	Governmental	Organizations	Commercial
.edu, .ac	.gov	.org	.com

The left-most word, such as **www** or **encarta**, is the **host name**. It specifies the name of a specific machine (with a specific IP address) in a domain. A given domain can potentially contain millions of host names as long as they are all unique within that domain.

Distributing Domain Names

Because all of the names in a given domain need to be unique, there has to be a single entity that controls the list and makes sure no duplicates arise. For example, the COM domain cannot contain any duplicate names, and a company called Network Solutions is in charge of maintaining this list. When you register a domain name, it goes through one of several dozen registrars who work with Network Solutions to add names to the list. Network Solutions, in turn, keeps a central database known as the whois database that contains information about the owner and name servers for each domain. If you go to the whois form, you can find information about any domain currently in existence.

The Distributed System

Name servers do two things all day long:

- They accept requests from programs to convert domain names into IP addresses.
- They accept requests from other name servers to convert domain names into IP addresses.

When a request comes in, the name server can do one of four things with it:

- It can answer the request with an IP address because it already knows the IP address for the domain.
- It can contact another name server and try to find the IP address for the name requested. It may have to do this multiple times.
- It can say, "I don't know the IP address for the domain you requested, but here's the IP address for a name server that knows more than I do."
- It can return an error message because the requested domain name is invalid or does not exist.

Name servers cache IP addresses to speed things up

A name server would start its search for an IP address by contacting one of the **root name servers**. The root servers know the IP address for all of the name servers that handle the top-level domains. Your name server would ask the root for www.howstuffworks.com, and the root would say (assuming no caching), "I don't know the IP address for www.howstuffworks.com, but here's the IP address for the COM name server." Obviously, these root servers are vital to this whole process, so:

- There are many of them scattered all over the planet.
- Every name server has a list of all of the known root servers. It contacts the first root server in the list, and if that doesn't work it contacts the next one in the list, and so on.

Creating a New Domain Name

When someone wants to create a new domain, he or she has to do two things:

- Find a name server for the domain name to live on.
- Register the domain name.

Technically, there does not need to be a machine in the domain -- there just needs to be a name server that can handle the requests for the domain name.

There are two ways to get a name server for a domain:

- You can create and administer it yourself.
- You can pay an ISP or hosting company to handle it for you

Most larger companies have their own domain name servers. Most smaller companies pay someone.

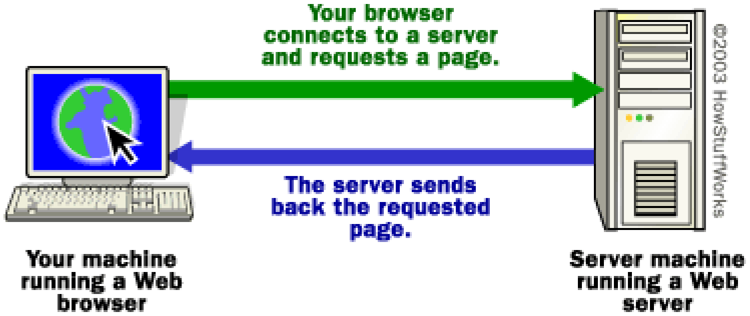
How Web Servers Work

Have you ever wondered about the mechanisms that delivered this page to you? Chances are you are sitting at a computer right now, viewing this page in a browser. So, when you clicked on the link for this page, or typed in its URL (**uniform resource locator**), what happened behind the scenes to bring this page onto your screen?

The Basic Process

Let's say that you are sitting at your computer, surfing the Web, and you get a call from a friend who says, "I just read a great article! Type in this URL and check it out. It's at http://computer.howstuffworks.com/web-server.htm." So you type that URL into your browser and press return. And magically, no matter where in the world that URL lives, the page pops up on your screen.

At the most basic level possible, the following diagram shows the steps that brought that page to your screen:


Your browser connects to a server and requests a page.
The server sends back the requested page.
Your machine running a Web browser
Server machine running a Web server
©2003 HowStuffWorks

Your browser formed a connection to a Web server, requested a page and received it.

Behind the Scenes

If you want to get into a bit more detail on the process of getting a Web page onto your computer screen, here are the basic steps that occurred behind the scenes:

- The browser broke the URL into three parts:
  1. The protocol ("http")
  2. The server name ("www.howstuffworks.com")
  3. The file name ("web-server.htm")
- The browser communicated with a name server to translate the server name "www.howstuffworks.com" into an **IP Address**, which it uses to connect to the server machine.

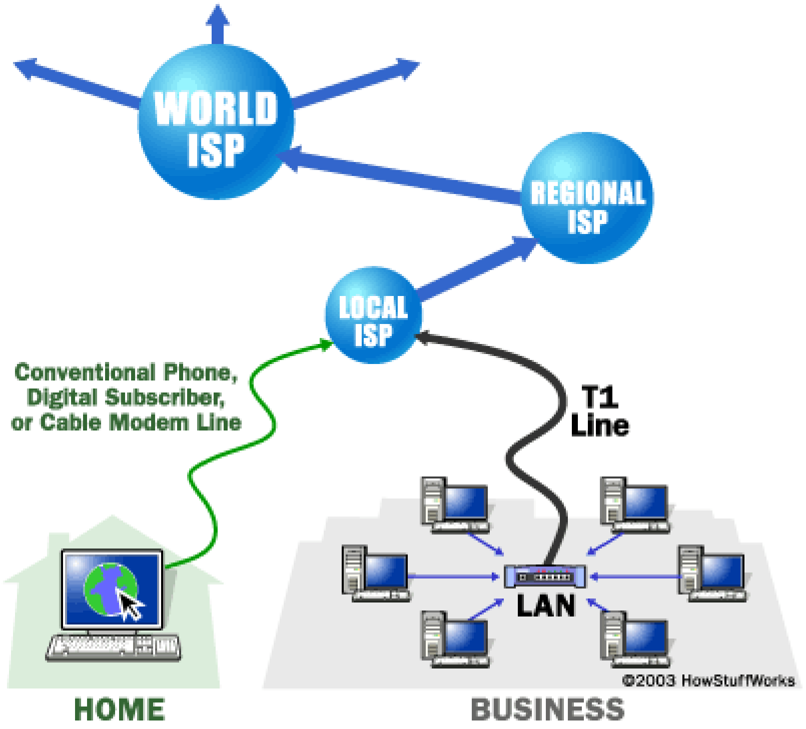
- The browser then formed a connection to the server at that IP address on port 80. Following the HTTP protocol, the browser sent a GET request to the server, asking for the file "http://computer.howstuffworks.com/web-server.htm."
- The server then sent the HTML text for the Web page to the browser.
- The browser read the HTML tags and formatted the page onto your screen.

REMARKS:

The Internet

So what is "the Internet"? The Internet is a gigantic collection of millions of computers, all linked together on a **computer network**. The network allows all of the computers to communicate with one another. A home computer may be linked to the Internet using a phone-line modem, DSL or cable modem that talks to an Internet service provider (**ISP**). A computer in a business or university will usually have a network interface card (**NIC**) that directly connects it to a local area network (**LAN**) inside the business. The business can then connect its LAN to an ISP using a high-speed phone line like a **T1 line**. A T1 line can handle approximately 1.5 million bits per second, while a normal phone line using a modem can typically handle 30,000 to 50,000 bits per second

ISPs then connect to larger ISPs, and the largest ISPs maintain fiber-optic "backbones" for an entire nation or region. Backbones around the world are connected through fiber-optic lines, undersea cables or satellite links. In this way, every computer on the Internet is connected to every other computer on the Internet.



Clients and Servers

In general, all of the machines on the Internet can be categorized as two types: servers and clients. Those machines that provide services (like Web servers or FTP servers) to other machines are servers. And the machines that are used to connect to those services are clients.

A server machine may provide one or more services on the Internet. For example, a server machine might have software running on it that allows it to act as a Web server, an e-mail server and an FTP server. Clients that come to a server machine do so with a specific intent, so clients direct their requests to a specific software server running on the overall server machine. For example, if you are running a Web browser on your machine, it will most likely want to talk to the Web server on the server machine. Your Telnet application will want to talk to the Telnet server, your e-mail application will talk to the e-mail server, and so on...

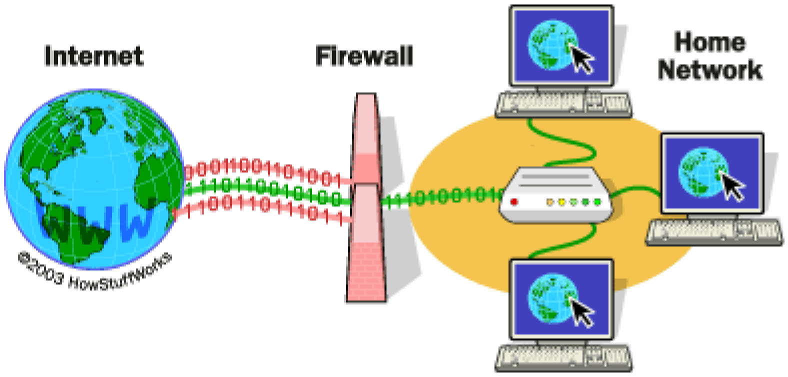
Ports

Any server machine makes its services available to the Internet using numbered ports, one for each service that is available on the server. For example, if a server machine is running a Web server and an FTP server, the Web server would typically be available on port 80, and the FTP server would be available on port 21. Clients connect to a service at a specific IP address and on a specific port.

How Firewalls Work

If you have been using the Internet for any length of time, and especially if you work at a larger company and browse the Web while you are at work, you have probably heard the term **firewall** used. For example, you often hear people in companies say things like, "I can't use that site because they won't let it through the firewall."

If you have a fast Internet connection into your home (either a DSL connection or a cable modem), you may have found yourself hearing about firewalls for your home network as well. It turns out that a small home network has many of the same security issues that a large corporate network does. You can use a firewall to protect your home network and family from offensive Web sites and potential hackers.



Basically, a firewall is a barrier to keep destructive forces away from your property. In fact, that's why its called a firewall. Its job is similar to a physical firewall that keeps a fire from spreading from one area to the next. As you read through this article, you will learn more about firewalls, how they work and what kinds of threats they can protect you from.

A firewall is simply a program or hardware device that filters the information coming through the Internet connection into your private network or computer system. If an incoming packet of information is flagged by the filters, it is not allowed through.

Firewalls are customizable. This means that you can add or remove filters based on several conditions. Some of these are:

- IP addresses - Each machine on the Internet is assigned a unique address called an **IP address**. IP addresses are 32-bit numbers, normally expressed as four "octets" in a "dotted decimal number." A typical IP address looks like this: 216.27.61.137. For example, if a certain IP address outside the company is reading too many files from a server, the firewall can block all traffic to or from that IP address.
- Domain names - Because it is hard to remember the string of numbers that make up an IP address, and because IP addresses sometimes need to change, all servers on the Internet also have human-readable names, called **domain names**. For example, it is easier for most of us to remember www.howstuffworks.com than it is to remember 216.27.61.137. A company might block all access to certain domain names, or allow access only to specific domain names.
- Ports - Any server machine makes its services available to the Internet using numbered **ports**, one for each service that is available on the server. For example, if a server machine is running a Web (HTTP) server and an FTP server, the Web server would typically be available on port 80, and the FTP server would be available on port 21. A company might block port 21 access on all machines but one inside the company.

Proxy Servers

- A function that is often combined with a firewall is a **proxy server**. The proxy server is used to access Web pages by the other computers. When another computer requests a Web page, it is retrieved by the proxy server and then sent to the requesting computer.

The net effect of this action is that the remote computer hosting the Web page never comes into direct contact with anything on your home network, other than the proxy server.

- Proxy servers can also make your Internet access work more efficiently. If you access a page on a Web site, it is **cached** (stored) on the proxy server. This means that the next time you go back to that page, it normally doesn't have to load again from the Web site. Instead it loads instantaneously from the proxy server.

Security

You can see from this description that a Web server can be a pretty simple piece of software. It takes the file name sent in with the GET command, retrieves that file and sends it down the wire to the browser.

Most servers add some level of **security** to the serving process. For example, if you have ever gone to a Web page and had the browser pop up a dialog box asking for your name and password, you have encountered a password-protected page. The server lets the owner of the page maintain a list of names and passwords for those people who are allowed to access the page; the server lets only those people who know the proper password see the page. More advanced servers add further security to allow an encrypted connection between server and browser, so that sensitive information like credit card numbers can be sent on the Internet.

There is a lot of information that we don't want other people to see, such as:

- Credit-card information
- Social Security numbers
- Private correspondence
- Personal details
- Sensitive company information
- Bank-account information