

(R15A0513) Computer Networks

Objectives:

- To introduce the fundamental types of computer networks.
- To demonstrate the TCP/IP & OSI model merits & demerits.
- To know the role of various protocols in Networking

UNIT - I:

Introduction: Network, Uses of Networks, Types of Networks, Reference Models: TCP/IP Model, The OSI Model, Comparison of the OSI and TCP/IP reference model. Architecture of Internet.

Physical Layer: Guided transmission media, Wireless transmission media, Switching

UNIT - II:

Data Link Layer - Design issues, Error Detection & Correction, Elementary Data Link Layer Protocols, Sliding window protocols

Multiple Access Protocols - ALOHA, CSMA, CSMA/CD, CSMA/CA, Collision free protocols, Ethernet- Physical Layer, Ethernet Mac Sub layer, Data link layer switching: Use of bridges, learning bridges, spanning tree bridges, repeaters, hubs, bridges, switches, routers and gateways.

UNIT - III:

Network Layer: Network Layer Design issues, store and forward packet switching connection less and connection oriented networks-routing algorithms-optimality principle, shortest path, flooding, Distance Vector Routing, Count to Infinity Problem, Link State Routing, Path Vector Routing, Hierarchical Routing; Congestion control algorithms, IP addresses, CIDR, Subnetting, SuperNetting, IPv4, Packet Fragmentation, IPv6 Protocol, Transition from IPv4 to IPv6, ARP, RARP.

UNIT - IV:

Transport Layer: Services provided to the upper layers elements of transport protocol addressing connection establishment, Connection release, Error Control & Flow Control, Crash Recovery.

The Internet Transport Protocols: UDP, Introduction to TCP, The TCP Service Model, The TCP Segment Header, The Connection Establishment, The TCP Connection Release, The TCP Sliding Window, The TCP Congestion Control Algorithm.

UNIT - V: Application Layer- Introduction, providing services, Applications layer paradigms: Client server model, HTTP, E-mail, WWW, TELNET, DNS; RSA algorithm,

TEXT BOOKS:

1. Computer Networks - Andrew S Tanenbaum, 4th Edition, Pearson Education.
2. Data Communications and Networking - Behrouz A. Forouzan, Fifth Edition TMH, 2013.

REFERENCES BOOKS:

1. An Engineering Approach to Computer Networks - S. Keshav, 2nd Edition, Pearson Education.
2. Understanding communications and Networks, 3rd Edition, W. A. Shay, Cengage Learning.
3. Computer Networking: A Top-Down Approach Featuring the Internet, James F. Kurose, K. W. Ross, 3rd Edition, Pearson Education.

Outcomes:

- Students should be understand and explore the basics of Computer Networks and Various Protocols.
- Student will be in a position to understand the World Wide Web concepts.
- Students will be in a position to administrate a network and flow of information further
- Student can understand easily the concepts of network security, Mobile

communication links. A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network.

“Computer network” to mean a collection of autonomous computers interconnected by a single technology. Two computers are said to be interconnected if they are able to exchange information.

The connection need not be via a copper wire; fiber optics, microwaves, infrared, and communication satellites can also be used.

Networks come in many sizes, shapes and forms, as we will see later. They are usually connected together to make larger networks, with the **Internet** being the most well-known example of a network of networks.

There is considerable confusion in the literature between a **computer network** and a **distributed system**. The key distinction is that in a distributed system, a collection of independent computers appears to its users as a single coherent system. Usually, it has a single model or paradigm that it presents to the users. Often a layer of software on top of the operating system, called **middleware**, is responsible for implementing this model. A well-known example of a distributed system is the **World Wide Web**. It runs on top of the Internet and presents a model in which everything looks like a document (Web page).

USES OF COMPUTER NETWORKS

1. Business Applications

- to distribute information throughout the company (**resource sharing**). sharing physical resources such as printers, and tape backup systems, is sharing information
- **client-server model**. It is widely used and forms the basis of much network usage.
- **communication medium** among employees.**email (electronic mail)**, which employees generally use for a great deal of daily communication.
- Telephone calls between employees may be carried by the computer network instead of by the phone company. This technology is called **IP telephony** or **Voice over IP (VoIP)** when Internet technology is used.
- **Desktop sharing** lets remote workers see and interact with a graphical computer screen
- doing business electronically, especially with customers and suppliers. This new model is called **e-commerce (electronic commerce)** and it has grown rapidly in recent years.

2 Home Applications

- **peer-to-peer** communication
- person-to-person communication

3 Mobile Users

- Text messaging or texting
- Smart phones,
- GPS (Global Positioning System)
- m-commerce
- NFC (Near Field Communication)

4 Social Issues

With the good comes the bad, as this new-found freedom brings with it many unsolved social, political, and ethical issues.

Social networks, message boards, content sharing sites, and a host of other applications allow people to share their views with like-minded individuals. As long as the subjects are restricted to technical topics or hobbies like gardening, not too many problems will arise.

The trouble comes with topics that people actually care about, like politics, religion, or sex. Views that are publicly posted may be deeply offensive to some people. Worse yet, they may not be politically correct. Furthermore, opinions need not be limited to text; high-resolution color photographs and video clips are easily shared over computer networks. Some people take a live-and-let-live view, but others feel that posting certain material (e.g., verbal attacks on particular countries or religions, pornography, etc.) is simply unacceptable and that such content must be censored. Different countries have different and conflicting laws in this area. Thus, the debate rages.

Computer networks make it very easy to communicate. They also make it easy for the people who run the network to snoop on the traffic. This sets up conflicts over issues such as **employee rights versus employer rights**. Many people read and write email at work. Many employers have claimed the right to read and possibly censor employee messages, including messages sent from a home computer outside working hours. Not all employees agree with this, especially the latter part.

Another conflict is centered around government versus citizen's rights.

A new twist with mobile devices is location privacy. As part of the process of providing service to your mobile device the network operators learn where you are at different times of day. This allows them to track your movements. They may know which nightclub you frequent and which medical center you visit.

Phishing ATTACK: *Phishing* is a type of social engineering **attack** often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message.

The effectiveness of a data communications system depends on four fundamental characteristics: delivery, accuracy, timeliness, and jitter.

1. **Delivery.** The system must deliver data to the correct destination. Data must be received by the intended device or user and only by that device or user.

2 **Accuracy.** The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable.

3. **Timeliness.** The system must deliver data in a timely manner. Data delivered late are useless. In the case of video and audio, timely delivery means delivering data as they are produced, in the same order that they are produced, and without significant delay. This kind of delivery is called *real-time* transmission.

4. **Jitter.** Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets. For example, let us assume that video packets are sent every 30 ms. If some of the packets arrive with 30-ms delay and others with 40-ms delay, an uneven quality in the video is the result.

A data communications system has five components

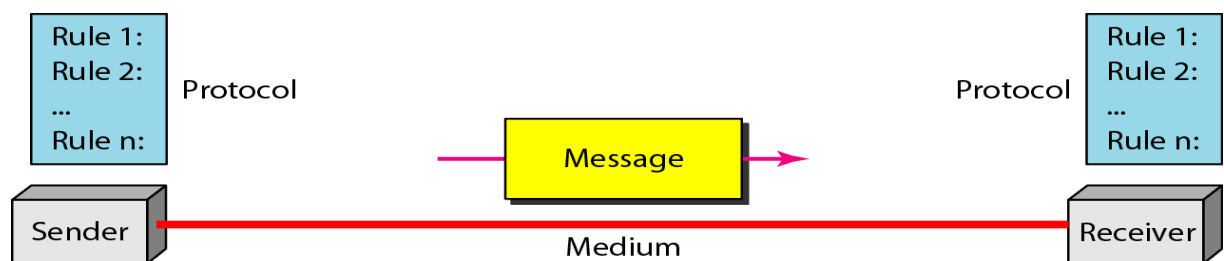
1. **Message.** The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.

2 **Sender.** The sender is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.

3. **Receiver.** The receiver is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.

4. **Transmission medium.** The transmission medium is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves.

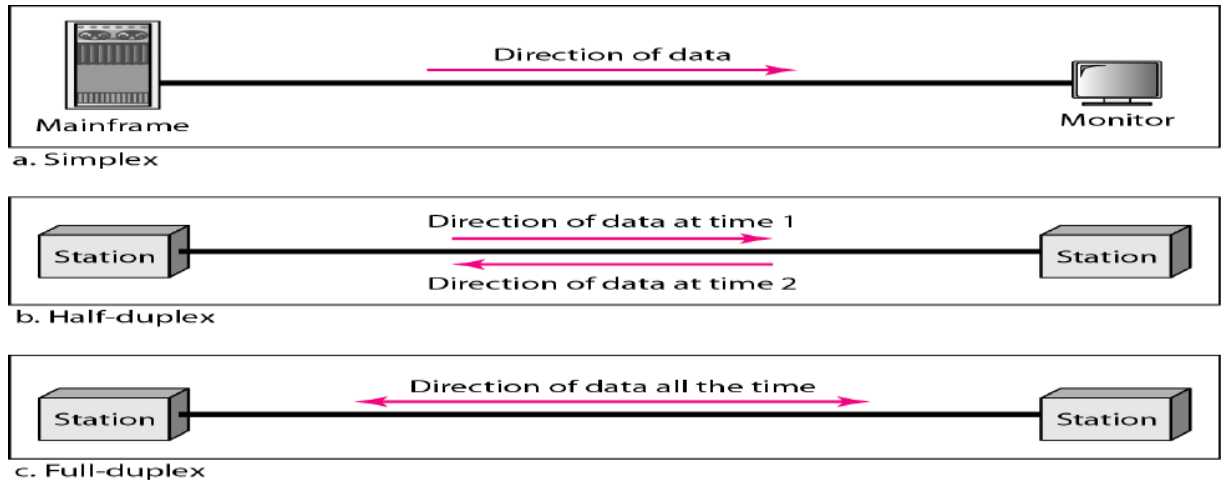
5. **Protocol.** A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating, just as a person speaking French cannot be understood by a person who speaks only Japanese.



Data Representation

Data Flow

Communication between two devices can be simplex, half-duplex, or full-duplex as shown in Figure.



Simplex In simplex mode, the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive (Figure a). Keyboards and traditional monitors are examples of simplex devices.

Half-Duplex

In half-duplex mode, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa (Figure b). Walkie-talkies and CB (citizens band) radios are both half-duplex systems.

Full-Duplex

In full-duplex, both stations can transmit and receive simultaneously (Figure c). One common example of full-duplex communication is the telephone network. When two people are communicating by a telephone line, both can talk and listen at the same time. The full-duplex mode is used when communication in both directions is required all the time.

Network Criteria

A network must be able to meet a certain number of criteria. The most important of these are performance, reliability, and security.

Performance

Performance can be measured in many ways, including transit time and response time. Transit time is the amount of time required for a message to travel from one device to another. Response time is the elapsed time between

Performance is often evaluated by two networking metrics: **throughput and delay**. We often need more throughput and less delay. However, these two criteria are often contradictory. If we try to send more data to the network, we may increase throughput but we increase the delay because of traffic congestion in the network.

Reliability: In addition to accuracy of delivery, network reliability is measured by the frequency of failure, the time it takes a link to recover from a failure, and the network's robustness in a catastrophe.

Security: Network security issues include protecting data from unauthorized access, protecting data from damage and development, and implementing policies and procedures for recovery from breaches and data losses.

Physical Structures

Before discussing networks, we need to define some network attributes.

Type of Connection

A network is two or more devices connected through links. A link is a communications pathway that transfers data from one device to another.

There are two possible types of connections: point-to-point and multipoint.

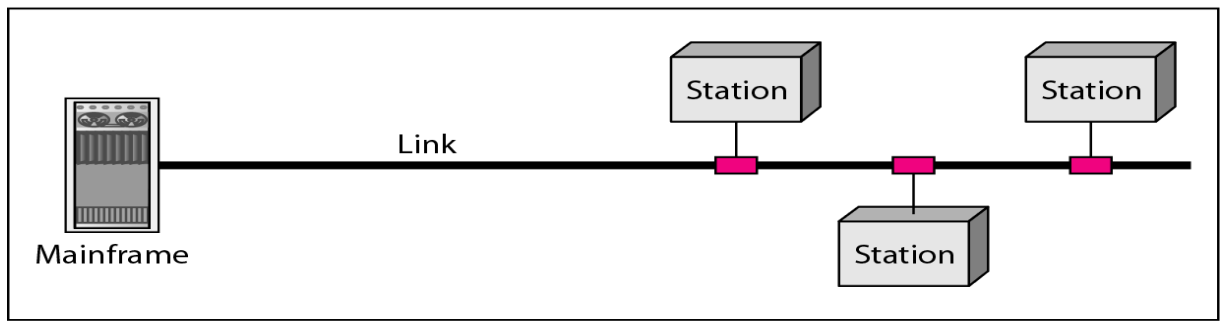
Point-to-Point A point-to-point connection provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices. Most point-to-point connections use an actual length of wire or cable to connect the two ends, but other options, such as microwave or satellite links, are also possible.

When you change television channels by infrared remote control, you are establishing a point-to-point connection between the remote control and the television's control system.

Multipoint A multipoint (also called multi-drop) connection is one in which more than two specific devices share a single link.

In a multipoint environment, the capacity of the channel is shared, either spatially or temporally. If several devices can use the link simultaneously, it is a *spatially shared* connection. If users must take turns, it is a *timeshared* connection.

a. Point-to-point



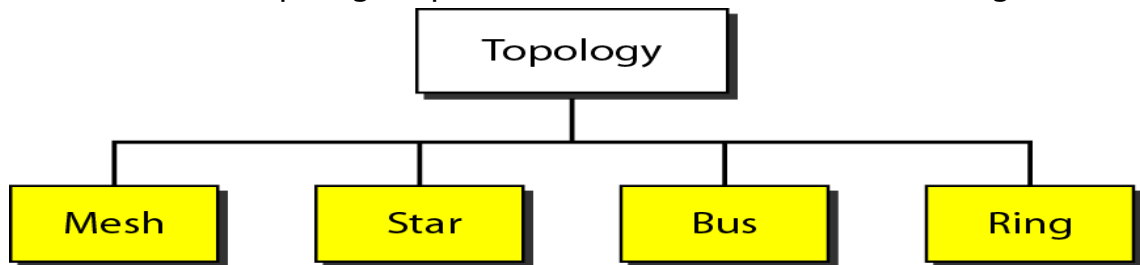
b. Multipoint

Physical Topology

The term *physical topology* refers to the way in which a network is laid out physically.

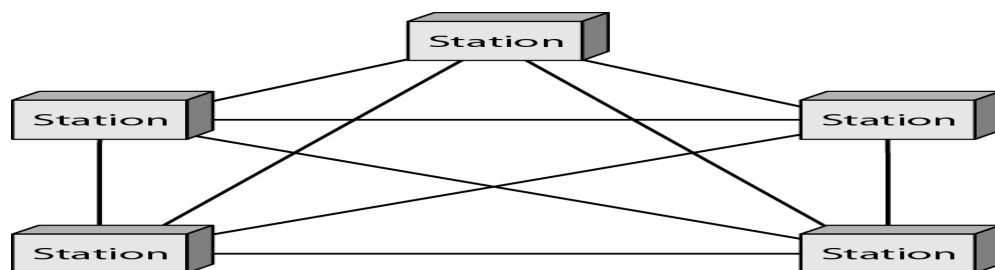
Two or more devices connect to a link; two or more links form a topology. The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to one another.

There are four basic topologies possible: mesh, star, bus, and ring



MESH:

A mesh topology is the one where every node is connected to every other node in the network.



A mesh topology can be a **full mesh topology** or a **partially connected mesh topology**.

In a *full mesh topology*, every computer in the network has a connection to each of the other computers in that network. The number of connections in this

network have connections to multiple other computers in that network. It is an inexpensive way to implement redundancy in a network. In the event that one of the primary computers or connections in the network fails, the rest of the network continues to operate normally.

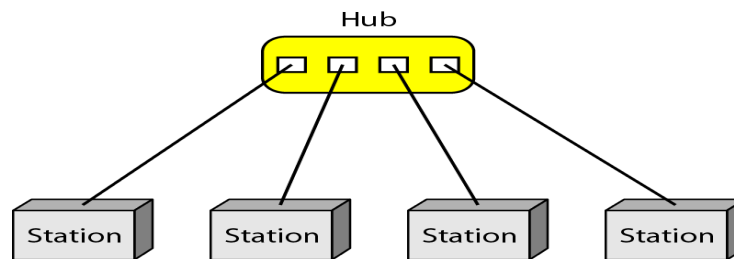
Advantages of a mesh topology

- Can handle high amounts of traffic, because multiple devices can transmit data simultaneously.
- A failure of one device does not cause a break in the network or transmission of data.
- Adding additional devices does not disrupt data transmission between other devices.

Disadvantages of a mesh topology

- The cost to implement is higher than other network topologies, making it a less desirable option.
- Building and maintaining the topology is difficult and time consuming.
- The chance of redundant connections is high, which adds to the high costs and potential for reduced efficiency.

STAR:



A star network, star topology is one of the most common network setups. In this configuration, every [node](#) connects to a central network device, like a [hub](#), [switch](#), or computer. The central network device acts as a [server](#) and the peripheral devices act as [clients](#). Depending on the type of [network card](#) used in each computer of the star topology, a [coaxial cable](#) or a [RJ-45](#) network cable is used to connect computers together.

Advantages of star topology

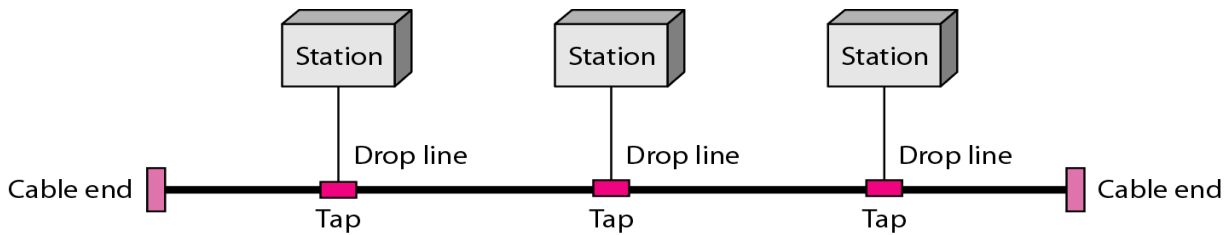
- Centralized management of the network, through the use of the central computer, hub, or switch.
- Easy to add another computer to the network.
- If one computer on the network fails, the rest of the network continues to function normally.
- The star topology is used in local-area networks (LANs), High-speed LANs often use a star topology with a central hub.

Disadvantages of star topology

nodes the network can handle.

- If the central computer, hub, or switch fails, the entire network goes down and all computers are disconnected from the network

BUS:



a **line topology**, a **bus topology** is a network setup in which each computer and network device are connected to a single cable or [backbone](#).

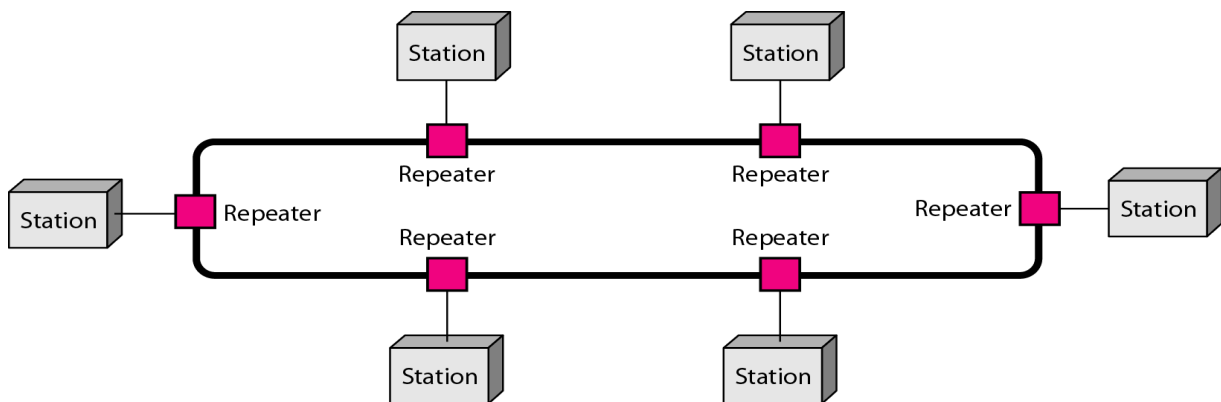
Advantages of bus topology

- It works well when you have a small network.
- It's the easiest network topology for connecting computers or peripherals in a linear fashion.
- It requires less cable length than a star topology.

Disadvantages of bus topology

- It can be difficult to identify the problems if the whole network goes down.
- It can be hard to troubleshoot individual device issues.
- Bus topology is not great for large networks.
- Terminators are required for both ends of the main cable.
- Additional devices slow the network down.
- If a main cable is damaged, the network fails or splits into two.

RING:



travel only in one direction, called a **unidirectional** ring network. Others permit data to move in either direction, called **bidirectional**.

The major disadvantage of a ring topology is that if any individual connection in the ring is broken, the entire network is affected.

Ring topologies may be used in either local area networks ([LANs](#)) or wide area networks ([WANs](#)).

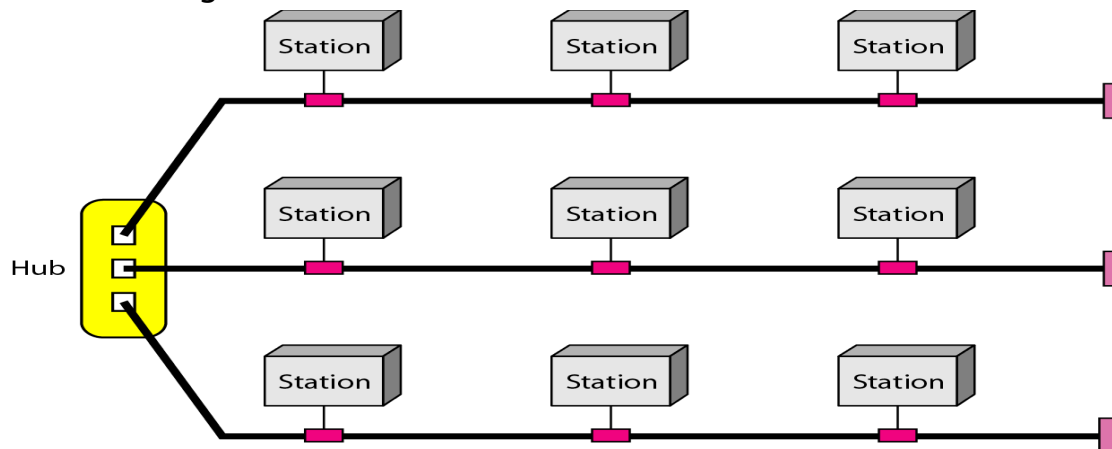
Advantages of ring topology

- All data flows in one direction, reducing the chance of packet collisions.
- A network server is not needed to control network connectivity between each workstation.
- Data can transfer between workstations at high speeds.
- Additional workstations can be added without impacting performance of the network.

Disadvantages of ring topology

- All data being transferred over the network must pass through each workstation on the network, which can make it slower than a [star topology](#).
- The entire network will be impacted if one workstation shuts down.
- The hardware needed to connect each workstation to the network is more expensive than Ethernet cards and hubs/switches.

Hybrid Topology A network can be hybrid. For example, we can have a main star topology with each branch connecting several stations in a bus topology as shown in Figure



Types of Network based on size

The types of network are classified based upon the size, the area it covers and its physical architecture. The three primary network categories are LAN, WAN and MAN. Each network differs in their characteristics such as distance, transmission speed, cables and cost.

Basic types

Two or more pc's can form a LAN to share files, folders, printers, applications and other devices.

Coaxial or CAT 5 cables are normally used for connections.

Due to short distances, errors and noise are minimum.

Data transfer rate is 10 to 100 mbps.

Example: A computer lab in a school.

MAN (Metropolitan Area Network)

Design to extend over a large area.

Connecting number of LAN's to form larger network, so that resources can be shared.

Networks can be up to 5 to 50 km.

Owned by organization or individual.

Data transfer rate is low compare to LAN.

Example: Organization with different branches located in the city.

WAN (Wide Area Network)

Are country and worldwide network.

Contains multiple LAN's and MAN's.

Distinguished in terms of geographical range.

Uses satellites and microwave relays.

Data transfer rate depends upon the ISP provider and varies over the location.

Best example is the internet.

Other types

WLAN (Wireless LAN)

A LAN that uses high frequency radio waves for communication.

Provides short range connectivity with high speed data transmission.

PAN (Personal Area Network)

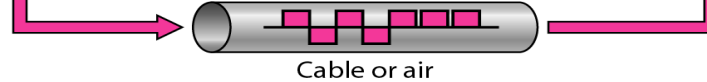
Network organized by the individual user for its personal use.

SAN (Storage Area Network)

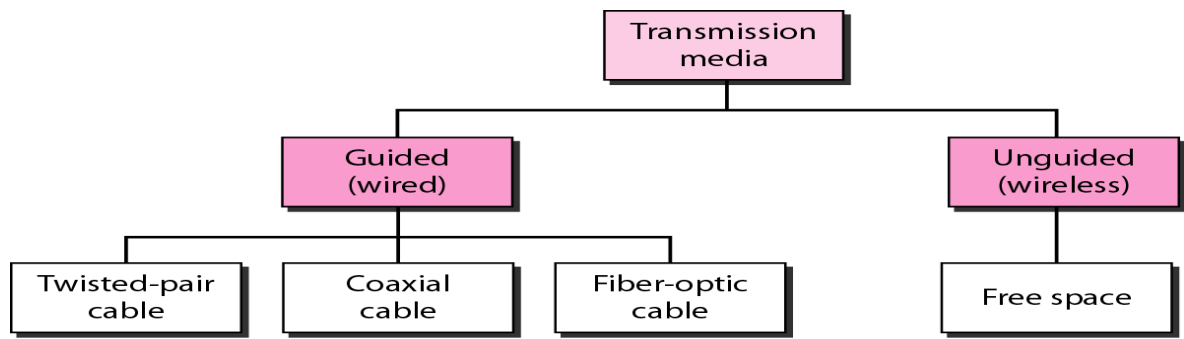
Connects servers to data storage devices via fiber-optic cables.

E.g.: Used for daily backup of organization or a mirror copy

A **transmission medium** can be broadly defined as anything that can carry information from a source to a destination.

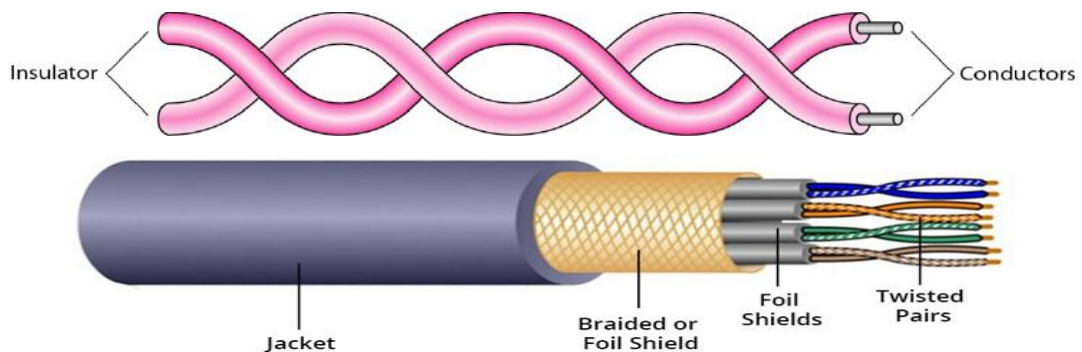


Classes of transmission media



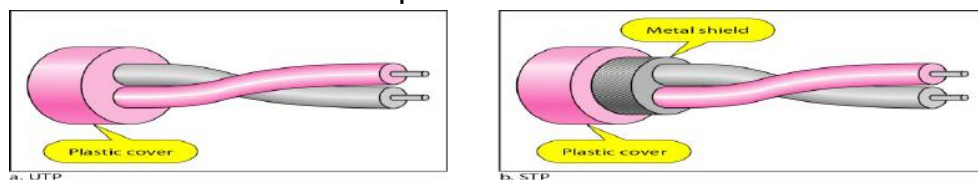
Guided Media: Guided media, which are those that provide a medium from one device to another, include twisted-pair cable, coaxial cable, and fiber-optic cable.

Twisted-Pair Cable: A twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together. One of the wires is used to carry signals to the receiver, and the other is used only as a ground reference.



Unshielded Versus Shielded Twisted-Pair Cable

The most common twisted-pair cable used in communications is referred to as unshielded twisted-pair (UTP). STP cable has a metal foil or braided mesh covering that encases each pair of insulated conductors. Although metal casing improves the quality of cable by preventing the penetration of noise or crosstalk, it is bulkier and more expensive.

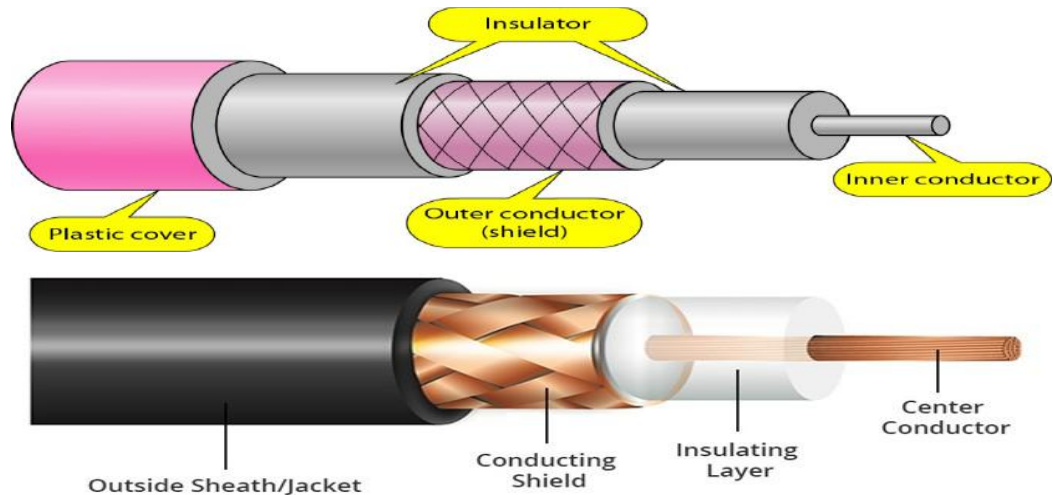


The most common UTP connector is RJ45 (RJ stands for registered jack)

Local-area networks, such as 10Base-T and 100Base-T, also use twisted-pair cables.

Coaxial Cable

Coaxial cable (or coax) carries signals of higher frequency ranges than those in twisted pair cable. Coax has a central core conductor of solid or stranded wire (usually copper) enclosed in an insulating sheath, which is, in turn, encased in an outer conductor of metal foil, braid, or a combination of the two. The outer metallic wrapping serves both as a shield against noise and as the second conductor, which completes the circuit. This outer conductor is also enclosed in an insulating sheath, and the whole cable is protected by a plastic cover.



The most common type of connector used today is the Bayonet-Neill-Concelman (BNC), connector.

Applications

Coaxial cable was widely used in analog telephone networks, digital telephone networks

Cable TV networks also use coaxial cables.

Another common application of coaxial cable is in traditional Ethernet LANs

Fiber-Optic Cable

A fiber-optic cable is made of glass or plastic and transmits signals in the form of light. Light travels in a straight line as long as it is moving through a single uniform substance.

If a ray of light traveling through one substance suddenly enters another substance (of a different density), the ray changes direction.

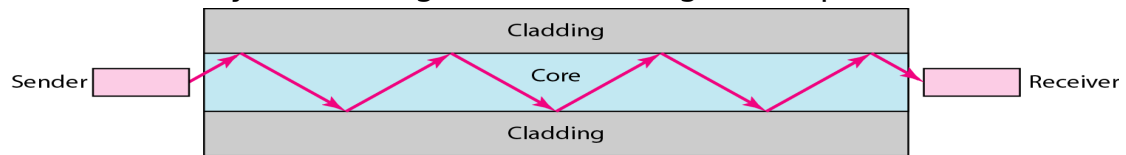
Bending of light ray

$\theta < \theta_c$,
refraction

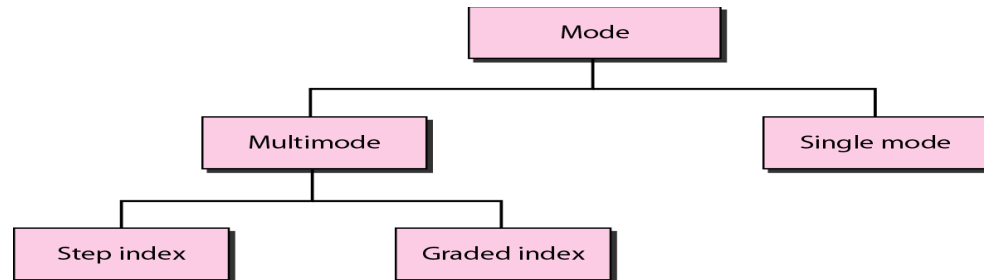
$\theta = \theta_c$,
refraction

$\theta > \theta_c$,
reflection

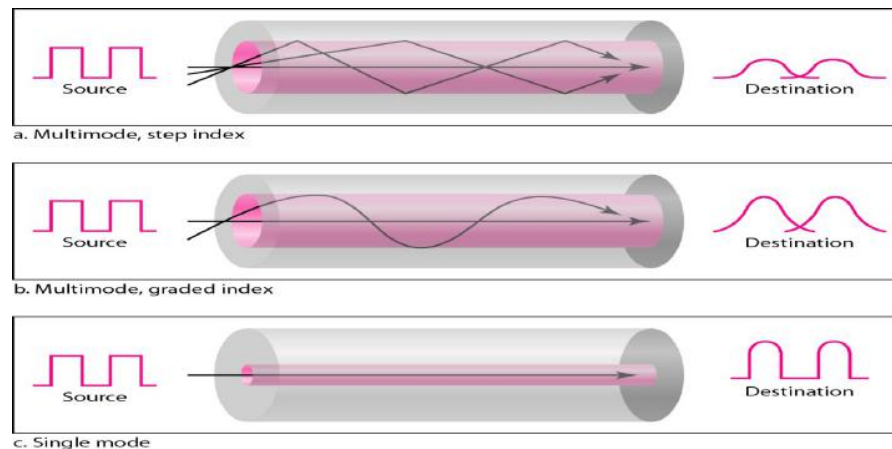
Optical fibers use reflection to guide light through a channel. A glass or plastic core is surrounded by a cladding of less dense glass or plastic.



Propagation Modes



Multimode is so named because multiple beams from a light source move through the core in different paths. How these beams move within the cable depends on the structure of the core, as shown in Figure.



In **multimode step-index fiber**, the density of the core remains constant from the center to the edges. A beam of light moves through this constant density in a straight line until it reaches the interface of the core and the cladding. The term *step index* refers to the suddenness of this change, which contributes to the distortion of the signal as it passes through the fiber.

A second type of fiber, called **multimode graded-index fiber**, decreases this distortion of the signal through the cable. The word *index* here refers to the index of refraction.

Single-Mode: Single-mode uses step-index fiber and a highly focused source of light that limits beams to a small range of angles, all close to the horizontal.



The **subscriber channel (SC) connector**, The **straight-tip (ST) connector**, **MT-RJ(mechanical transfer registered jack)** is a connector

Applications

Fiber-optic cable is often found in backbone networks because its wide bandwidth is cost-effective..

Some cable TV companies use a combination of optical fiber and coaxial cable,thus creating a hybrid network.

Local-area networks such as 100Base-FX network (Fast Ethernet) and 1000Base-X also use fiber-optic cable

Advantages and Disadvantages of Optical Fiber

Advantages Fiber-optic cable has several advantages over metallic cable (twisted pair or coaxial).

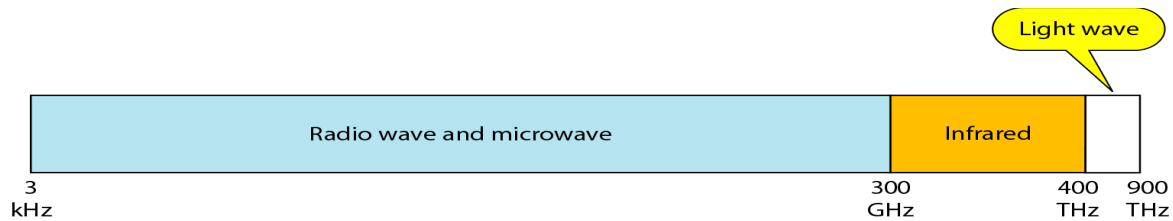
- 1 Higher bandwidth.
- 2 Less signal attenuation. Fiber-optic transmission distance is significantly greater than that of other guided media. A signal can run for 50 km without requiring regeneration. We need repeaters every 5 km for coaxial or twisted-pair cable.
- 3 Immunity to electromagnetic interference. Electromagnetic noise cannot affect fiber-optic cables.
- 4 Resistance to corrosive materials. Glass is more resistant to corrosive materials than copper.
- 5 Light weight. Fiber-optic cables are much lighter than copper cables.
- 6 Greater immunity to tapping. Fiber-optic cables are more immune to tapping than copper cables. Copper cables create antenna effects that can easily be tapped.

Disadvantages There are some disadvantages in the use of optical fiber.

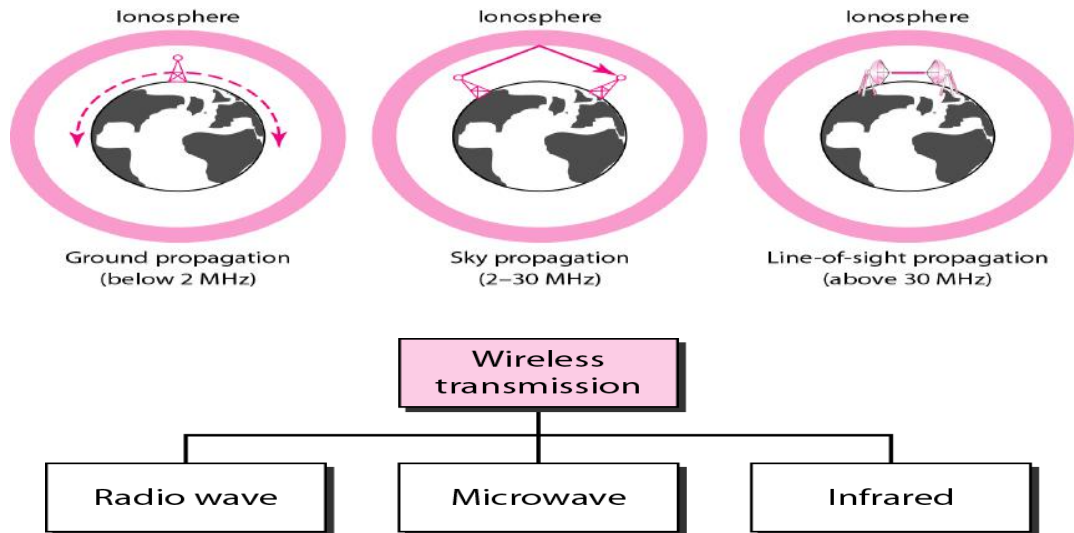
- 1 Installation and maintenance
- 2 Unidirectional light propagation. Propagation of light is unidirectional. If we need bidirectional communication, two fibers are needed.
- 3 Cost. The cable and the interfaces are relatively more expensive than those of other guided media. If the demand for bandwidth is not high, often the use of optical fiber cannot be justified.

UNGUIDED MEDIA: WIRELESS

Radio Waves
Microwaves
Infrared



Unguided signals can travel from the source to destination in several ways: ground propagation, sky propagation, and line-of-sight propagation, as shown in Figure

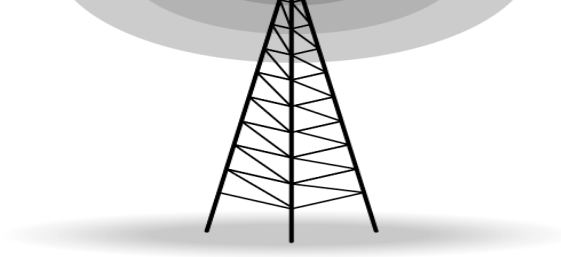


Radio Waves

Electromagnetic waves ranging in frequencies between 3 kHz and 1 GHz are normally called radio waves. Radio waves are omni directional. When an antenna transmits radio waves, they are propagated in all directions. This means that the sending and receiving antennas do not have to be aligned. A sending antenna sends waves that can be received by any receiving antenna. The omni directional property has a disadvantage, too. The radio waves transmitted by one antenna are susceptible to interference by another antenna that may send signals using the same frequency or band.

Omni directional Antenna

Radio waves use omnidirectional antennas that send out signals in all directions. Based on the wavelength, strength, and the purpose of transmission, we can have several types of antennas. Figure shows an omnidirectional antenna.



Applications

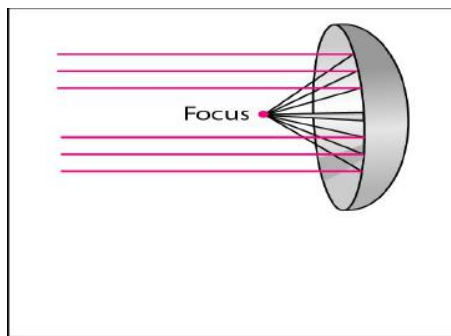
The Omni directional characteristics of radio waves make them useful for multicasting, in which there is one sender but many receivers. AM and FM radio, television, maritime radio, cordless phones, and paging are examples of multicasting.

Microwaves

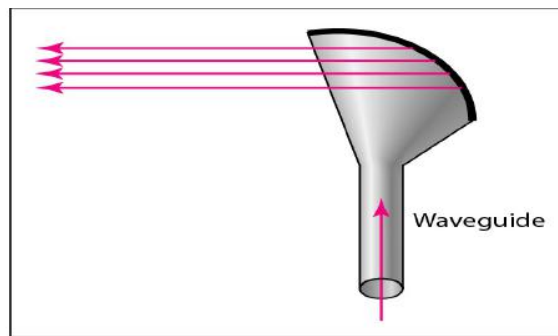
Electromagnetic waves having frequencies between 1 and 300 GHz are called microwaves. Microwaves are unidirectional. The sending and receiving antennas need to be aligned. The unidirectional property has an obvious advantage. A pair of antennas can be aligned without interfering with another pair of aligned antennas

Unidirectional Antenna

Microwaves need unidirectional antennas that send out signals in one direction. Two types of antennas are used for microwave communications: the parabolic dish and the horn



a. Dish antenna



b. Horn antenna

Applications:

Microwaves are used for unicast communication such as cellular telephones, satellite networks, and wireless LANs

Infrared

Infrared waves, with frequencies from 300 GHz to 400 THz (wavelengths from 1 mm to 770 nm), can be used for short-range communication. Infrared waves, having high frequencies, cannot penetrate walls. This advantageous

When we use our infrared remote control, we do not interfere with the use of the remote by our neighbors. Infrared signals are useless for long-range communication. In addition, we cannot use infrared waves outside a building because the sun's rays contain infrared waves that can interfere with the communication.

Applications:

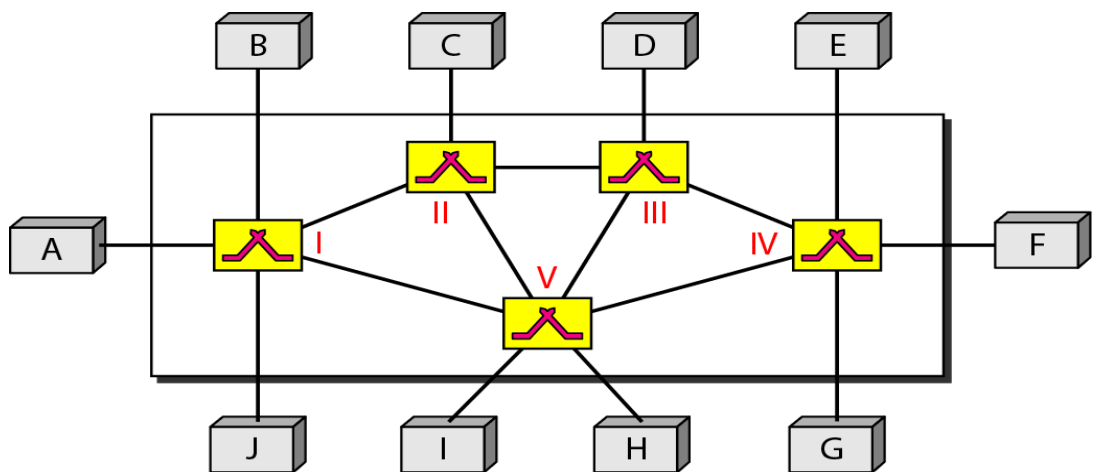
Infrared signals can be used for short-range communication in a closed area using line-of-sight propagation.

Switching

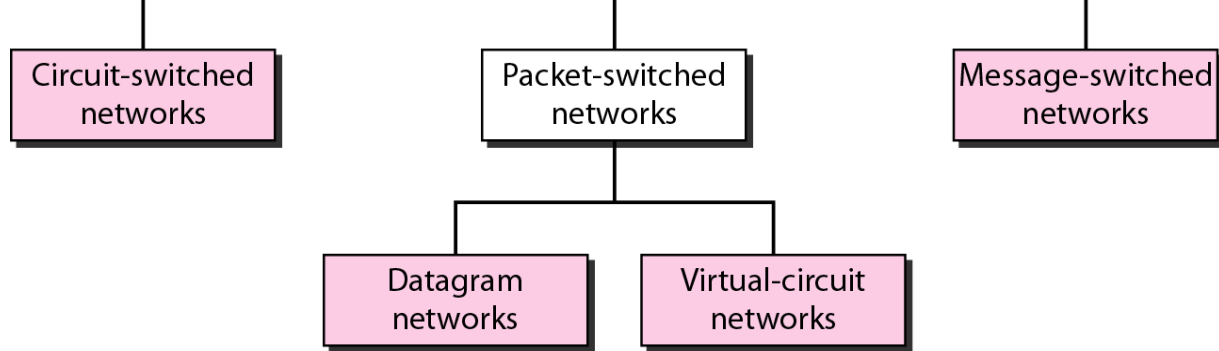
A network is a set of connected devices. Whenever we have multiple devices, we have the problem of how to connect them to make one-to-one communication possible. One solution is to make a point-to-point connection between each pair of devices (a mesh topology) or between a central device and every other device (a star topology). These methods, however, are impractical and wasteful when applied to very large networks.

The number and length of the links require too much infrastructure to be cost-efficient, and the majority of those links would be idle most of the time.

A better solution is switching. A switched network consists of a series of interlinked nodes, called switches. Switches are devices capable of creating temporary connections between two or more devices linked to the switch. In a switched network, some of these nodes are connected to the end systems (computers or telephones, for example). Others are used only for routing. Figure shows a switched network.



We can then divide today's networks into three broad categories: circuit-switched networks, packet-switched networks, and message-switched. Packet-switched networks can further be divided into two subcategories-virtual-circuit networks and datagram networks as shown in Figure.

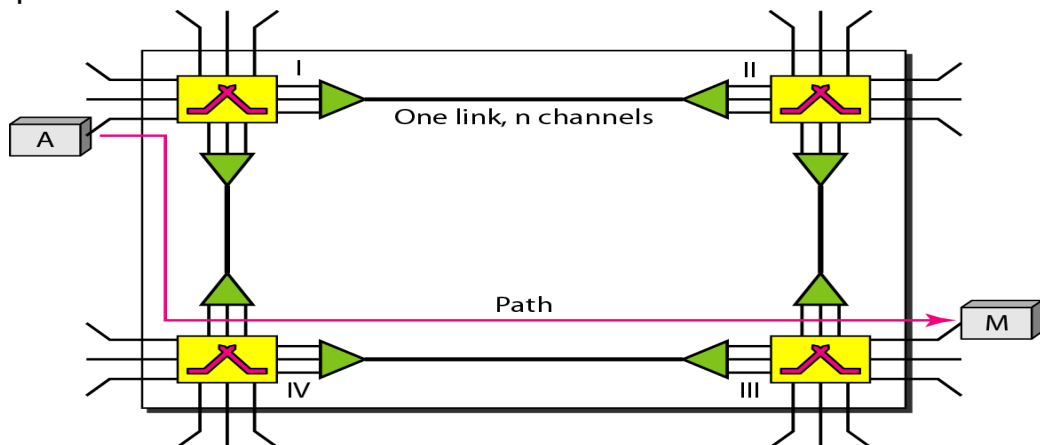


CIRCUIT-SWITCHED NETWORKS

A circuit-switched network consists of a set of switches connected by physical links. A connection between two stations is a dedicated path made of one or more links. However, each connection uses only one dedicated channel on each link. Each link is normally divided into n channels by using FDM or TDM.

In circuit switching, the resources need to be reserved during the setup phase;

the resources remain dedicated for the entire duration of data transfer until the teardown phase



Three Phases

The actual communication in a circuit-switched network requires three phases: connection setup, data transfer, and connection teardown.

Setup Phase

Before the two parties (or multiple parties in a conference call) can communicate, a dedicated circuit (combination of channels in links) needs to be established. Connection setup means creating dedicated channels between the switches. For example, in Figure, when system A needs to connect to system M, it sends a setup request that includes the address of system M, to switch I. Switch I finds a channel between itself and switch IV that can be dedicated for this purpose. Switch I then sends the request to switch IV, which finds a

M needs to be sent in the opposite direction to system A. Only after system A receives this acknowledgment is the connection established.

Data Transfer Phase

After the establishment of the dedicated circuit (channels), the two parties can transfer data.

Teardown Phase

When one of the parties needs to disconnect, a signal is sent to each switch to release the resources.

Efficiency

It can be argued that circuit-switched networks are not as efficient as the other two types of networks because resources are allocated during the entire duration of the connection. These resources are unavailable to other connections.

Delay

Although a circuit-switched network normally has low efficiency, the delay in this type of network is minimal. During data transfer the data are not delayed at each switch; the resources are allocated for the duration of the connection. The total delay is due to the time needed to create the connection, transfer data, and disconnect the circuit.

Switching at the physical layer in the traditional telephone network uses the circuit-switching approach.

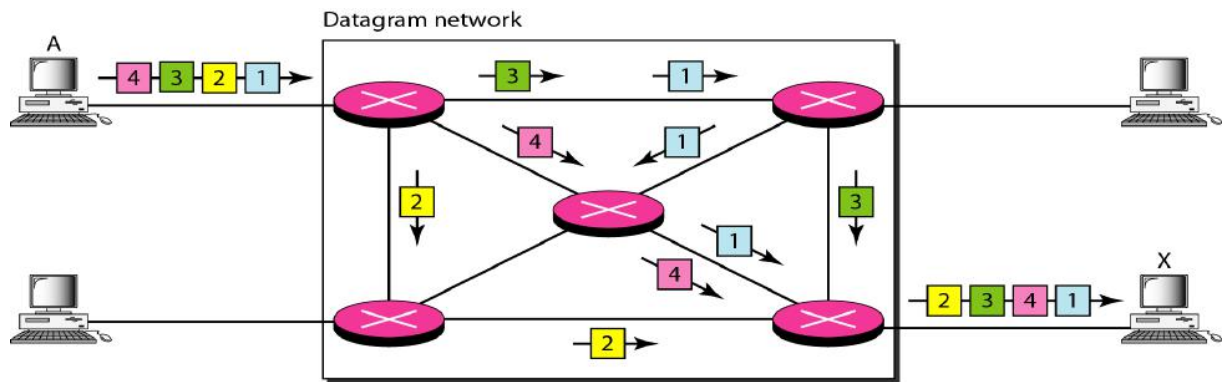
DATAGRAM NETWORKS

In a packet-switched network, there is no resource reservation; resources are allocated on demand. The allocation is done on a first come, first-served basis. When a switch receives a packet, no matter what is the source or destination, the packet must wait if there are other packets being processed. This lack of reservation may create delay. For example, if we do not have a reservation at a restaurant, we might have to wait.

In a datagram network, each packet is treated independently of all others. Packets in this approach are referred to as datagrams. Datagram switching is normally done at the network layer.

Figure shows how the datagram approach is used to deliver four packets from station A to station X. The switches in a datagram network are traditionally referred to as routers.

The datagram networks are sometimes referred to as connectionless networks. The term **connectionless** here means that the switch (packet switch) does not keep information about the connection state. There are no setup or teardown phases. Each packet is treated the same by a switch regardless of its source or destination.



Efficiency

The efficiency of a datagram network is better than that of a circuit-switched network; resources are allocated only when there are packets to be transferred.

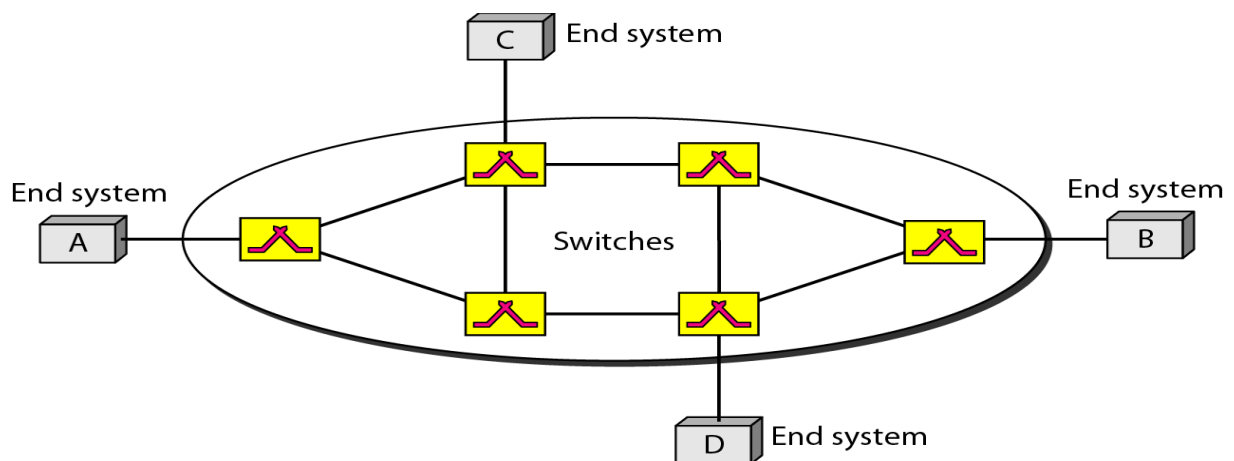
Delay

There may be greater delay in a datagram network than in a virtual-circuit network. Although there are no setup and teardown phases, each packet may experience a wait at a switch before it is forwarded. In addition, since not all packets in a message necessarily travel through the same switches, the delay is not uniform for the packets of a message.

Switching in the Internet is done by using the datagram approach to packet switching at the network layer.

VIRTUAL-CIRCUIT NETWORKS

A virtual-circuit network is a cross between a circuit-switched network and a datagram network. It has some characteristics of both.



network, or on demand, as in a datagram network.

3. As in a datagram network, data are packetized and each packet carries an address in the header. However, the address in the header has local jurisdiction (it defines what should be the next switch and the channel on which the packet is being carried), not end-to-end jurisdiction.

4. As in a circuit-switched network, all packets follow the same path established during the connection.

5. A virtual-circuit network is normally implemented in the data link layer, while a circuit-switched network is implemented in the physical layer and a datagram network in the network layer.

Addressing

In a virtual-circuit network, two types of addressing are involved: global and local (virtual-circuit identifier).

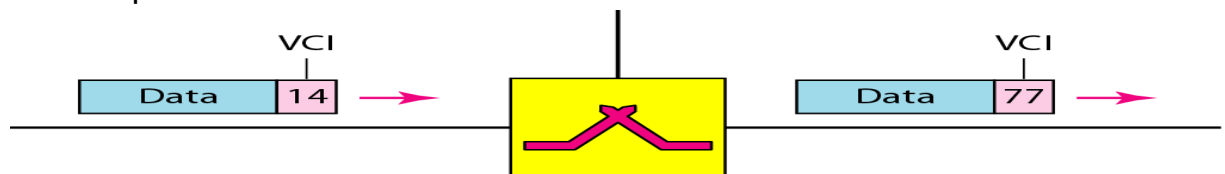
Global Addressing

A source or a destination needs to have a global address—an address that can be unique in the scope of the network.

Virtual-Circuit Identifier

The identifier that is actually used for data transfer is called the virtual-circuit identifier (VCI). A VCI, unlike a global address, is a small number that has only switch scope; it is used by a frame between two switches. When a frame arrives at a switch, it has a VCI; when it leaves, it has a different VCI.

Figure shows how the VCI in a data frame changes from one switch to another. Note that a VCI does not need to be a large number since each switch can use its own unique set of VCIs.



Three Phases

Three phases in a virtual-circuit network: setup, data transfer, and teardown.

We first discuss the data transfer phase, which is more straightforward; we then talk about the setup and teardown phases.

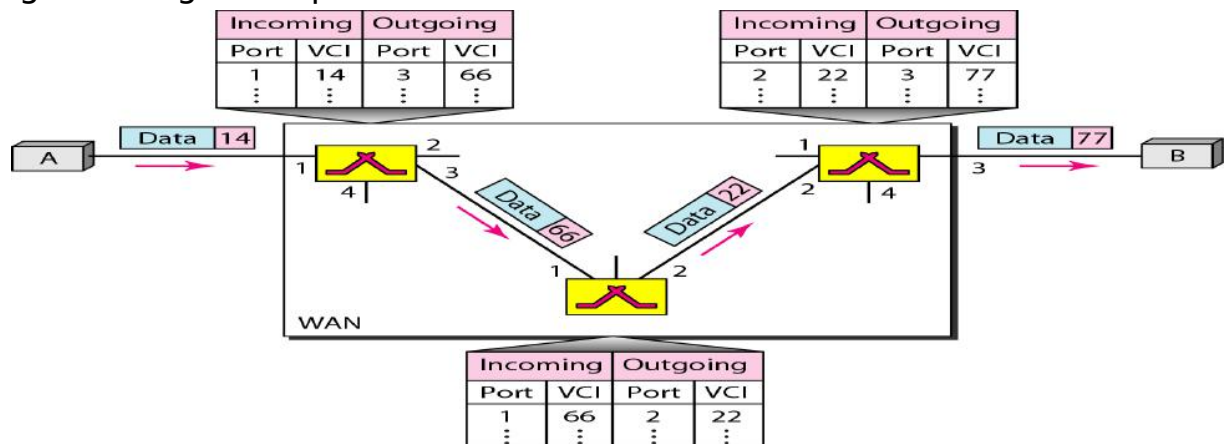
Data Transfer Phase

To transfer a frame from a source to its destination, all switches need to have a table entry for this virtual circuit. The table, in its simplest form, has four columns.

We show later how the switches make their table entries, but for the moment we assume that each switch has a table with entries for all active virtual circuits. Figure shows such a switch and its corresponding table.

Figure shows a frame arriving at port 1 with a VCI of 14. When the frame arrives, the switch looks in its table to find port 1 and a VCI of 14. When it is found, the switch knows to change the VCI to 22 and send out the frame from port 3.

Figure shows how a frame from source A reaches destination B and how its VCI changes during the trip.



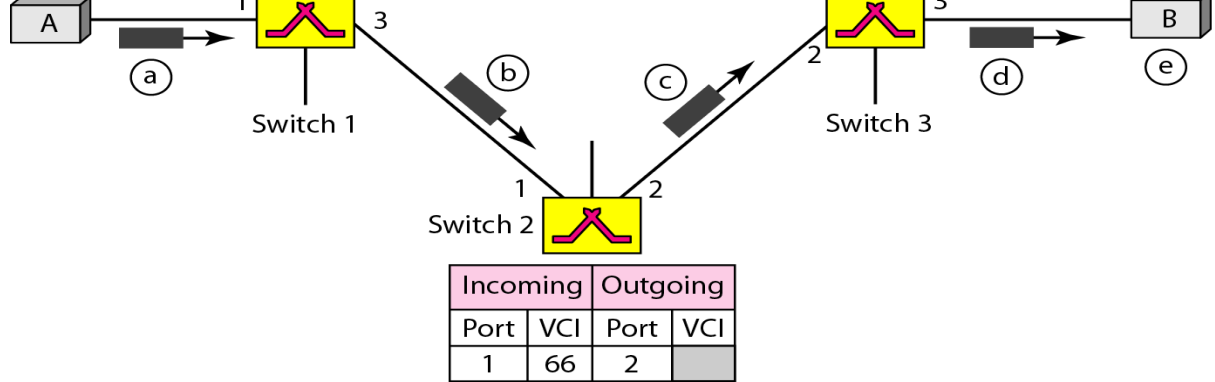
Each switch changes the VCI and routes the frame.

The data transfer phase is active until the source sends all its frames to the destination. The procedure at the switch is the same for each frame of a message. The process creates a virtual circuit, not a real circuit, between the source and destination.

Setup Phase

In the setup phase, a switch creates an entry for a virtual circuit. For example, suppose source A needs to create a virtual circuit to B. Two steps are required: the setup request and the acknowledgment.

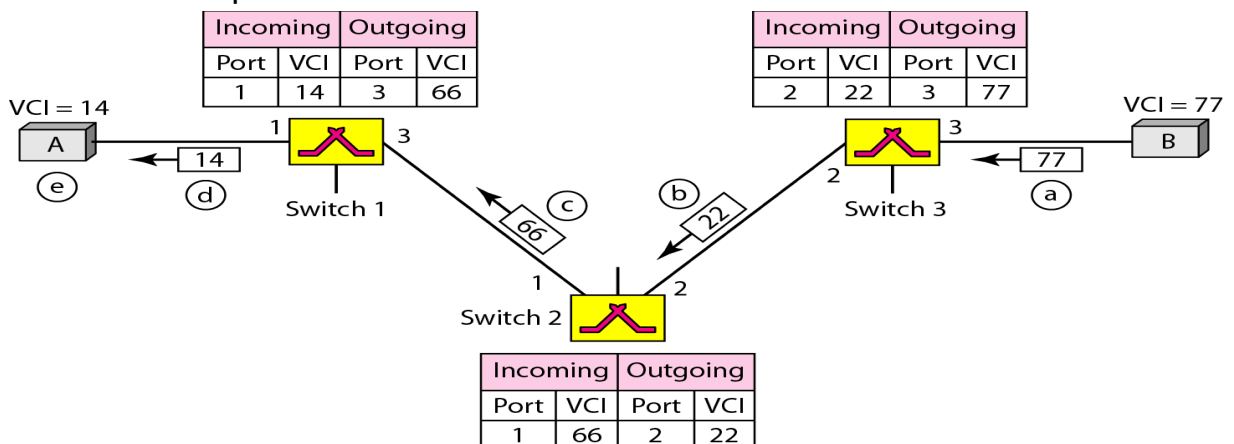
Setup Request A setup request frame is sent from the source to the destination. Figure shows the process.



- Source A sends a setup frame to switch 1.
- Switch 1 receives the setup request frame. It knows that a frame going from A to B goes out through port 3. For the moment, assume that it knows the output port. The switch creates an entry in its table for this virtual circuit, but it is only able to fill three of the four columns. The switch assigns the incoming port (1) and chooses an available incoming VCI (14) and the outgoing port (3). It does not yet know the outgoing VCI, which will be found during the acknowledgment step. The switch then forwards the frame through port 3 to switch 2.
- Switch 2 receives the setup request frame. The same events happen here as at switch 1; three columns of the table are completed: in this case, incoming port (1), incoming VCI (66), and outgoing port (2).
- Switch 3 receives the setup request frame. Again, three columns are completed: incoming port (2), incoming VCI (22), and outgoing port (3).
- Destination B receives the setup frame, and if it is ready to receive frames from A, it assigns a VCI to the incoming frames that come from A, in this case 77. This VCI lets the destination know that the frames come from A, and not other sources.

Acknowledgment A special frame, called the acknowledgment frame, completes the entries in the switching tables.

Figure shows the process.



the destination as the incoming VCI for frames from A. Switch 3 uses this VCI to complete the outgoing VCI column for this entry. Note that 77 is the incoming VCI for destination B, but the outgoing VCI for switch 3.

b. Switch 3 sends an acknowledgment to switch 2 that contains its incoming VCI in the table, chosen in the previous step. Switch 2 uses this as the outgoing VCI in the table.

c. Switch 2 sends an acknowledgment to switch 1 that contains its incoming VCI in the table, chosen in the previous step. Switch 1 uses this as the outgoing VCI in the table.

d. Finally switch 1 sends an acknowledgment to source A that contains its incoming VCI in the table, chosen in the previous step.

e. The source uses this as the outgoing VCI for the data frames to be sent to destination B.

Teardown Phase

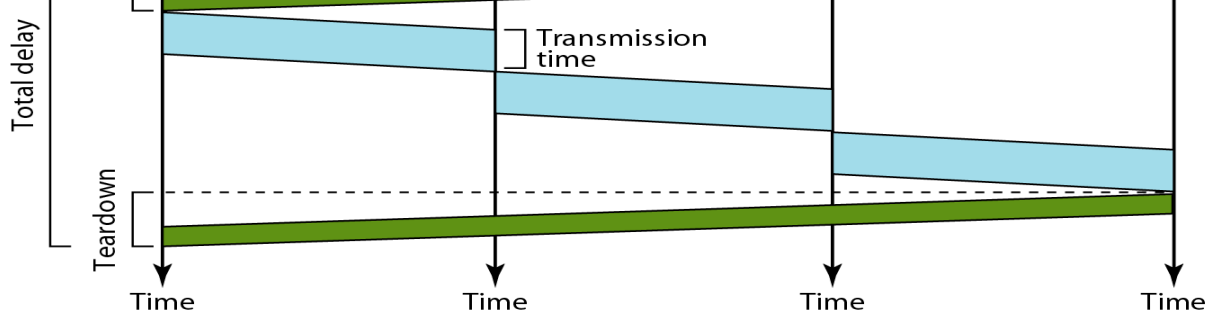
In this phase, source A, after sending all frames to B, sends a special frame called a *teardown request*. Destination B responds with a teardown confirmation frame. All switches delete the corresponding entry from their tables.

Efficiency

In virtual-circuit switching, all packets belonging to the same source and destination travel the same path; but the packets may arrive at the destination with different delays if resource allocation is on demand.

Delay

In a virtual-circuit network, there is a one-time delay for setup and a one-time delay for teardown. If resources are allocated during the setup phase, there is no wait time for individual packets. Figure shows the delay for a packet traveling through two switches in a virtual-circuit network



Switching at the data link layer in a switched WAN is normally implemented by using virtual-circuit techniques.

Comparison

Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Diagrams from Tanenbaum Textbook

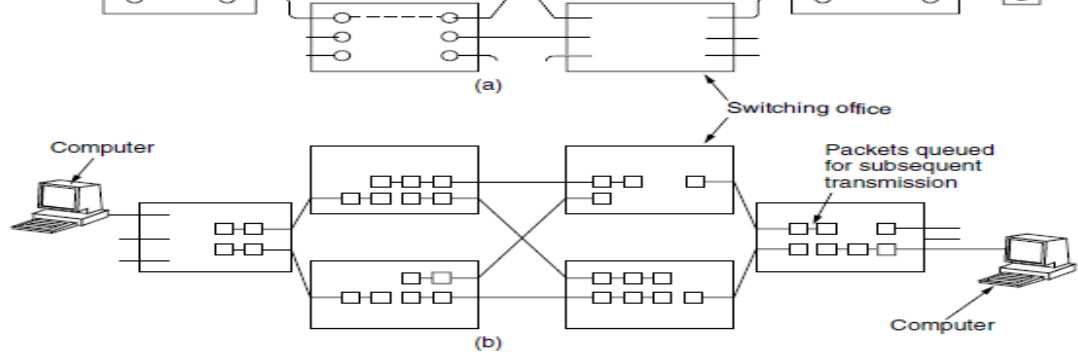


Figure 2-42. (a) Circuit switching. (b) Packet switching.

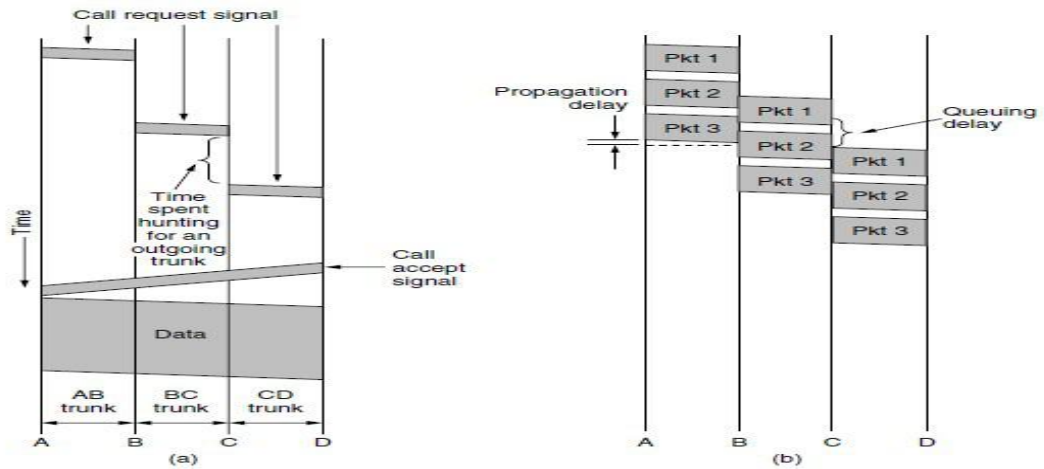
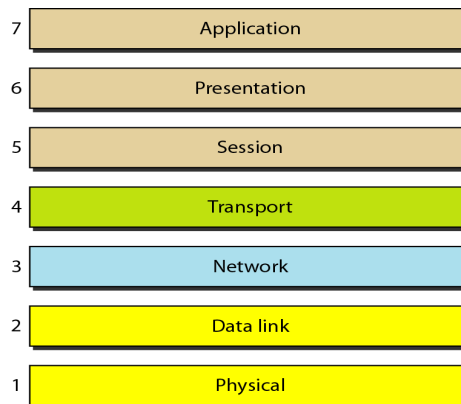
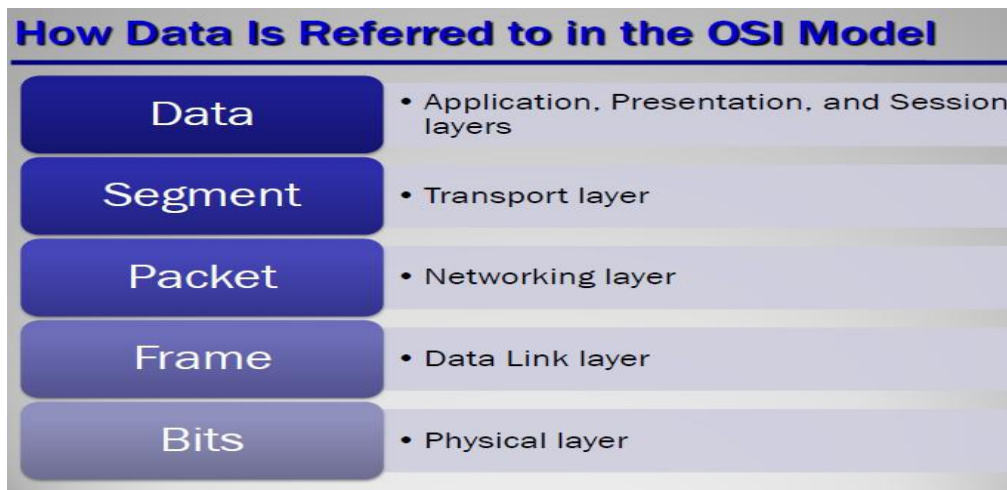


Figure 2-43. Timing of events in (a) circuit switching, (b) packet switching.

OSI

- OSI stands for Open Systems Interconnection
- Created by International Standards Organization (ISO)
- Was created as a framework and reference model to explain how different networking technologies work together and interact
- It is not a standard that networking protocols must follow
- Each layer has specific functions it is responsible for
- All layers work together in the correct order to move data around a network





Physical Layer

- Deals with all aspects of physically moving data from one computer to the next
- Converts data from the upper layers into 1s and 0s for transmission over media
- Defines how data is encoded onto the media to transmit the data
- Defined on this layer: Cable standards, wireless standards, and fiber optic standards.

Copper wiring, fiber optic cable, radio frequencies, anything that can be used to transmit data is defined on the Physical layer of the OSI Model

- Device example: Hub
- Used to transmit data

Data Link Layer

- Is responsible for moving frames from node to node or computer to computer
- Can move frames from one adjacent computer to another, cannot move frames across routers
- Encapsulation = frame
- Requires MAC address or *physical address*
- Protocols defined include Ethernet Protocol and Point-to-Point Protocol (PPP)
- Device example: Switch
- Two sublayers: Logical Link Control (LLC) and the Media Access Control (MAC)
 - Logical Link Control (LLC)
 - -Data Link layer addressing, flow control, address notification, error control
 - Media Access Control (MAC)
 - -Determines which computer has access to the network media at any given time
 - -Determines where one frame ends and the next one starts, called frame synchronization

Network Layer

- Device example: Router
- -Routing is the ability of various network devices and their related software to move data packets from source to destination

Transport Layer

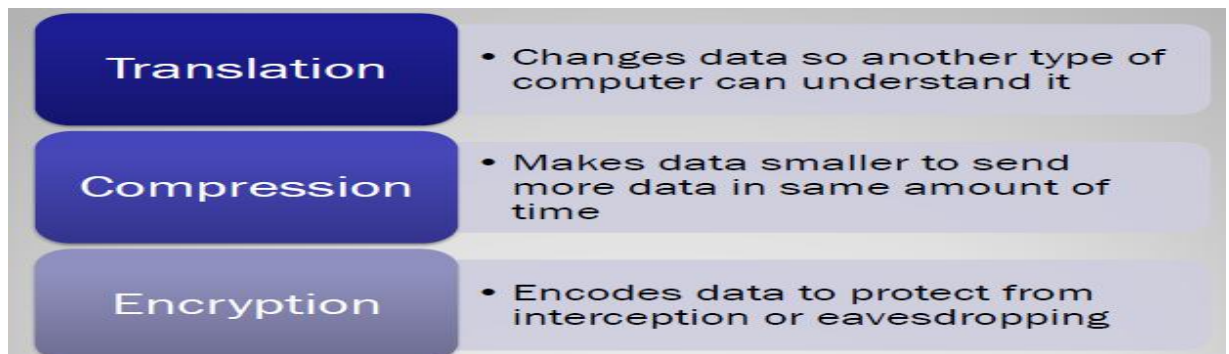
- Takes data from higher levels of OSI Model and breaks it into segments that can be sent to lower-level layers for data transmission
- Conversely, reassembles data segments into data that higher-level protocols and applications can use
- Also puts segments in correct order (called sequencing) so they can be reassembled in correct order at destination
- Concerned with the reliability of the transport of sent data
- May use a *connection-oriented protocol* such as TCP to ensure destination received segments
- May use a *connectionless protocol* such as UDP to send segments without assurance of delivery
- Uses port addressing

Session Layer

- Responsible for managing the dialog between networked devices
- Establishes, manages, and terminates connections
- Provides duplex, half-duplex, or simplex communications between devices
- Provides procedures for establishing checkpoints, adjournment, termination, and restart or recovery procedures

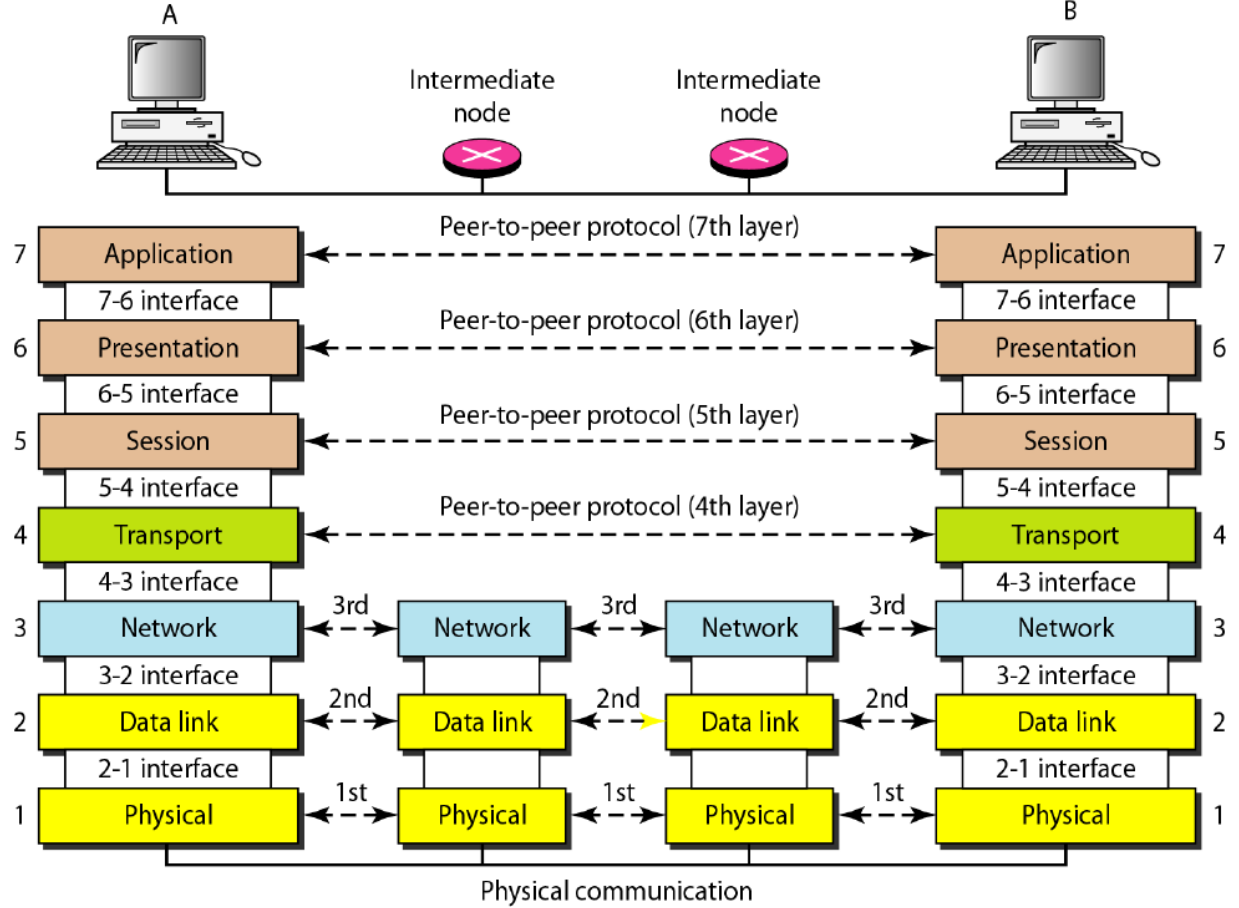
Presentation Layer

- Concerned with how data is presented to the network
- Handles three primary tasks: -Translation , -Compression , -Encryption

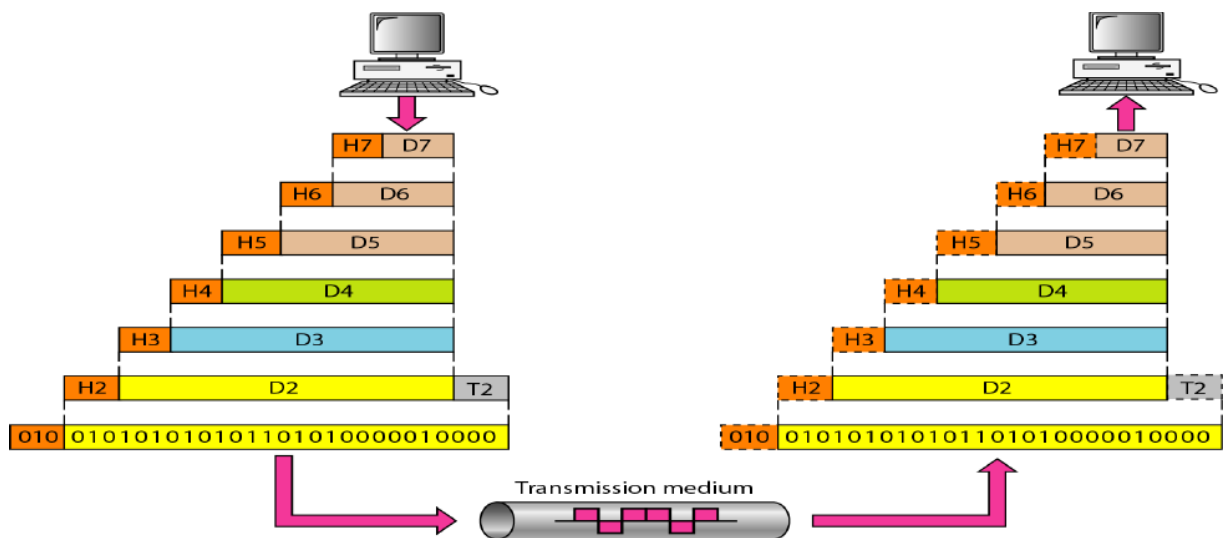


Application Layer

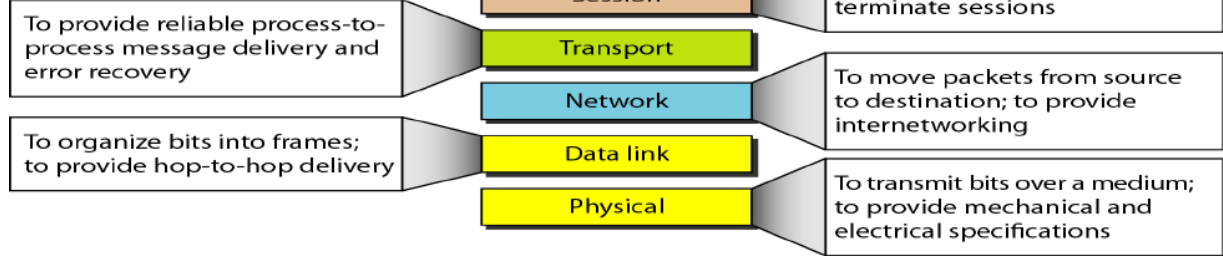
- Contains all services or protocols needed by application software or operating system to communicate on the network
- Examples
 - -Firefox web browser uses HTTP (Hyper-Text Transport Protocol)
 - -E-mail program may use POP3 (Post Office Protocol version 3) to read e-mails and SMTP (Simple Mail Transport Protocol) to send e-mails



An exchange using the OSI model

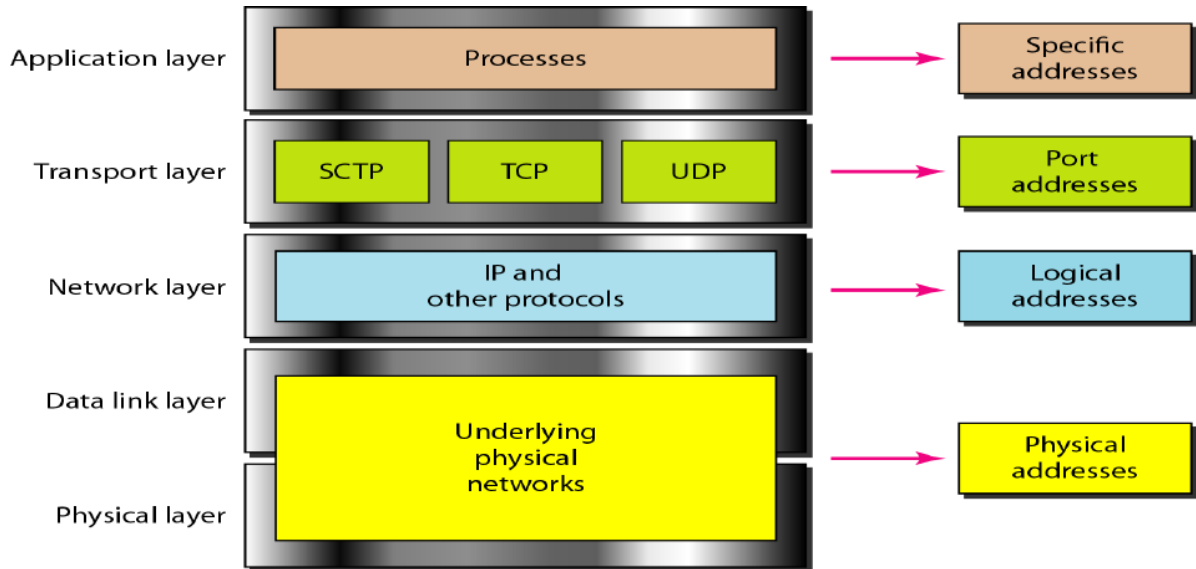


SUMMARY:



TCP/IP Model (Transmission Control Protocol/Internet Protocol)

-A *protocol suite* is a large number of related protocols that work together to allow networked computers to communicate



Relationship of layers and addresses in TCP/IP

Application Layer

- Application layer protocols define the rules when implementing specific network applications
- Rely on the underlying layers to provide accurate and efficient data delivery
- Typical protocols:
 - FTP – File Transfer Protocol
 - For file transfer
 - Telnet – Remote terminal protocol
 - For remote login on any other computer on the network
 - SMTP – Simple Mail Transfer Protocol
 - For mail transfer
 - HTTP – Hypertext Transfer Protocol
 - For Web browsing
- Encompasses same functions as these OSI Model layers Application Presentation Session

Transport Layer

TCP & UDP

Virtual circuit

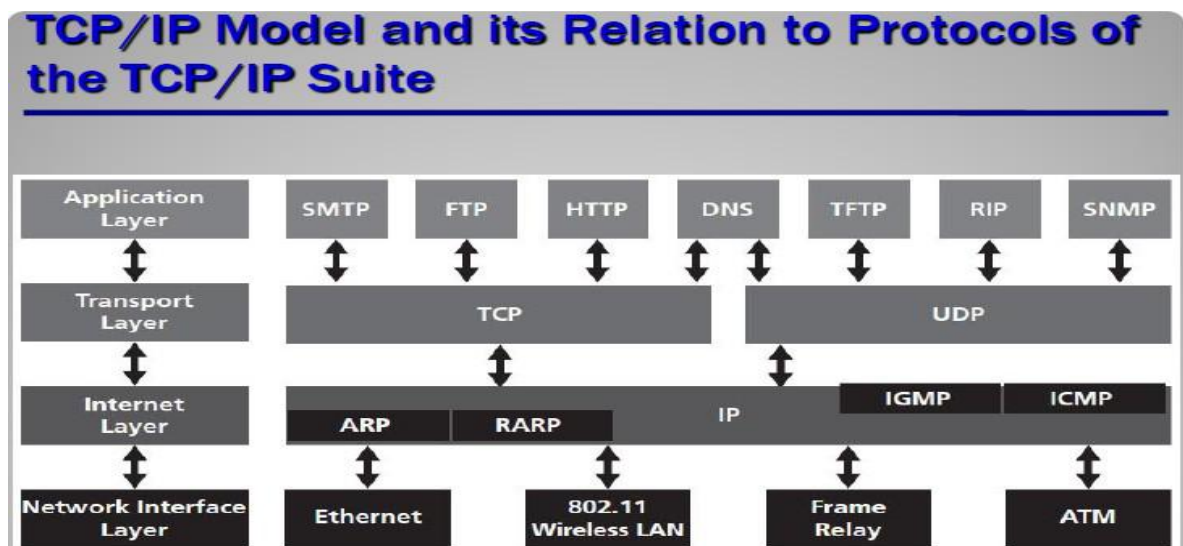
- UDP provides the functions:
 - Dividing a chunk of data into segments
 - Reassembly segments into the original chunk
 - Provide further the functions such as reordering and data resend
- Offering a reliable byte-stream delivery service
- Functions the same as the Transport layer in OSI
- Synchronize source and destination computers to set up the session between the respective computers

Internet Layer

- The network layer, also called the internet layer, deals with packets and connects independent networks to transport the packets across network boundaries. The network layer protocols are the IP and the Internet Control Message Protocol ([ICMP](#)), which is used for error reporting.

Host-to-network layer

The **Host-to-network layer** is the lowest **layer** of the **TCP/IP** reference model. It combines the link **layer** and the physical **layer** of the ISO/OSI model. At this **layer**, data is transferred between adjacent **network** nodes in a WAN or between nodes on the same LAN.



Uses strict layering resulting in vertical layers.	horizontal layers.
Supports both connectionless & connection-oriented communication in the Network layer, but only connection-oriented communication in Transport Layer	Supports only connectionless communication in the Network layer, but both connectionless & connection-oriented communication in Transport Layer
It distinguishes between Service, Interface and Protocol.	Does not clearly distinguish between Service, Interface and Protocol.
Protocols are better hidden and can be replaced relatively easily as technology changes (No transparency)	Protocols are not hidden and thus cannot be replaced easily. (Transparency) Replacing IP by a substantially different protocol would be virtually impossible
OSI reference model was devised before the corresponding protocols were designed.	The protocols came first and the model was a description of the existing protocols

THE INTERNET

The Internet has revolutionized many aspects of our daily lives. It has affected the way we do business as well as the way we spend our leisure time. Count the ways you've used the Internet recently. Perhaps you've sent electronic mail (e-mail) to a business associate, paid a utility bill, read a newspaper from a distant city, or looked up a local movie schedule-all by using the Internet. Or maybe you researched a medical topic, booked a hotel reservation, chatted with a fellow Trekkie, or comparison-shopped for a car. The Internet is a communication system that has brought a wealth of information to our fingertips and organized it for our use.

A Brief History

A network is a group of connected communicating devices such as computers and printers. An internet (note the lowercase letter i) is two or more networks that can communicate with each other. The most notable internet is called the Internet (uppercase letter I), a collaboration of more than hundreds of thousands of interconnected networks. Private individuals as well as various organizations such as government agencies, schools, research facilities, corporations, and libraries in more than 100 countries use the Internet. Millions of people are users. Yet this extraordinary communication system only came into being in 1969.

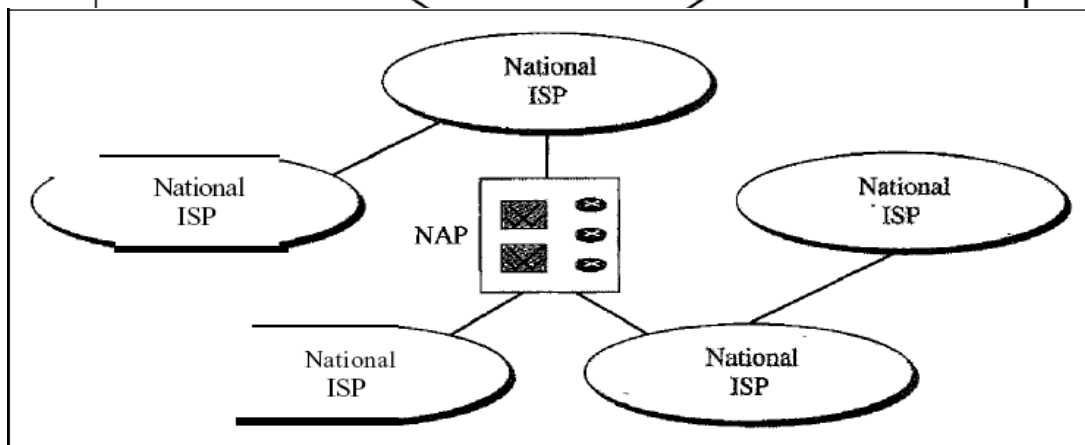
In the mid-1960s, mainframe computers in research organizations were standalone devices. Computers from different manufacturers were unable to communicate with one another. The Advanced Research Projects Agency

In 1967, at an Association for Computing Machinery (ACM) meeting, ARPA presented its ideas for ARPANET, a small network of connected computers. The idea was that each host computer (not necessarily from the same manufacturer) would be attached to a specialized computer, called an *interface message processor* (IMP). The IMPs, in turn, would be connected to one another. Each IMP had to be able to communicate with other IMPs as well as with its own attached host. By 1969, ARPANET was a reality. Four nodes, at the University of California at Los Angeles (UCLA), the University of California at Santa Barbara (UCSB), Stanford Research Institute (SRI), and the University of Utah, were connected via the IMPs to form a network. Software called the *Network Control Protocol* (NCP) provided communication between the hosts.

In 1972, Vint Cerf and Bob Kahn, both of whom were part of the core ARPANET group, collaborated on what they called the *Internetting Project*. Cerf and Kahn's landmark 1973 paper outlined the protocols to achieve end-to-end delivery of packets. This paper on Transmission Control Protocol (TCP) included concepts such as encapsulation, the datagram, and the functions of a gateway. Shortly thereafter, authorities made a decision to split TCP into two protocols: Transmission Control Protocol (TCP) and Internetworking Protocol (IP). IP would handle datagram routing while TCP would be responsible for higher-level functions such as segmentation, reassembly, and error detection. The internetworking protocol became known as TCP/IP.

The Internet Today

The Internet has come a long way since the 1960s. The Internet today is not a simple hierarchical structure. It is made up of many wide- and local-area networks joined by connecting devices and switching stations. It is difficult to give an accurate representation of the Internet because it is continually changing—new networks are being added, existing networks are adding addresses, and networks of defunct companies are being removed. Today most end users who want Internet connection use the services of Internet service providers (ISPs). There are international service providers, national service providers, regional service providers, and local service providers. The Internet today is run by private companies, not the government. Figure 1.13 shows a conceptual (not geographic) view of the Internet.



b. Interconnection of national ISPs

International Internet Service Providers:

At the top of the hierarchy are the international service providers that connect nations together.

National Internet Service Providers:

The national Internet service providers are backbone networks created and maintained by specialized companies. There are many national ISPs operating in North America; some of the most well known are SprintLink, PSINet, UUNet Technology, AGIS, and internet Mel. To provide connectivity between the end users, these backbone networks are connected by complex switching stations (normally run by a third party) called network access points (NAPs). Some national ISP networks are also connected to one another by private switching stations called *peering points*. These normally operate at a high data rate (up to 600 Mbps).

Regional Internet Service Providers:

Regional internet service providers or regional ISPs are smaller ISPs that are connected to one or more national ISPs. They are at the third level of the hierarchy with a smaller data rate. ***Local Internet Service Providers:***

Local Internet service providers provide direct service to the end users. The local ISPs can be connected to regional ISPs or directly to national ISPs. Most end users are connected to the local ISPs. Note that in this sense, a local ISP can be a company that just provides Internet services, a corporation with a network that supplies services to its own employees, or a nonprofit organization, such as a college or a university, that runs its own network. Each of these local ISPs can be connected to a regional or national service provider.

1. **Providing services to the network layer:**

1. Unacknowledged connectionless service.

Appropriate for low error rate and real-time traffic. Ex: Ethernet

2. Acknowledged connectionless service.

Useful in unreliable channels, WiFi. Ack/Timer/Resend

3. Acknowledged connection-oriented service.

Guarantee frames are received exactly once and in the right order. Appropriate over long, unreliable links such as a satellite channel or a long-distance telephone circuit

2. **Framing:** Frames are the streams of bits received from the network layer into manageable data units. This division of stream of bits is done by Data Link Layer.
3. **Physical Addressing:** The Data Link layer adds a header to the frame in order to define physical address of the sender or receiver of the frame, if the frames are to be distributed to different systems on the network.
4. **Flow Control:** A receiving node can receive the frames at a faster rate than it can process the frame. Without flow control, the receiver's buffer can overflow, and frames can get lost. To overcome this problem, the data link layer uses the flow control to prevent the sending node on one side of the link from overwhelming the receiving node on another side of the link. This prevents traffic jam at the receiver side.
5. **Error Control:** Error control is achieved by adding a trailer at the end of the frame. Duplication of frames are also prevented by using this mechanism. Data Link Layers adds mechanism to prevent duplication of frames.

Error detection: Errors can be introduced by signal attenuation and noise. Data Link Layer protocol provides a mechanism to detect one or more errors. This is achieved by adding error detection bits in the frame and then receiving node can perform an error check.

Error correction: Error correction is similar to the Error detection, except that receiving node not only detects the errors but also determine where the errors have occurred in the frame.

6. **Access Control:** Protocols of this layer determine which of the devices has control over the link at any given time, when two or more devices are connected to the same link.
7. **Reliable delivery:** Data Link Layer provides a reliable delivery service, i.e., transmits the network layer datagram without any error. A reliable delivery service is accomplished with transmissions and acknowledgements. A data link layer mainly provides the reliable delivery

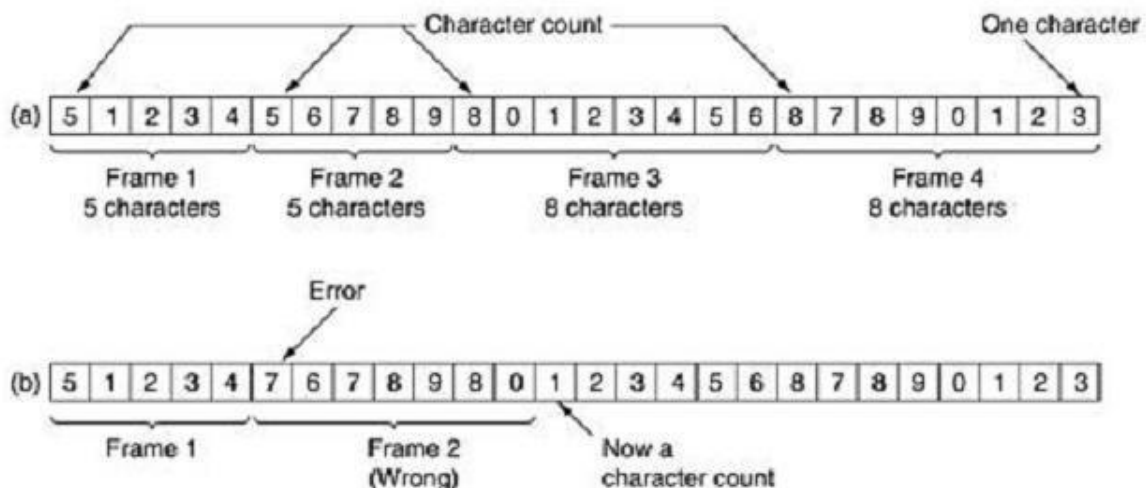
8. **Half-Duplex & Full-Duplex:** In a Full-Duplex mode, both the nodes can transmit the data at the same time. In a Half-Duplex mode, only one node can transmit the data at the same time.

FRAMING:

To provide service to the network layer, the data link layer must use the service provided to it by the physical layer. What the physical layer does is accept a raw bit stream and attempt to deliver it to the destination. This bit stream is not guaranteed to be error free. The number of bits received may be less than, equal to, or more than the number of bits transmitted, and they may have different values. It is up to the data link layer to **detect and, if necessary, correct errors**. The usual approach is for the data link layer to break the bit stream up into discrete frames and compute the checksum for each frame (**framing**). When a frame arrives at the destination, the checksum is recomputed. If the newly computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it (e.g., discarding the bad frame and possibly also sending back an error report). We will look at four framing methods:

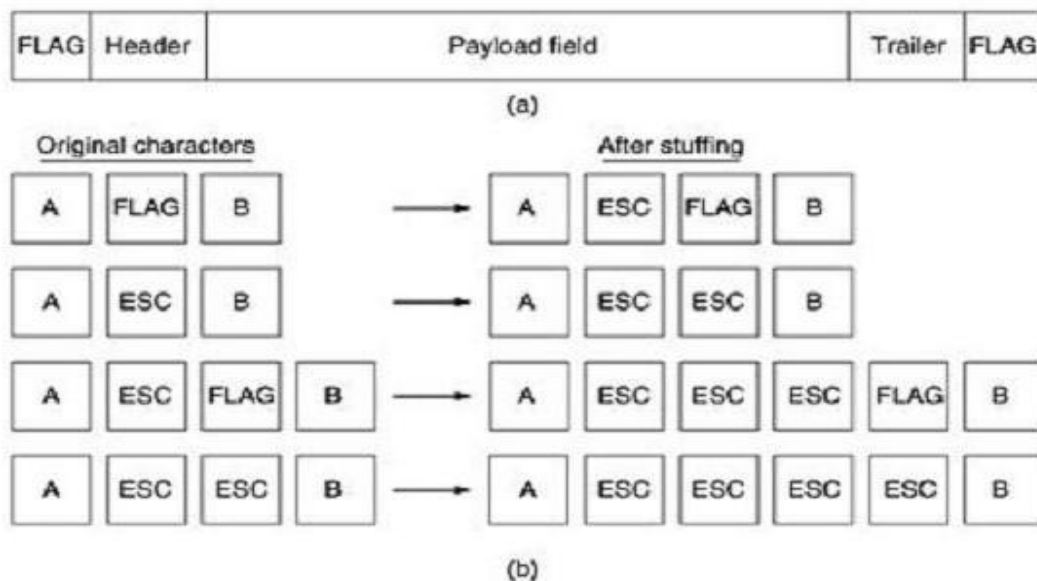
1. Character count.
2. Flag bytes with byte stuffing.
3. Starting and ending flags, with bit stuffing.
4. Physical layer coding violations.

Character count 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. This technique is shown in Fig. (a) For four frames of sizes 5, 5, 8, and 8 characters, respectively.



frame of Fig. (b) becomes a 7, the destination will get out of synchronization and will be unable to locate the start of the next frame. Even if the checksum is incorrect so the destination knows that the frame is bad, it still has no way of telling where the next frame starts. Sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many characters to skip over to get to the start of the retransmission. For this reason, the character count method is rarely used anymore.

Flag bytes with byte stuffing method gets around the problem of resynchronization after an error by having each frame start and end with special bytes. In the past, the starting and ending bytes were different, but in recent years most protocols have used the same byte, called a flag byte, as both the starting and ending delimiter, as shown in Fig. (a) as FLAG. In this way, if the receiver ever loses synchronization, it can just search for the flag byte to find the end of the current frame. Two consecutive flag bytes indicate the end of one frame and start of the next one.



(a) A frame delimited by flag bytes (b) Four examples of byte sequences before and after byte stuffing

It may easily happen that the flag byte's bit pattern occurs in the data. This situation will usually interfere with the framing. One way to solve this problem is to have the sender's data link layer insert a special escape byte (ESC) just before each "accidental" flag byte in the data. The data link layer on the receiving end removes the escape byte before the data are given to the network layer. This technique is called byte stuffing or character stuffing.

answer is that, it too is stuffed with an escape byte. Thus, any single escape byte is part of an escape sequence, whereas a doubled one indicates that a single escape occurred naturally in the data. Some examples are shown in Fig. (b). In all cases, the byte sequence delivered after de stuffing is exactly the same as the original byte sequence.

A major disadvantage of using this framing method is that it is closely tied to the use of 8-bit characters. Not all character codes use 8-bit characters. For example UNICODE uses 16-bit characters, so a new technique had to be developed to allow arbitrary sized characters

Starting and ending flags, with bit stuffing allows data frames to contain an arbitrary number of bits and allows character codes with an arbitrary number of bits per character. It works like this. Each frame begins and ends with a special bit pattern, 01111110 (in fact, a flag byte). Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream. This bit stuffing is analogous to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data.

When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically de- stuffs (i.e., deletes) the 0 bit. Just as byte stuffing is completely transparent to the network layer in both computers, so is bit stuffing. If the user data contain the flag pattern, 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110.

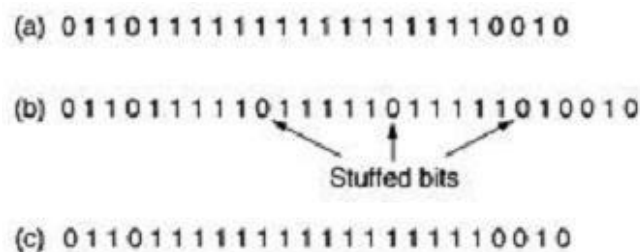
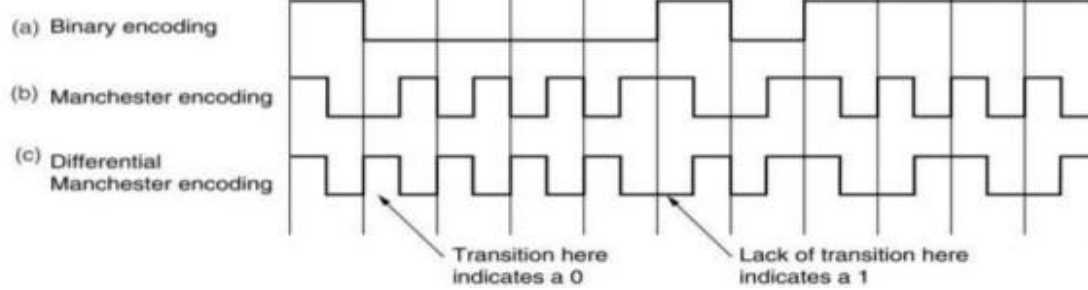


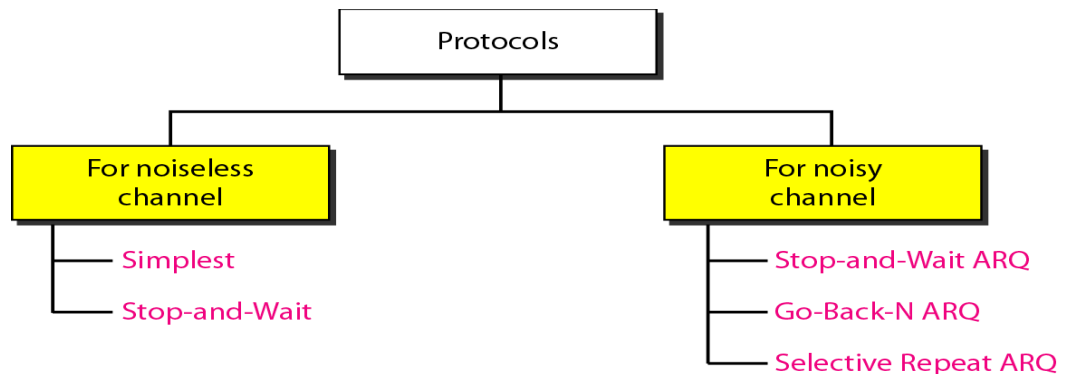
Fig:Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing.

With bit stuffing, the boundary between two frames can be unambiguously recognized by the flag pattern. Thus, if the receiver loses track of where it is, all it has to do is scan the input for flag sequences, since they can only occur at frame boundaries and never within the data.

Physical layer coding violations method of framing is only applicable to networks in which the encoding on the physical medium contains some redundancy. For example, some LANs encode 1 bit of data by using 2 physical bits. Normally, a 1 bit is a high-low pair and a 0 bit is a low-high pair. The scheme means that every data bit has a transition in the middle, making it easy for the receiver to locate the bit boundaries. The combinations high-



As a final note on framing, many data link protocols use combination of a character count with one of the other methods for extra safety. When a frame arrives, the count field is used to locate the end of the frame. Only if the appropriate delimiter is present at that position and the checksum is correct is the frame accepted as valid. Otherwise, the input stream is scanned for the next delimiter



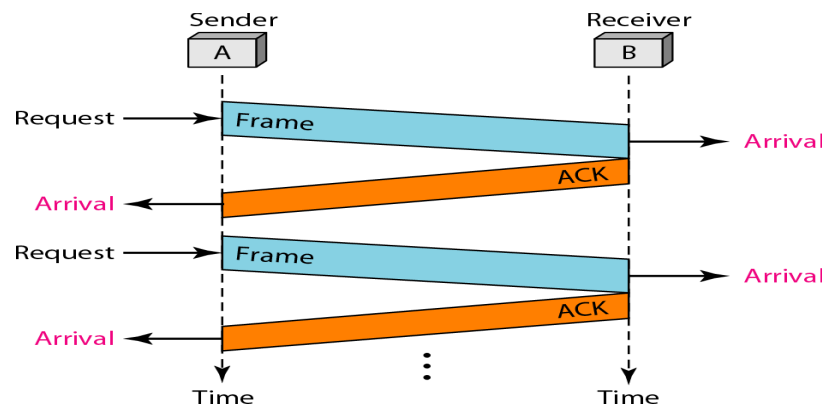
ELEMENTARY DATA LINK PROTOCOLS

Simplest Protocol



It is very simple. The sender sends a sequence of frames without even thinking about the receiver. Data are transmitted in one direction only. Both sender & receiver always ready. Processing time can be ignored. Infinite buffer space is available. And best of all, the communication channel between the data link layers never damages or loses frames. This thoroughly unrealistic protocol, which we will nickname “Utopia,”. The utopia protocol is unrealistic because **it does not handle either flow control or error correction**

Stop-and-wait Protocol



It is still very simple. The sender sends one frame and waits for feedback from the receiver. When the ACK arrives, the sender sends the next frame

It is Stop-and-Wait Protocol because the sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame. We still have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens of acknowledgment) travel from the other direction. We add flow control to our previous protocol.

NOISY CHANNELS

Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are nonexistent. We can ignore the error (as we sometimes do), or we need to add error control to our protocols. We discuss three protocols in this section that use error control.

Sliding Window Protocols:

3 Selective Repeat Automatic Repeat Request

1 Stop-and-Wait Automatic Repeat Request

To detect and correct corrupted frames, we need to add redundancy bits to our data frame. When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded. The detection of errors in this protocol is manifested by the silence of the receiver.

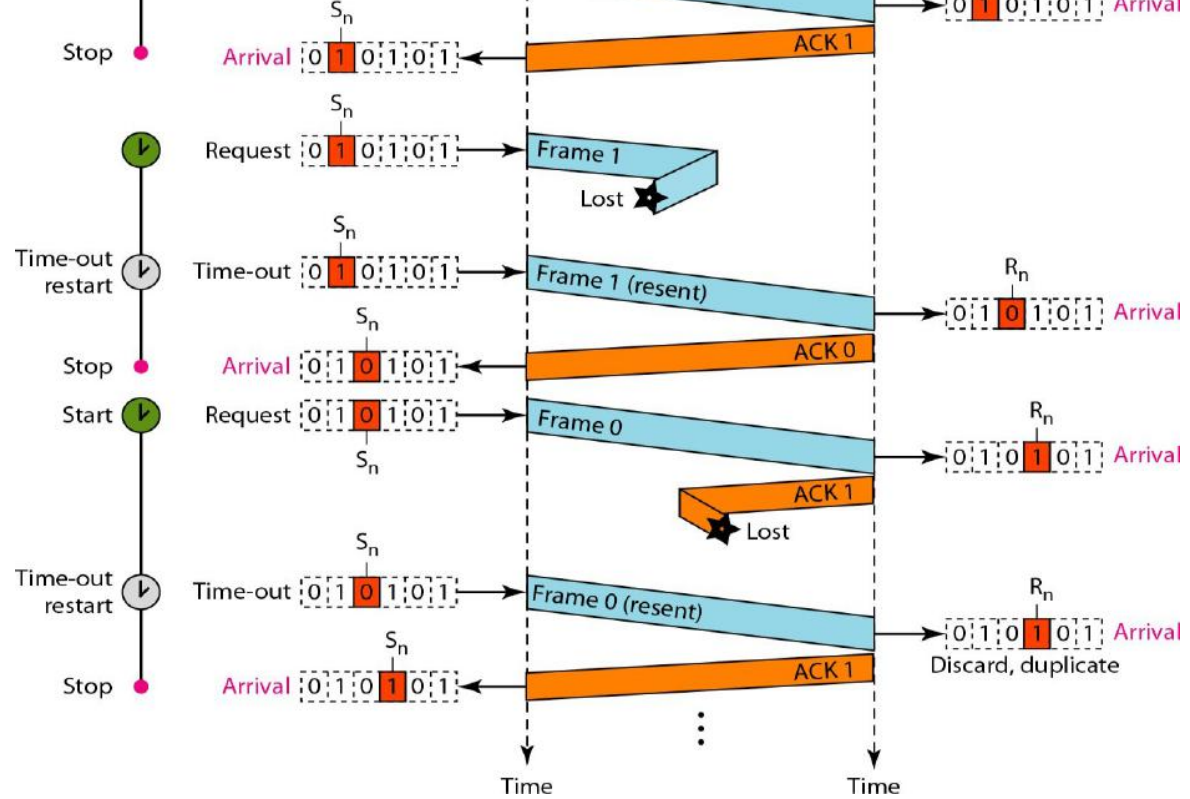
Lost frames are more difficult to handle than corrupted ones. In our previous protocols, there was no way to identify a frame. The received frame could be the correct one, or a duplicate, or a frame out of order. The solution is to number the frames. When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated

The lost frames need to be resent in this protocol. If the receiver does not respond when there is an error, how can the sender know which frame to resend? To remedy this problem, the sender keeps a copy of the sent frame. At the same time, it starts a timer. If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted. Since the protocol uses the stop-and-wait mechanism, there is only one specific frame that needs an ACK

Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires

In Stop-and-Wait ARQ, we use sequence numbers to number the frames. The sequence numbers are based on modulo-2 arithmetic.

In Stop-and-Wait ARQ, the acknowledgment number always announces in modulo-2 arithmetic the sequence number of the next frame expected.



Bandwidth Delay Product:

Assume that, 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?

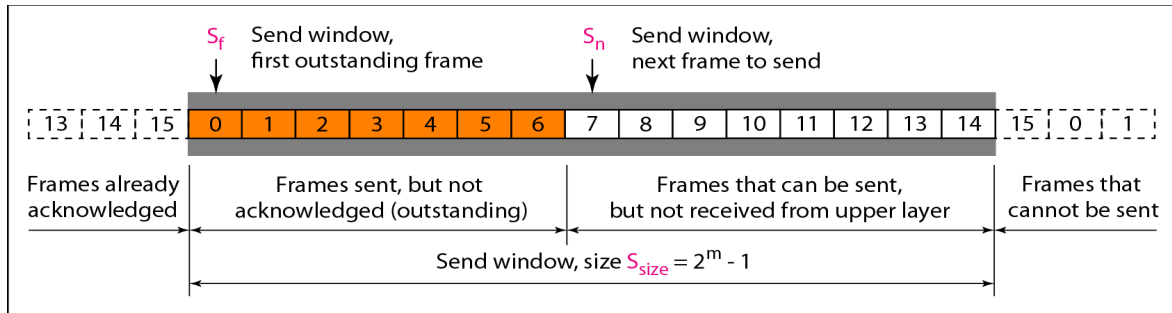
$$(1 \times 10^6) \times (20 \times 10^{-3}) = 20,000 \text{ bits}$$

The link utilization is only 1000/20,000, or 5 percent. For this reason, for a link with a high bandwidth or long delay, the use of Stop-and-Wait ARQ wastes the capacity of the link.

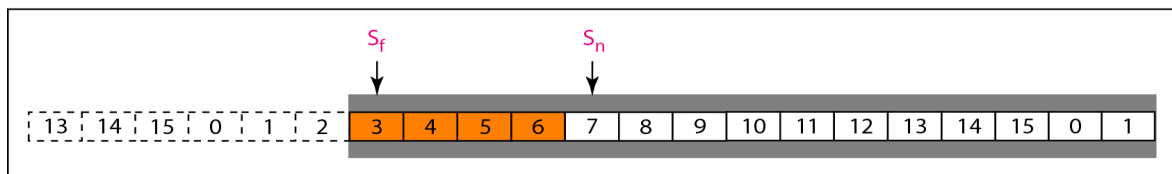
2. Go-Back-N Automatic Repeat Request

To improve the efficiency of transmission (filling the pipe), multiple frames must be in transition while waiting for acknowledgment. In other words, we need to let more than one frame be outstanding to keep the channel busy while the sender is waiting for acknowledgment.

In the Go-Back-N Protocol, the sequence numbers are modulo 2^m , where m is the size of the sequence number field in bits. The sequence numbers range from 0 to $2^m - 1$. For example, if m is 4, the only sequence numbers are 0 through 15 inclusive.



a. Send window before sliding



b. Send window after sliding

The **sender window** at any time divides the possible sequence numbers into four regions.

The first region, from the far left to the left wall of the window, defines the sequence numbers belonging to frames that are already acknowledged. The sender does not worry about these frames and keeps no copies of them.

The second region, colored in Figure (a), defines the range of sequence numbers belonging to the frames that are sent and have an unknown status. The sender needs to wait to find out if these frames have been received or were lost. We call these outstanding frames.

The third range, white in the figure, defines the range of sequence numbers for frames that can be sent; however, the corresponding data packets have not yet been received from the network layer.

Finally, the fourth region defines sequence numbers that cannot be used until the window slides

The send window is an abstract concept defining an imaginary box of size $2^m - 1$ with three variables: S_f , S_n , and S_{size} . The variable S_f defines the sequence number of the first (oldest) outstanding frame. The variable S_n holds the sequence number that will be assigned to the next frame to be sent. Finally, the variable S_{size} defines the size of the window.

Figure (b) shows how a send window can slide one or more slots to the right when an acknowledgment arrives from the other end. The acknowledgments in this protocol are cumulative, meaning that more than one frame can be acknowledged by an ACK frame. In Figure, frames 0, 1, and 2 are

arrives.

Receiver window: variable R_n (receive window, next frame expected) .

The sequence numbers to the left of the window belong to the frames already received and acknowledged; the sequence numbers to the right of this window define the frames that cannot be received. Any received frame with a sequence number in these two regions is discarded. Only a frame with a sequence number matching the value of R_n is accepted and acknowledged. The receive window also slides, but only one slot at a time. When a correct frame is received (and a frame is received only one at a time), the window slides.(see below figure for receiving window)

The receive window is an abstract concept defining an imaginary box of size 1 with one single variable R_n . The window slides when a correct frame has arrived; sliding occurs one slot at a time

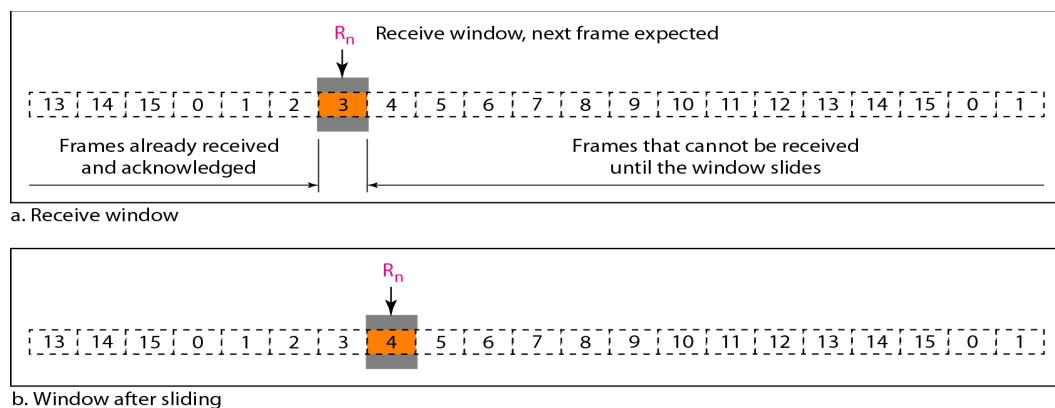


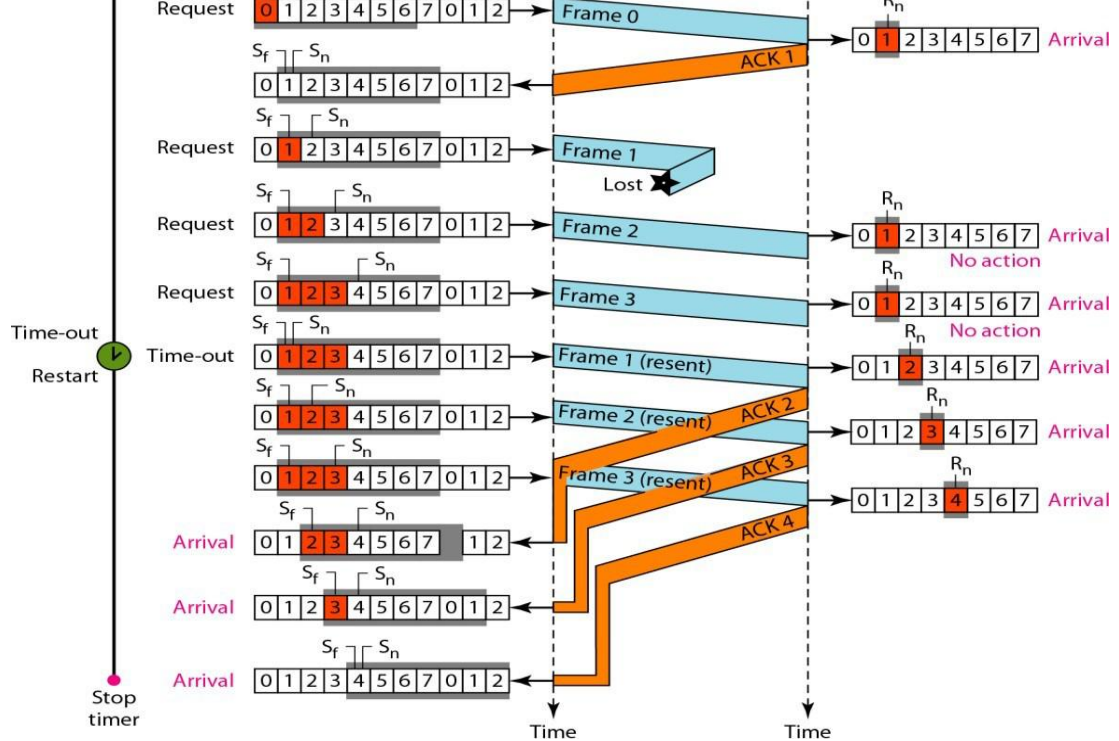
Fig: Receiver window (before sliding (a), After sliding (b))

Timers

Although there can be a timer for each frame that is sent, in our protocol we use only one. The reason is that the timer for the first outstanding frame always expires first; we send all outstanding frames when this timer expires.

Acknowledgment

The receiver sends a positive acknowledgment if a frame has arrived safe and sound and in order. If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting. The silence of the receiver causes the timer of the unacknowledged frame at the sender side to expire. This, in turn, causes the sender to go back and resend all frames, beginning with the one with the expired timer. The receiver does not have to acknowledge each frame received. It can send one cumulative acknowledgment for several frames.



Stop-and-Wait ARQ is a special case of Go-Back-N ARQ in which the size of the send window is 1.

3 Selective Repeat Automatic Repeat Request

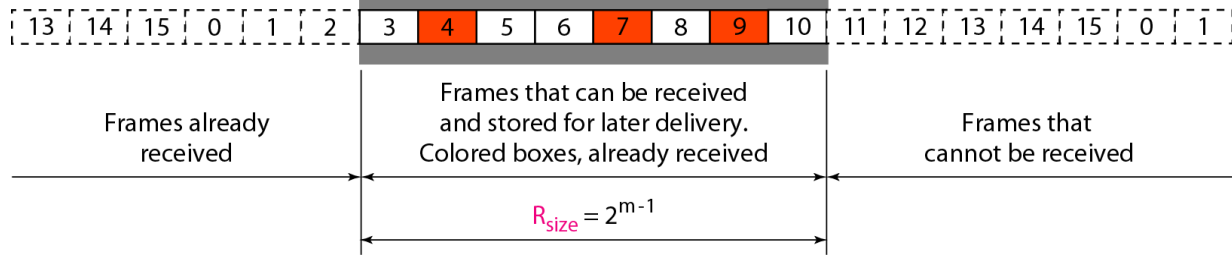
In Go-Back-N ARQ, The receiver keeps track of only one variable, and there is no need to buffer out-of- order frames; they are simply discarded. However, this protocol is very inefficient for a noisy link.

In a noisy link a frame has a higher probability of damage, which means the resending of multiple frames. This resending uses up the bandwidth and slows down the transmission.

For noisy links, there is another mechanism that does not resend N frames when just one frame is damaged; only the damaged frame is resent. This mechanism is called Selective Repeat ARQ.

It is more efficient for noisy links, but the processing at the receiver is more complex.

Sender Window (explain go-back N sender window concept (before & after sliding.) The only difference in sender window between Go-back N and Selective Repeat is Window size)



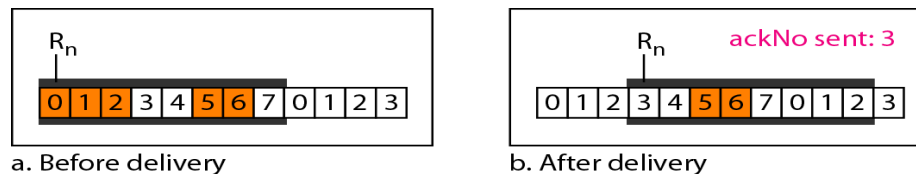
Receiver window

The receiver window in Selective Repeat is totally different from the one in Go Back-N. First, the size of the receive window is the same as the size of the send window (2^{m-1}).

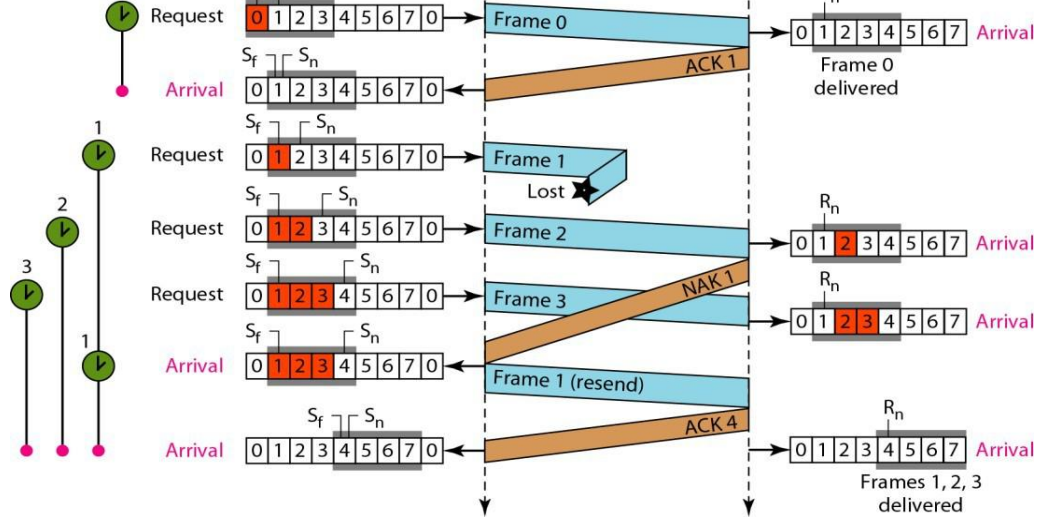
The Selective Repeat Protocol allows as many frames as the size of the receiver window to arrive out of order and be kept until there is a set of in-order frames to be delivered to the network layer. Because the sizes of the send window and receive window are the same, all the frames in the send window can arrive out of order and be stored until they can be delivered. However the receiver never delivers packets out of order to the network layer. Above Figure shows the receive window. Those slots inside the window that are colored define frames that have arrived out of order and are waiting for their neighbors to arrive before delivery to the network layer.

In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m

Delivery of Data in Selective Repeat ARQ:



Flow Diagram



Differences between Go-Back N & Selective Repeat

One main difference is the number of timers. Here, each frame sent or resent needs a timer, which means that the timers need to be numbered (0, 1, 2, and 3). The timer for frame 0 starts at the first request, but stops when the ACK for this frame arrives.

There are two conditions for the delivery of frames to the network layer: First, a set of consecutive frames must have arrived. Second, the set starts from the beginning of the window. After the first arrival, there was only one frame and it started from the beginning of the window. After the last arrival, there are three frames and the first one starts from the beginning of the window.

Another important point is that a NAK is sent.

The next point is about the ACKs. Notice that only two ACKs are sent here. The first one acknowledges only the first frame; the second one acknowledges three frames. In Selective Repeat, ACKs are sent when data are delivered to the network layer. If the data belonging to n frames are delivered in one shot, only one ACK is sent for all of them.

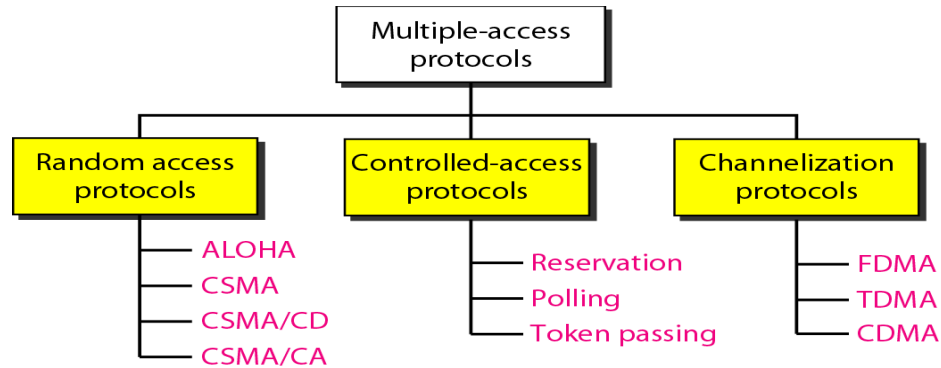
Piggybacking

A technique called **piggybacking** is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.

RANDOM ACCESS PROTOCOLS

We can consider the data link layer as two sub layers. The upper sub layer is responsible for data link control, and the lower sub layer is responsible for resolving access to the shared media

The upper sub layer that is responsible for flow and error control is called the logical link control (LLC) layer; the lower sub layer that is mostly responsible for multiple access resolution is called the media access control (MAC) layer. When nodes or stations are connected and use a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link.



Taxonomy of multiple-access protocols

RANDOM ACCESS

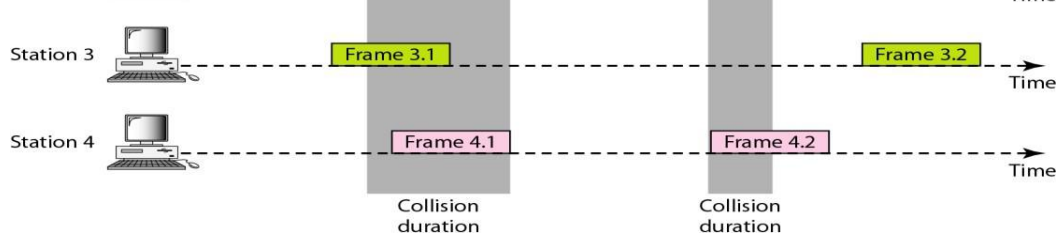
In random access or contention methods, no station is superior to another station and none is assigned the control over another.

Two features give this method its name. First, there is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called *random access*. Second, no rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called *contention* methods.

ALOHA

1 Pure ALOHA

The original ALOHA protocol is called pure ALOHA. This is a simple, but elegant protocol. The idea is that each station sends a frame whenever it has a frame to send. However, since there is only one channel to share, there is the possibility of collision between frames from different stations. Below Figure shows an example of frame collisions in pure ALOHA.

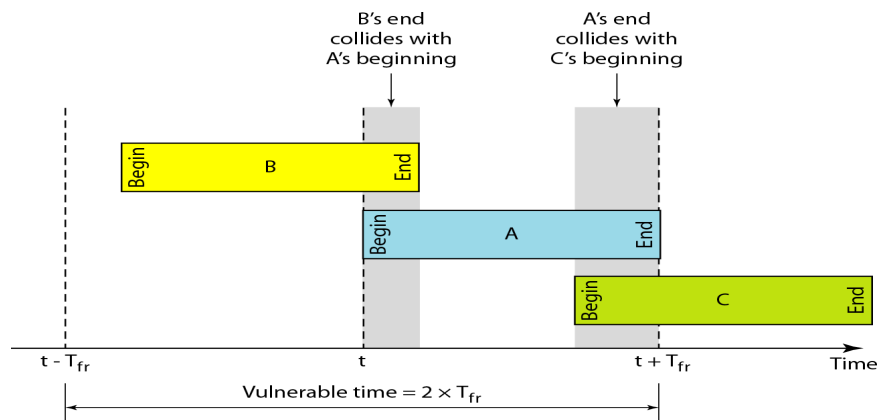


Frames in a pure ALOHA network

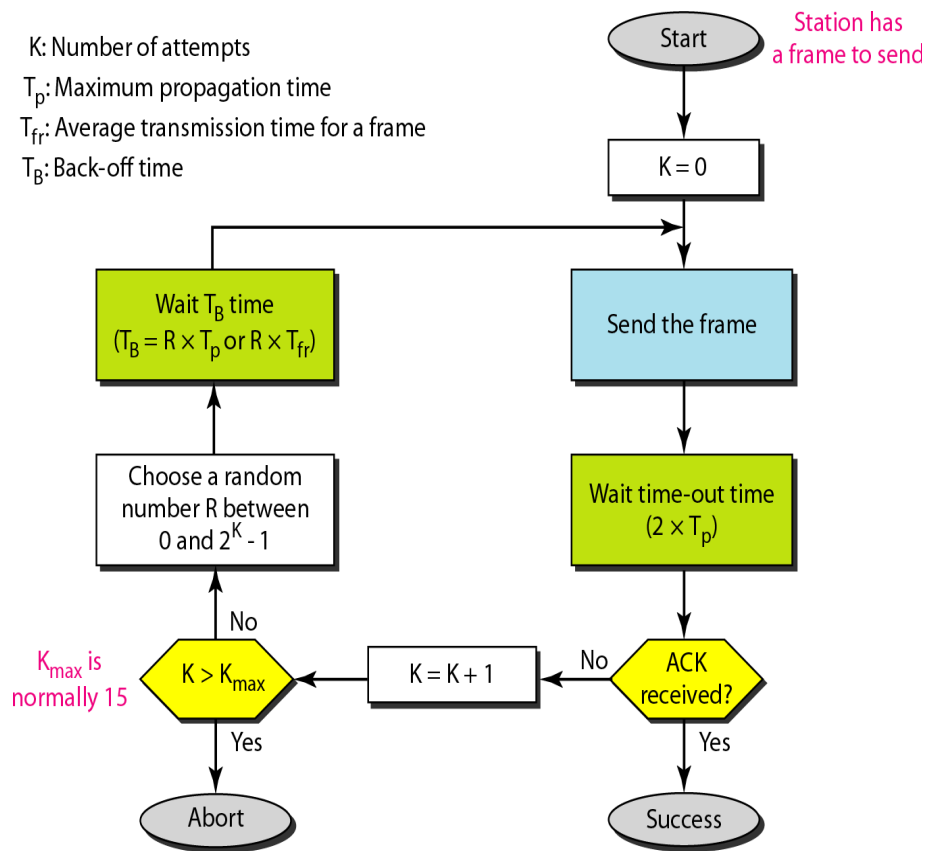
In pure ALOHA, the stations transmit frames whenever they have data to send.

- When two or more stations transmit simultaneously, there is collision and the frames are destroyed.
- In pure ALOHA, whenever any station transmits a frame, it expects the acknowledgement from the receiver.
- If acknowledgement is not received within specified time, the station assumes that the frame (or acknowledgement) has been destroyed.
- If the frame is destroyed because of collision the station waits for a random amount of time and sends it again. This waiting time must be random otherwise same frames will collide again and again.
- Therefore pure ALOHA dictates that when time-out period passes, each station must wait for a random amount of time before resending its frame. This randomness will help avoid more collisions.

Vulnerable time Let us find the length of time, the **vulnerable time**, in which there is a possibility of collision. We assume that the stations send fixed-length frames with each frame taking T_{fr} S to send. Below Figure shows the vulnerable time for station A.



Station A sends a frame at time t . Now imagine station B has already sent a frame between $t - T_{fr}$ and t . This leads to a collision between the frames from station A and station B. The end of B's frame collides with the beginning of A's frame. On the other hand, suppose that station C sends a frame between t and $t + T_{fr}$. Here, there is a collision between frames from station A and station C. The beginning of C's frame collides with the end of A's frame



Procedure for pure ALOHA protocol

Example

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Solution

Average frame transmission time T_{fr} is 200 bits/200 kbps or 1 ms. The vulnerable time is $2 \times 1 \text{ ms} = 2 \text{ ms}$. This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the one 1-ms period that this station is sending.

The throughput for pure ALOHA is $S = G \times e^{-2G}$. The maximum throughput $S_{max} = 0.184$ when $G = (1/2)$.

PROBLEM

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces a. 1000 frames per second b. 500 frames per second c. 250 frames per second. The frame transmission time is 200/200 kbps or 1 ms.

- a. If the system creates 1000 frames per second, this is 1 frame per

- b. If the system creates 500 frames per second, this is $(1/2)$ frame per millisecond. The load is $(1/2)$. In this case $S = G \times e^{-2G}$ or $S = 0.184$ (18.4 percent). This means that the throughput is $500 \times 0.184 = 92$ and that only 92 frames out of 500 will probably survive. Note that this is the maximum throughput case, percentage wise.
- c. If the system creates 250 frames per second, this is $(1/4)$ frame per millisecond. The load is $(1/4)$. In this case $S = G \times e^{-2G}$ or $S = 0.152$ (15.2 percent). This means that the throughput is $250 \times 0.152 = 38$. Only 38 frames out of 250 will probably survive.

2 Slotted ALOHA

Pure ALOHA has a vulnerable time of $2 \times T_{fr}$. This is so because there is no rule that defines when the station can send. A station may send soon after another station has started or soon before another station has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA.

In slotted ALOHA we divide the time into slots of T_{fr} s and force the station to send only at the beginning of the time slot. Figure 3 shows an example of frame collisions in slotted ALOHA

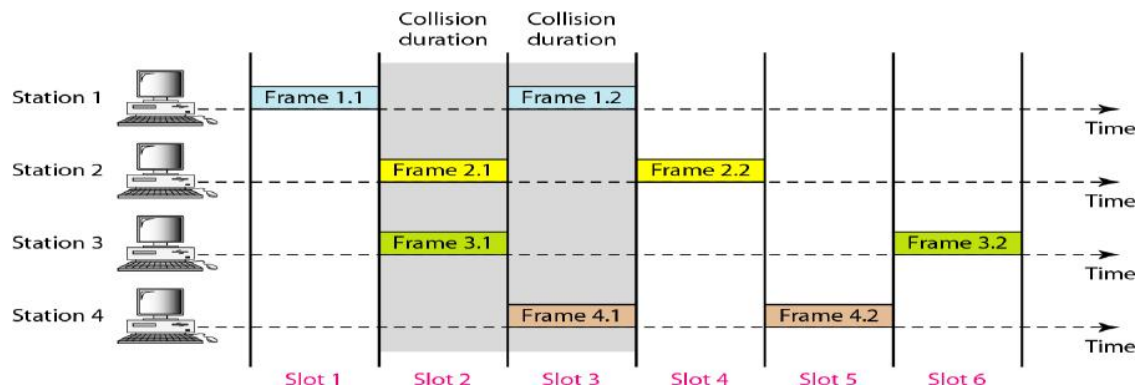
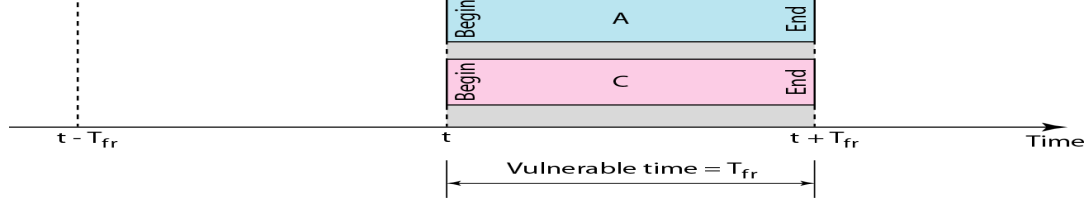


FIG:3

Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame. Of course, there is still the possibility of collision if two stations try to send at the beginning of the same time slot. However, the vulnerable time is now reduced to one-half, equal to T_{fr} . Figure 4 shows the situation

Below fig shows that the vulnerable time for slotted ALOHA is one-half that of pure ALOHA. Slotted ALOHA vulnerable time = T_{fr}



The throughput for slotted ALOHA is $S = G \times e^{-G}$. The maximum throughput $S_{max} = 0.368$ when $G = 1$.

A slotted ALOHA network transmits 200-bit frames using a shared channel with a 200-Kbps bandwidth. Find the throughput if the system (all stations together) produces

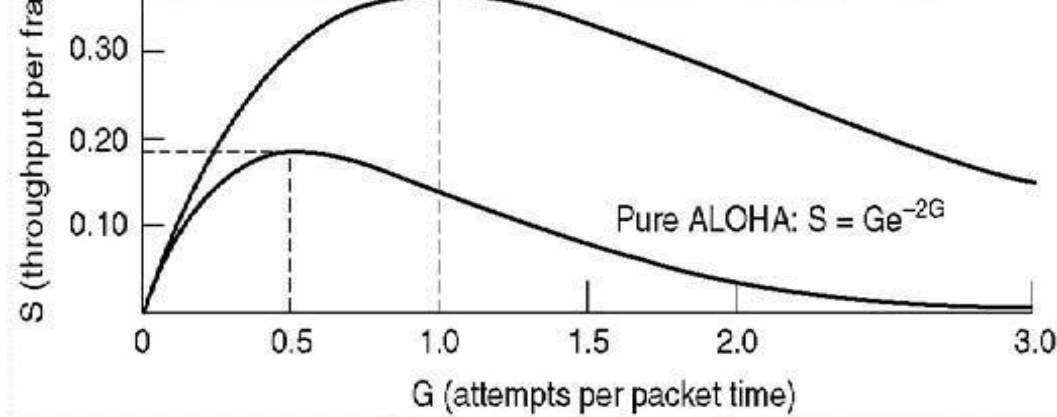
- 1000 frames per second
- 500 frames per second
- 250 frames per second

Solution

This situation is similar to the previous exercise except that the network is using slotted ALOHA instead of pure ALOHA. The frame transmission time is $200/200$ kbps or 1 ms.

- In this case G is 1. So $S = G \times e^{-G}$ or $S = 0.368$ (36.8 percent). This means that the throughput is $1000 \times 0.368 = 368$ frames. Only 368 out of 1000 frames will probably survive. Note that this is the maximum throughput case, percentagewise.
- Here G is $1/2$. In this case $S = G \times e^{-G}$ or $S = 0.303$ (30.3 percent). This means that the throughput is $500 \times 0.303 = 151$. Only 151 frames out of 500 will probably survive.
- Now G is $1/4$. In this case $S = G \times e^{-G}$ or $S = 0.195$ (19.5 percent). This means that the throughput is $250 \times 0.195 = 49$. Only 49 frames out of 250 will probably survive.

Comparison between Pure Aloha & Slotted Aloha



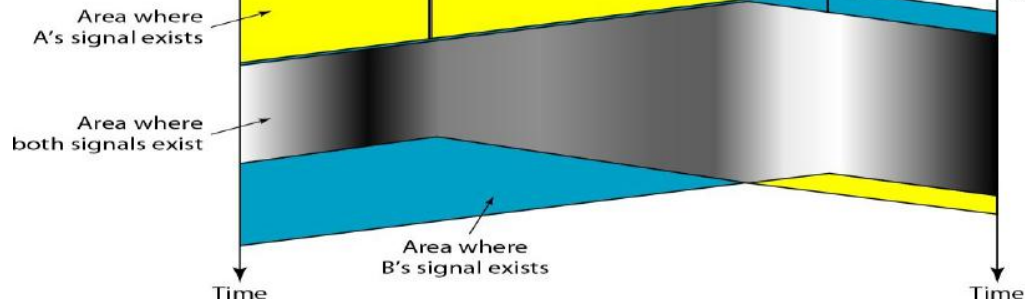
Carrier Sense Multiple Access (CSMA)

To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it. Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending. In other words, CSMA is based on the principle "sense before transmit" or "listen before talk."

CSMA can reduce the possibility of collision, but it cannot eliminate it. The reason for this is shown in below Figure. Stations are connected to a shared channel (usually a dedicated medium).

The possibility of collision still exists because of propagation delay; station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.

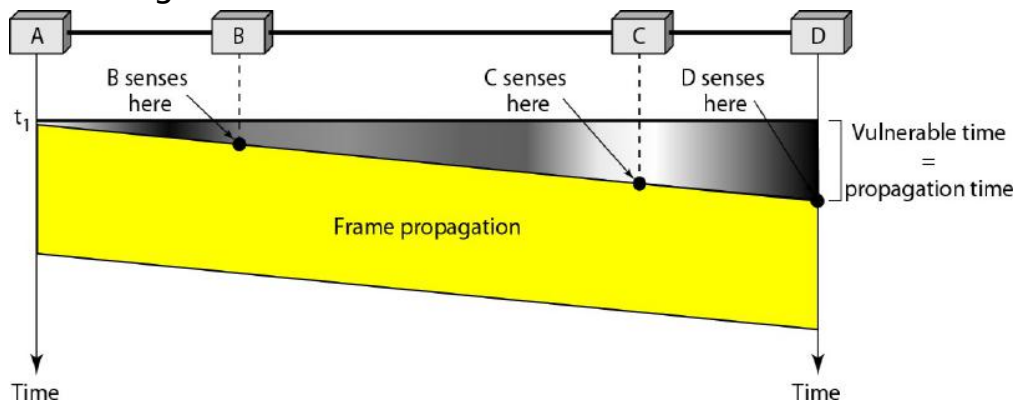
At time t_1' station B senses the medium and finds it idle, so it sends a frame. At time t_2 ($t_2 > t_1'$) station C senses the medium and finds it idle because, at this time, the first bits from station B have not reached station C. Station C also sends a frame. The two signals collide and both frames are destroyed.



Space/time model of the collision in CSMA

Vulnerable Time

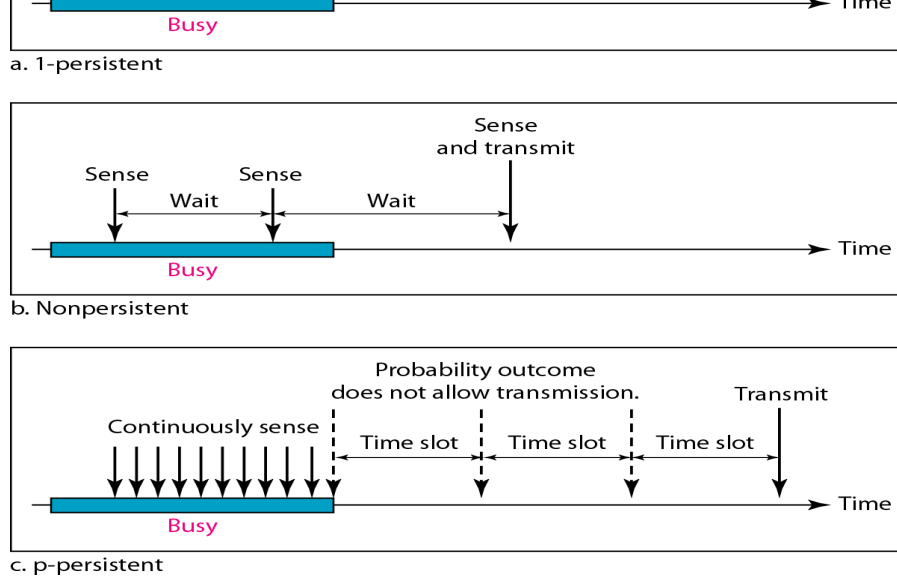
The vulnerable time for CSMA is the propagation time T_p . This is the time needed for a signal to propagate from one end of the medium to the other. When a station sends a frame, and any other station tries to send a frame during this time, a collision will result. But if the first bit of the frame reaches the end of the medium, every station will already have heard the bit and will refrain from sending.



Vulnerable time in CSMA

Persistence Methods

What should a station do if the channel is busy? What should a station do if the channel is idle? Three methods have been devised to answer these questions: the 1-persistent method, the non-persistent method, and the p-persistent method.



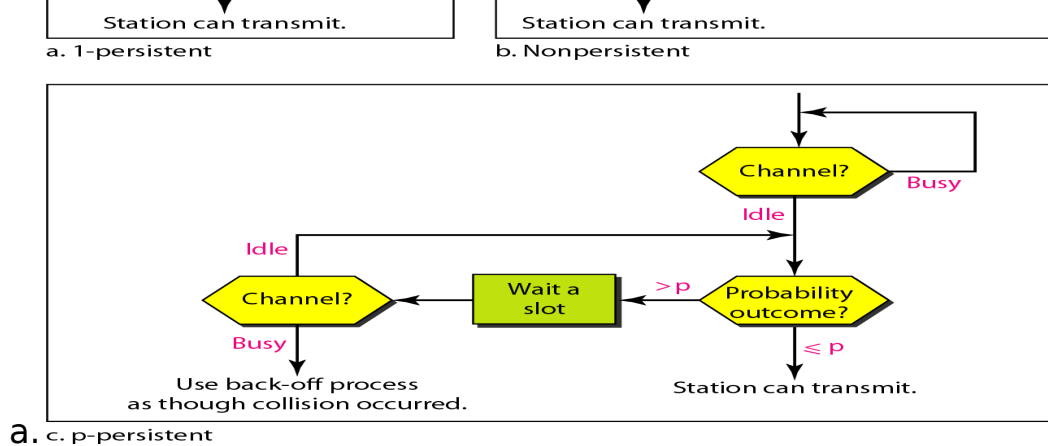
1-Persistent: In this method, after the station finds the line idle, it sends its frame immediately (with probability 1). This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.

Non-persistent: a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again. This approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously. However, this method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.

p-Persistent: This is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time. The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves efficiency.

In this method, after the station finds the line idle it follows these steps:

1. With probability p , the station sends its frame.
2. With probability $q = 1 - p$, the station waits for the beginning of the next time slot and checks the line again.
 - a. If the line is idle, it goes to step 1.
 - b. If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.

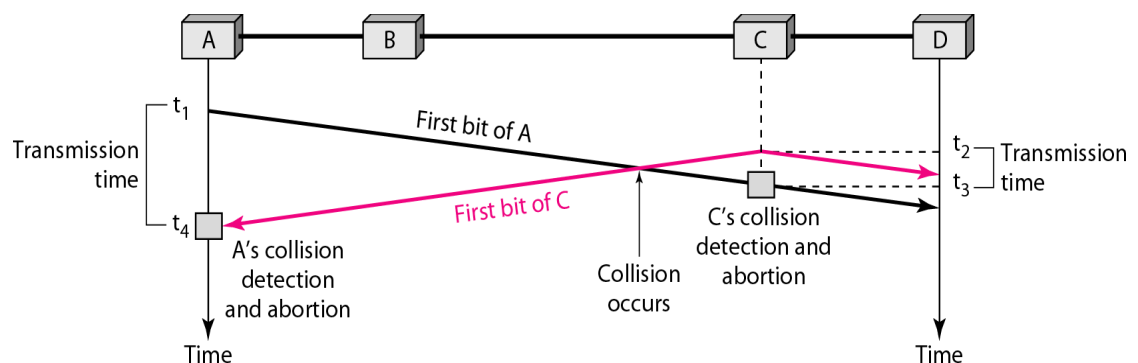


Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

The CSMA method does not specify the procedure following a collision. Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision.

In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If, however, there is a collision, the frame is sent again.

To better understand CSMA/CD, let us look at the first bits transmitted by the two stations involved in the collision. Although each station continues to send bits in the frame until it detects the collision, we show what happens as the first bits collide. In below Figure, stations A and C are involved in the collision.

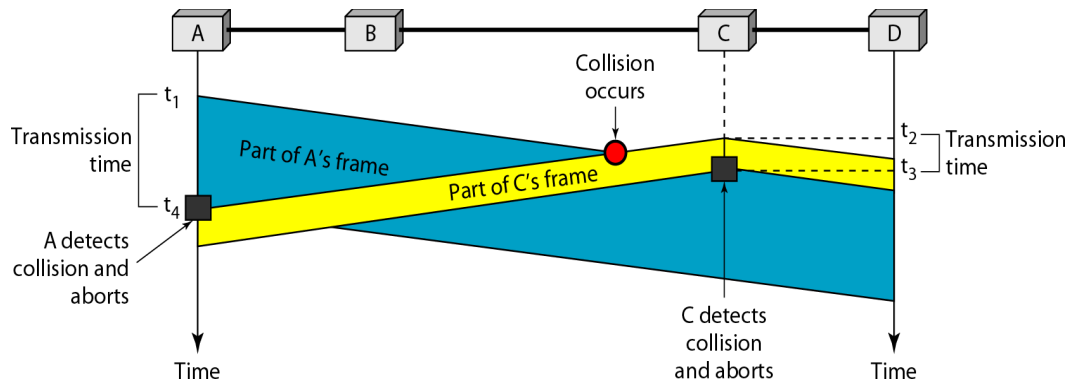


Collision of the first bit in CSMA/CD

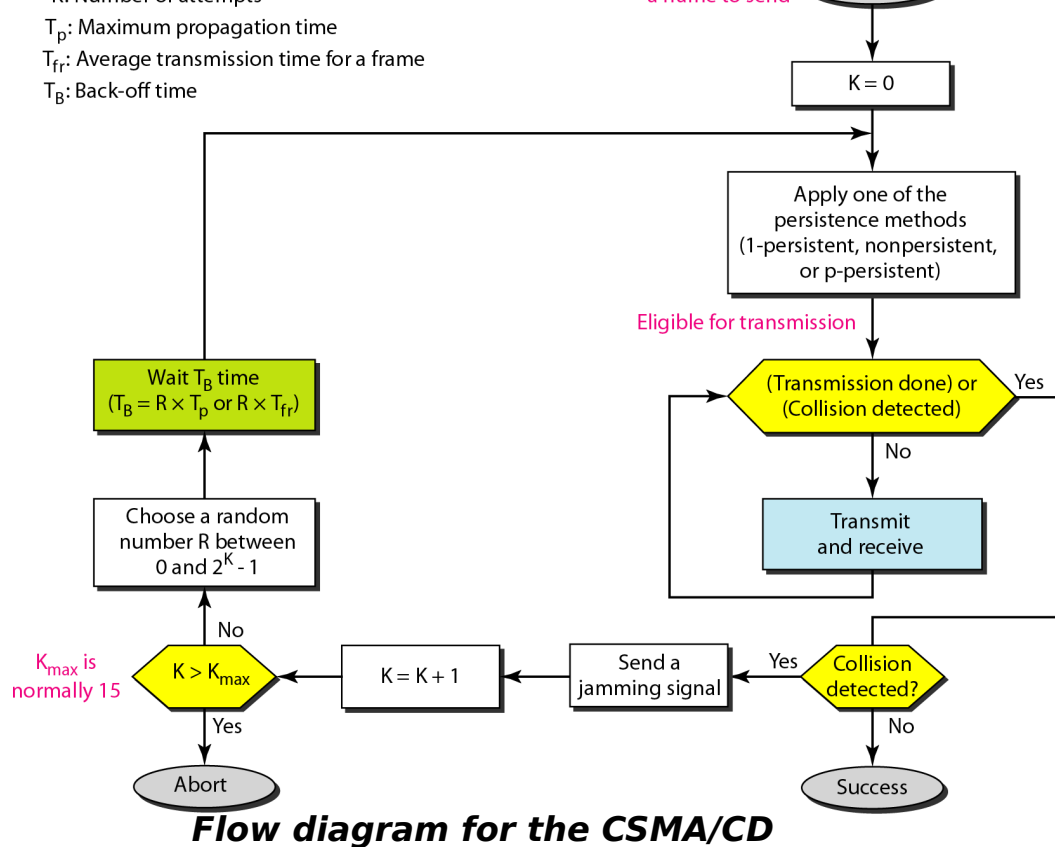
At time t_1 , station A has executed its persistence procedure and starts sending the bits of its frame. At time t_2 , station C has not yet sensed the first bit sent by A. Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right. The collision occurs sometime after time t_2 . Station C detects a collision at time t_3 when it receives the first bit of A's frame. Station C immediately (or after a short time, but we assume immediately) aborts transmission.

Station A detects collision at time t_4 when it receives the first bit of C's frame; it also immediately aborts transmission. Looking at the figure, we see that A

the last bit of the frame, the sending station must detect a collision, if any, and abort the transmission. This is so because the station, once the entire frame is sent, does not keep a copy of the frame and does not monitor the line for collision detection. Therefore, the frame transmission time T_{fr} must be at least two times the maximum propagation time T_p . To understand the reason, let us think about the worst-case scenario. If the two stations involved in a collision are the maximum distance apart, the signal from the first takes time T_p to reach the second, and the effect of the collision takes another time T_p to reach the first. So the requirement is that the first station must still be transmitting after $2T_p$.



T_p : Maximum propagation time
 T_{fr} : Average transmission time for a frame
 T_B : Back-off time



PROBLEM

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices and ignoring the time needed to send a jamming signal, as we see later) is $25.6 \mu\text{s}$, what is the minimum size of the frame?

SOL

The frame transmission time is $T_{fr} = 2 \times T_p = 51.2 \mu\text{s}$. This means, in the worst case, a station needs to transmit for a period of $51.2 \mu\text{s}$ to detect the collision. The minimum size of the frame is $10 \text{ Mbps} \times 51.2 \mu\text{s} = 512 \text{ bits}$ or 64 bytes . This is actually the minimum size of the frame for Standard Ethernet.

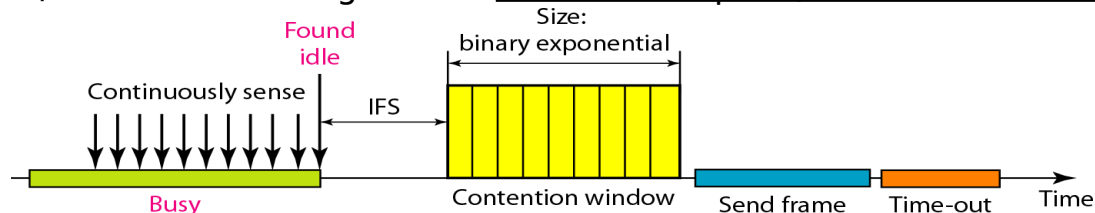
DIFFERENCES BETWEEN ALOHA & CSMA/CD

The first difference is the addition of the persistence process. We need to sense the channel before we start sending the frame by using one of the persistence processes

The second difference is the frame transmission. In ALOHA, we first transmit the entire frame and then wait for an acknowledgment. In CSMA/CD, transmission and collision detection is a continuous process. We do not send the entire frame and then look for a collision. The station transmits and receives continuously and simultaneously

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

We need to avoid collisions on wireless networks because they cannot be detected. Carrier sense multiple access with collision avoidance (CSMA/CA) was invented for wireless network. Collisions are avoided through the use of CSMA/CA's three strategies: the inter frame space, the contention window, and



acknowledgments, as shown in Figure

Timing in CSMA/CA

Inter frame Space (IFS)

First, collisions are avoided by deferring transmission even if the channel is found idle. When an idle channel is found, the station does not send immediately. It waits for a period of time called the inter frame space or IFS.

Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting. The distant station's signal has not yet reached this station. The IFS time allows the front of the transmitted signal by the distant station to reach this station. If after the IFS time the channel is still idle, the station can send, but it still needs to wait a time equal to the contention time. The IFS variable can also be used to prioritize stations or frame types. For example, a station that is assigned shorter IFS has a higher priority.

In CSMA/CA, the IFS can also be used to define the priority of a station or a frame.

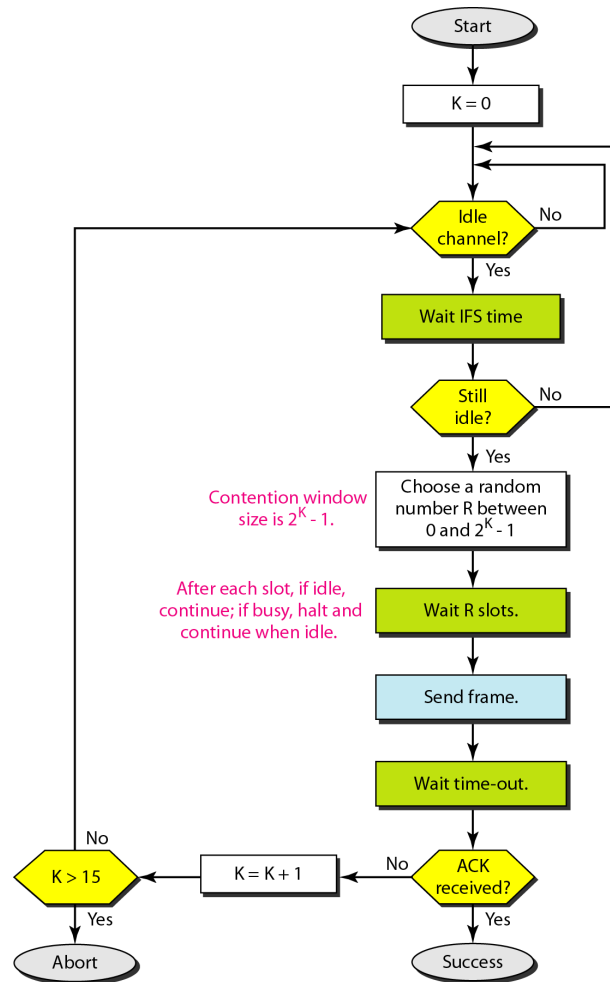
Contention Window

The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes according to the binary exponential back-off strategy. This means that it is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time. This is very similar to the p-persistent method except that a random outcome defines the number of slots taken by the waiting station.

One interesting point about the contention window is that the station needs to sense the channel after each time slot. However, if the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle. This gives priority to the station with the longest waiting time.

Acknowledgment

With all these precautions, there still may be a collision resulting in destroyed data. In addition, the data may be corrupted during the transmission. The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.



This is the CSMA protocol with collision avoidance.

- The station ready to transmit, senses the line by using one of the persistent strategies.
- As soon as it finds the line to be idle, the station waits for an IFS (Inter frame space) amount of time.
- If then waits for some random time and sends the frame.
- After sending the frame, it sets a timer and waits for the acknowledgement from the receiver.
- If the acknowledgement is received before expiry of the timer, then the transmission is successful.
- But if the transmitting station does not receive the expected acknowledgement before the timer expiry then it increments the back off

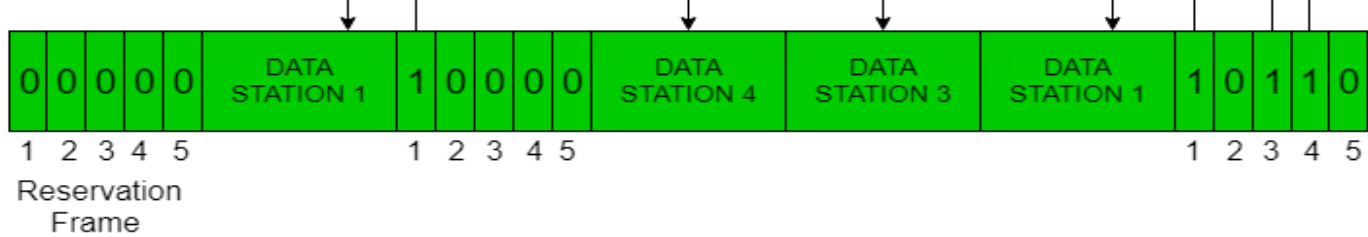
In controlled access, the stations seek information from one another to find which station has the right to send. It allows only one node to send at a time, to avoid collision of messages on shared medium. The three controlled-access methods are:

1 Reservation 2 Polling 3 Token Passing

Reservation

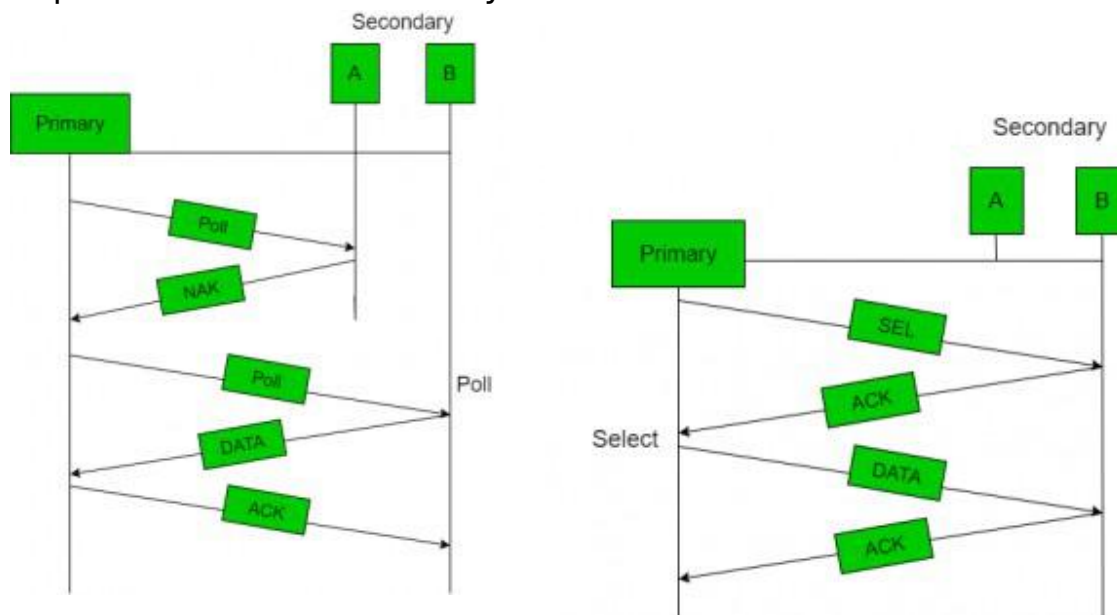
- In the reservation method, a station needs to make a reservation before sending data.
- The time line has two kinds of periods:
 1. Reservation interval of fixed time length
 2. Data transmission period of variable frames.
- If there are M stations, the reservation interval is divided into M slots, and each station has one slot.
- Suppose if station 1 has a frame to send, it transmits 1 bit during the slot 1. No other station is allowed to transmit during this slot.
- In general, i^{th} station may announce that it has a frame to send by inserting a 1 bit into i^{th} slot. After all N slots have been checked, each station knows which stations wish to transmit.
- The stations which have reserved their slots transfer their frames in that order.
- After data transmission period, next reservation interval begins.
- Since everyone agrees on who goes next, there will never be any collisions.

The following figure shows a situation with five stations and a five slot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.



Polling

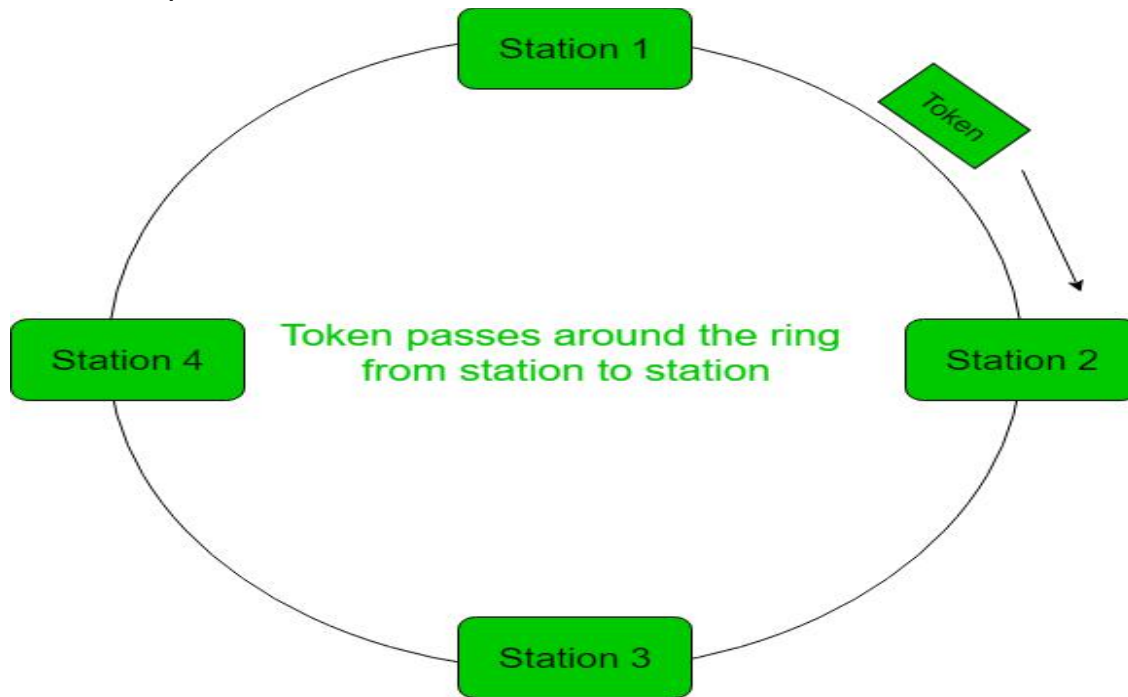
- Polling process is similar to the roll-call performed in class. Just like the teacher, a controller sends a message to each node in turn.
- In this, one acts as a primary station(controller) and the others are secondary stations. All data exchanges must be made through the controller.
- The message sent by the controller contains the address of the node being selected for granting access.
- Although all nodes receive the message but the addressed one responds to it and sends data, if any. If there is no data, usually a “poll reject”(NAK) message is sent back.
- Problems include high overhead of the polling messages and high dependence on the reliability of the controller.



Token Passing

- In token passing scheme, the stations are connected logically to each other in form of ring and access of stations is governed by tokens.
- A token is a special bit pattern or a small message, which circulate from one station to the next in the some predefined order.

- In both cases, token represents permission to send. If a station has a frame queued for transmission when it receives the token, it can send that frame before it passes the token to the next station. If it has no queued frame, it passes the token simply.
- After sending a frame, each station must wait for all N stations (including itself) to send the token to their neighbors and the other N - 1 stations to send a frame, if they have one.
- There exists problems like duplication of token or token is lost or insertion of new station, removal of a station, which need be tackled for correct and reliable operation of this scheme.



Error Detection

Error

A condition when the receiver's information does not match with the sender's information. During transmission, digital signals suffer from noise that can introduce errors in the binary bits travelling from sender to receiver. That means a 0 bit may change to 1 or a 1 bit may change to 0.

Error Detecting Codes (Implemented either at Data link layer or Transport Layer of OSI Model)

Whenever a message is transmitted, it may get scrambled by noise or data may get corrupted. To avoid this, we use error-detecting codes which are additional data added to a given digital message to help us detect if any error has occurred during transmission of the message.

Basic approach used for error detection is the use of redundancy bits, where

1. Simple Parity check

2. Two-dimensional Parity check

3. Checksum

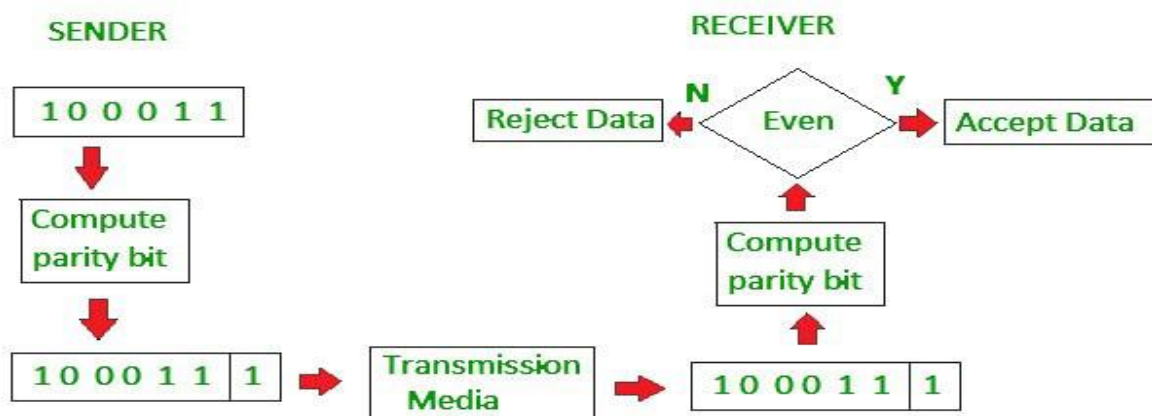
4. Cyclic redundancy check

Simple Parity check

Blocks of data from the source are subjected to a check bit or parity bit generator form, where a parity of : 1 is added to the block if it contains odd number of 1's, and

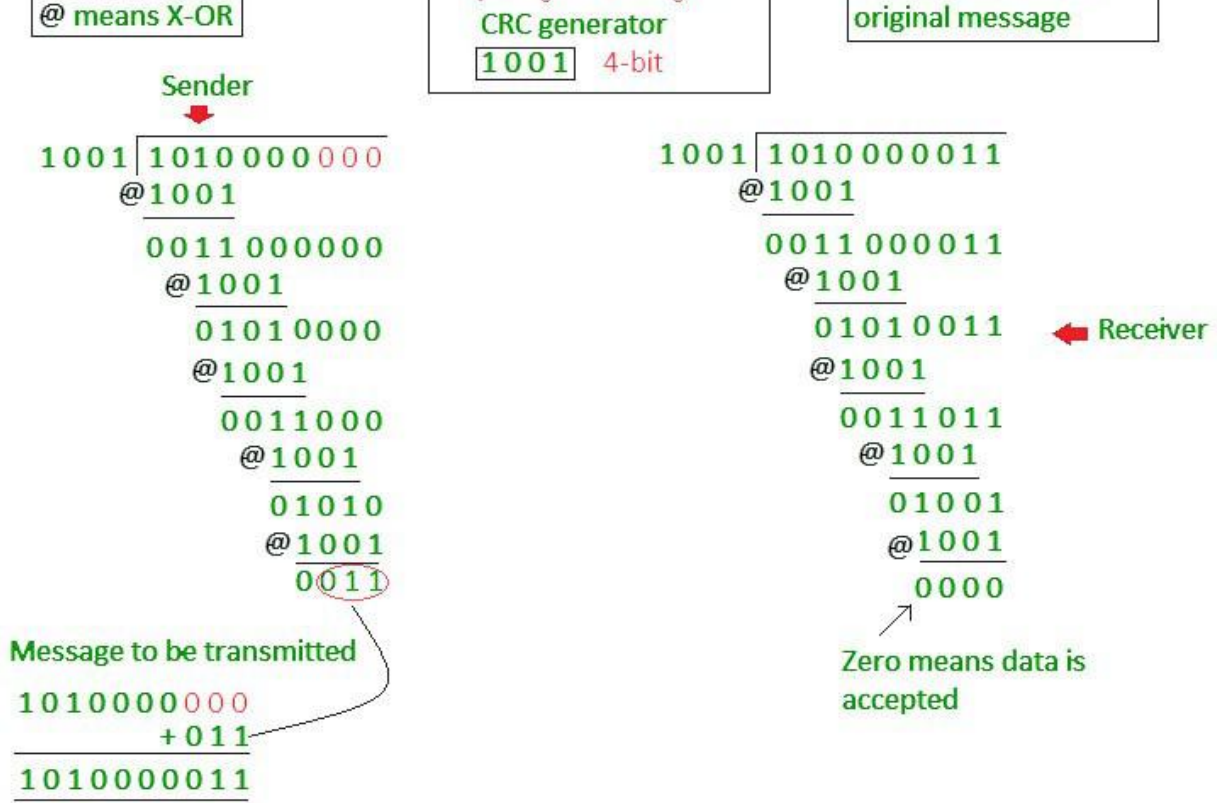
0 is added if it contains even number of 1's

This scheme makes the total number of 1's even, that is why it is called even parity checking.



Two-dimensional Parity check

Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are also calculated for all columns, then both are sent along with the data. At the receiving end these are compared with the parity bits calculated on the received data.



- Unlike checksum scheme, which is based on addition, CRC is based on binary division.
- In CRC, a sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
- At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted.
- A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.

Error Correction

Error Correction codes are used to detect and correct the errors when data is transmitted from the sender to the receiver.

Error Correction can be handled in two ways:

Backward error correction: Once the error is discovered, the receiver requests the sender to retransmit the entire data unit.

Forward error correction: In this case, the receiver uses the error-correcting code which automatically corrects the errors.

A single additional bit can detect the error, but cannot correct it.

some additional redundant bits.

Suppose r is the number of redundant bits and d is the total number of the data bits. The number of redundant bits r can be calculated by using the formula:

$$2^r \geq d+r+1$$

The value of r is calculated by using the above formula. For example, if the value of d is 4, then the possible smallest value that satisfies the above relation would be 3.

To determine the position of the bit which is in error, a technique developed by R.W Hamming is Hamming code which can be applied to any length of the data unit and uses the relationship between data units and redundant units.

Hamming Code

Parity bits: The bit which is appended to the original data of binary bits so that the total number of 1s is even or odd.

Even parity: To check for even parity, if the total number of 1s is even, then the value of the parity bit is 0. If the total number of 1s occurrences is odd, then the value of the parity bit is 1.

Odd Parity: To check for odd parity, if the total number of 1s is even, then the value of parity bit is 1. If the total number of 1s is odd, then the value of parity bit is 0.

Algorithm of Hamming code:

An information of ' d ' bits are added to the redundant bits ' r ' to form $d+r$.

The location of each of the $(d+r)$ digits is assigned a decimal value.

The ' r ' bits are placed in the positions $1, 2, \dots, 2^{k-1}$

At the receiving end, the parity bits are recalculated. The decimal value of the parity bits determines the position of an error.

Relationship b/w Error position & binary number.

Error Position	Binary Number
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Let's understand the concept of Hamming code through an example:

Suppose the original data is 1010 which is to be sent.

Total number of data bits ' d ' = 4

Number of redundant bits r : $2^r \geq d+r+1$

Determining the position of the redundant bits

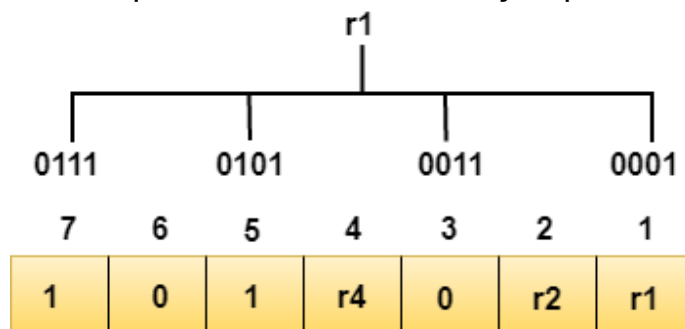
The number of redundant bits is 3. The three bits are represented by r_1 , r_2 , r_4 . The position of the redundant bits is calculated with corresponds to the raised power of 2. Therefore, their corresponding positions are 1, 2^1 , 2^2 . The position of $r_1 = 1$, The position of $r_2 = 2$, The position of $r_4 = 4$

Representation of Data on the addition of parity bits:

7	6	5	4	3	2	1
1	0	1	r_4	0	r_2	r_1

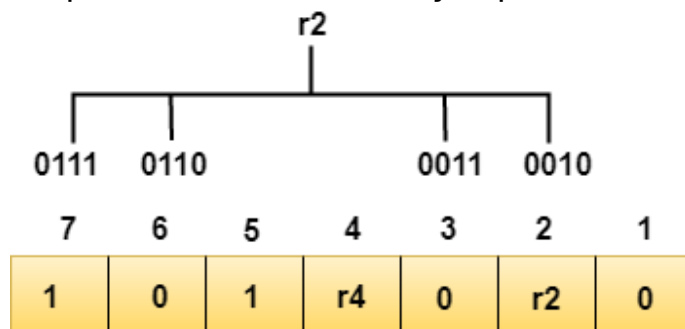
Determining the Parity bits

Determining the r_1 bit: The r_1 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the first position.



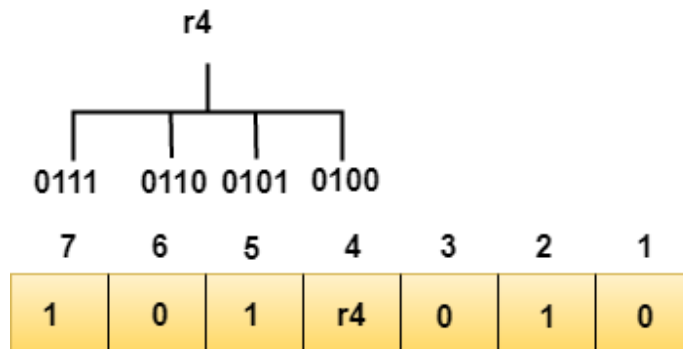
We observe from the above figure that the bit position that includes 1 in the first position are 1, 3, 5, 7. Now, we perform the even-parity check at these bit positions. The total number of 1 at these bit positions corresponding to r_1 is even, therefore, the value of the r_1 bit is 0.

Determining r_2 bit: The r_2 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the second position



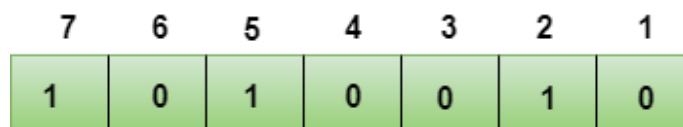
We observe from the above figure that the bit positions that includes 1 in the second position are 2, 3, 6, 7. Now, we perform the even-parity check at these

Determining r4 bit: The r4 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the third position.



We observe from the above figure that the bit positions that includes 1 in the third position are 4, 5, 6, 7. Now, we perform the even-parity check at these bit positions. The total number of 1 at these bit positions corresponding to r4 is even, therefore, the value of the r4 bit is 0.

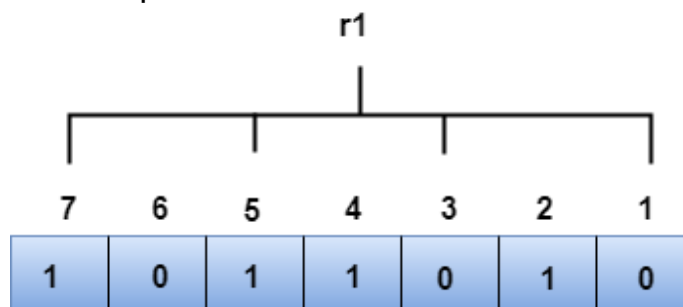
Data transferred is given below:



Suppose the 4th bit is changed from 0 to 1 at the receiving end, then parity bits are recalculated.

R1 bit

The bit positions of the r1 bit are 1,3,5,7



We observe from the above figure that the binary representation of r1 is 1100. Now, we perform the even-parity check, the total number of 1s appearing in the r1 bit is an even number. Therefore, the value of r1 is 0.

R2 bit

The bit positions of r2 bit are 2,3,6,7.

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

We observe from the above figure that the binary representation of r2 is 1001. Now, we perform the even-parity check, the total number of 1s appearing in the r2 bit is an even number. Therefore, the value of r2 is 0.

R4 bit

The bit positions of r4 bit are 4,5,6,7.

r4						
7	6	5	4	3	2	1
1	0	1	1	0	1	0

We observe from the above figure that the binary representation of r4 is 1011. Now, we perform the even-parity check, the total number of 1s appearing in the r4 bit is an odd number. Therefore, the value of r4 is 1.

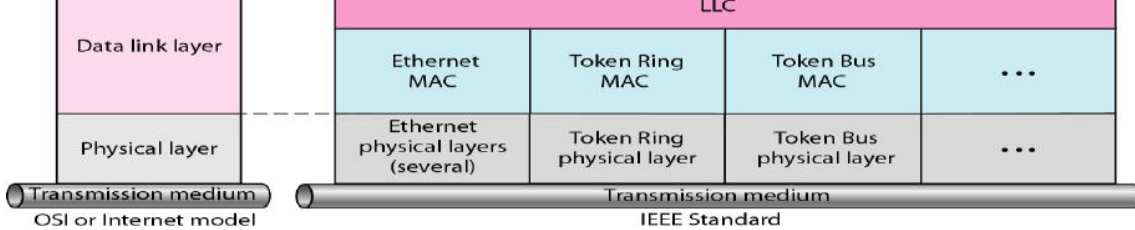
The binary representation of redundant bits, i.e., r4r2r1 is 100, and its corresponding decimal value is 4. Therefore, the error occurs in a 4th bit position. The bit value must be changed from 1 to 0 to correct the error.

Wired LANs: Ethernet

In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers. Project 802 is a way of specifying functions of the physical layer and the data link layer of major LAN protocols.

The relationship of the 802 Standard to the traditional OSI model is shown in below Figure. The IEEE has subdivided the data link layer into two sub layers: logical link control (LLC) and media access control).

IEEE has also created several physical layer standards for different LAN protocols



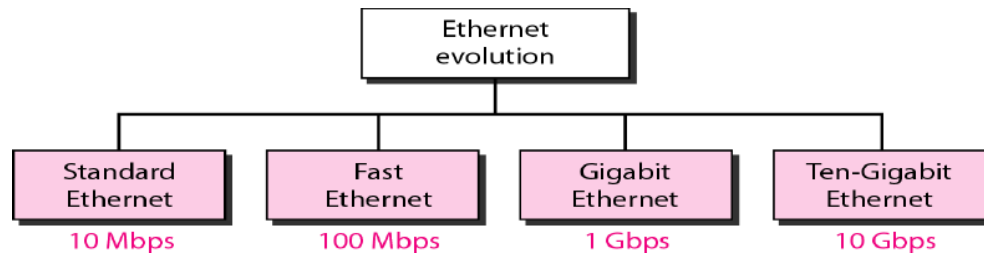
IEEE standard for LANs

STANDARD ETHERNET

The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC). Since then, it has gone through four generations.

Standard Ethernet (10 Mbps), Fast Ethernet (100 Mbps), Gigabit Ethernet (1 Gbps), and Ten-Gigabit Ethernet (10 Gbps),

We briefly discuss the Standard (or traditional) Ethernet in this section



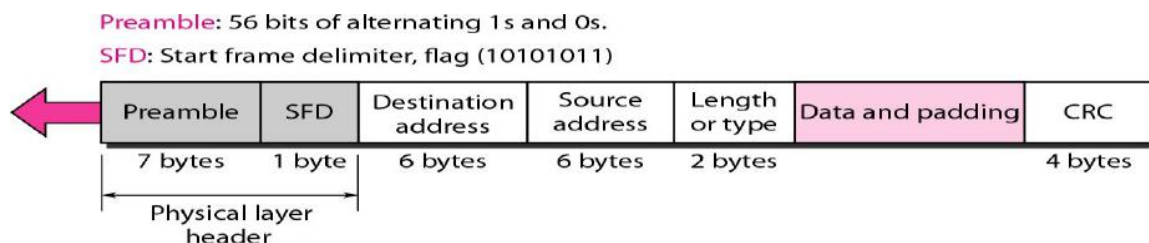
Ethernet evolution through four generations

MAC Sublayer

In Standard Ethernet, the MAC sublayer governs the operation of the access method. It also frames data received from the upper layer and passes them to the physical layer.

Frame Format

The Ethernet frame contains seven fields: preamble, SFD, DA, SA, length or type of protocol data unit (PDU), upper-layer data, and the CRC. Ethernet does not provide any mechanism for acknowledging received frames, making it what is known as an unreliable medium. Acknowledgments must be implemented at the higher layers. The format of the MAC frame is shown in below figure



alternating 0s and 1s that alerts the receiving system to the coming frame and enables it to synchronize its input timing. The pattern provides only an alert and a timing pulse. The 56-bit pattern allows the stations to miss some bits at the beginning of the frame. The preamble is actually added at the physical layer and is not (formally) part of the frame.

Start frame delimiter (SFD). The second field (1 byte: 10101011) signals the beginning of the frame. The SFD warns the station or stations that this is the last chance for synchronization. The last 2 bits is 11 and alerts the receiver that the next field is the destination address.

Destination address (DA). The DA field is 6 bytes and contains the physical address of the destination station or stations to receive the packet.

Source address (SA). The SA field is also 6 bytes and contains the physical address of the sender of the packet.

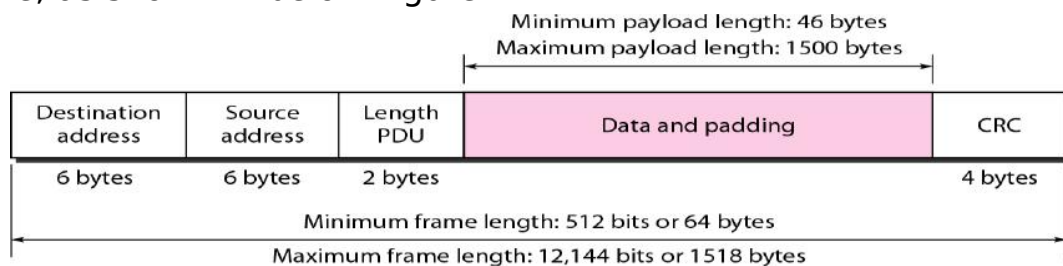
Length or type. This field is defined as a type field or length field. The original Ethernet used this field as the type field to define the upper-layer protocol using the MAC frame. The IEEE standard used it as the length field to define the number of bytes in the data field. Both uses are common today.

Data. This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes.

CRC. The last field contains error detection information, in this case a CRC-32

Frame Length

Ethernet has imposed restrictions on both the minimum and maximum lengths of a frame, as shown in below Figure



Minimum and maximum lengths

An Ethernet frame needs to have a minimum length of 512 bits or 64 bytes. Part of this length is the header and the trailer. If we count 18 bytes of header and trailer (6 bytes of source address, 6 bytes of destination address, 2 bytes of length or type, and 4 bytes of CRC), then the minimum length of data from the upper layer is $64 - 18 = 46$ bytes. If the upper-layer packet is less than 46 bytes, padding is added to make up the difference

The standard defines the maximum length of a frame (without preamble and SFD field) as 1518 bytes. If we subtract the 18 bytes of header and trailer,

Second, the maximum length restriction prevents one station from monopolizing the shared medium, blocking other stations that have data to send.

Addressing

The Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes.

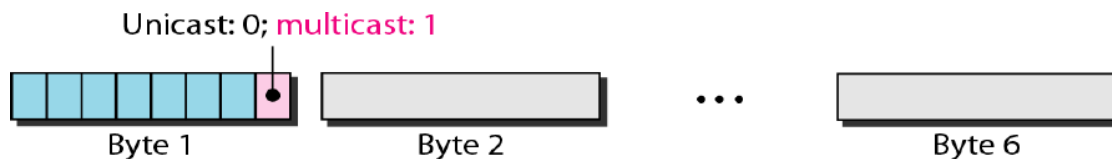
Example of an Ethernet address in hexadecimal notation

06 : 01 : 02 : 01 : 2C : 4B

6 bytes = 12 hex digits = 48 bits

Unicast, Multicast, and Broadcast Addresses A source address is always a unicast address-the frame comes from only one station. The destination address, however, can be **unicast, multicast, or broadcast**. Below Figure shows how to distinguish a unicast address from a multicast address.

If the least significant bit of the first byte in a destination address is 0, the address is unicast; otherwise, it is multicast.



Unicast and multicast addresses

A unicast destination address defines only one recipient; the relationship between the sender and the receiver is one-to-one.

A multicast destination address defines a group of addresses; the relationship between the sender and the receivers is one-to-many.

The broadcast address is a special case of the multicast address; the recipients are all the stations on the LAN. A broadcast destination address is forty-eight 1s.

Access Method: CSMA/CD

Standard Ethernet uses I-persistent CSMA/CD

Slot Time In an Ethernet network.

Slot time = round-trip time + time required to send the jam sequence

The slot time in Ethernet is defined in bits. It is the time required for a station

Slot Time and Maximum Network Length There is a relationship between the slot time and the maximum length of the network (collision domain). It is dependent on the propagation speed of the signal in the particular medium.

In most transmission media, the signal propagates at 2×10^8 m/s (two-thirds of the rate for propagation in air).

For traditional Ethernet, we calculate

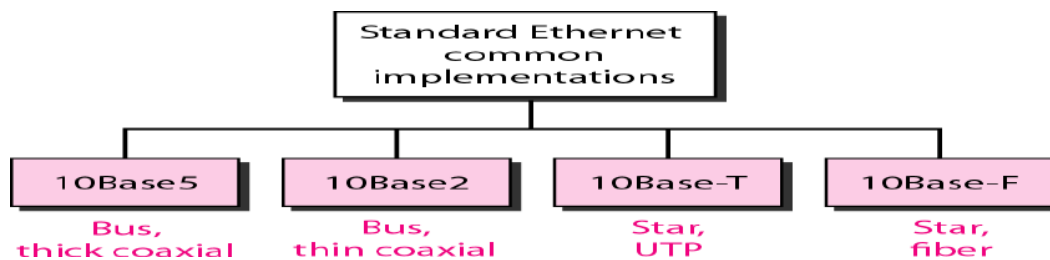
$$\text{MaxLength} = \text{PropagationSpeed} \times (\text{SlotTime}/2)$$

$$\text{MaxLength} = (2 \times 10^8) \times (51.2 \times 10^{-6}) / 2 = 5120 \text{ m}$$

Of course, we need to consider the delay times in repeaters and interfaces, and the time required to send the jam sequence. These reduce the maximum-length of a traditional Ethernet network to 2500 m, just 48 percent of the theoretical calculation. $\text{MaxLength} = 2500 \text{ m}$

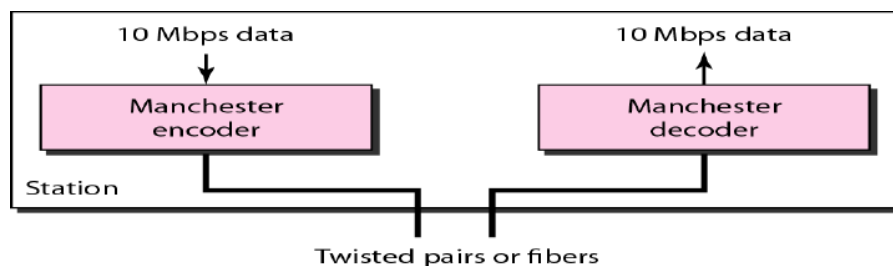
Physical Layer

The Standard Ethernet defines several physical layer implementations; four of the most common, are shown in Figure



Encoding and Decoding

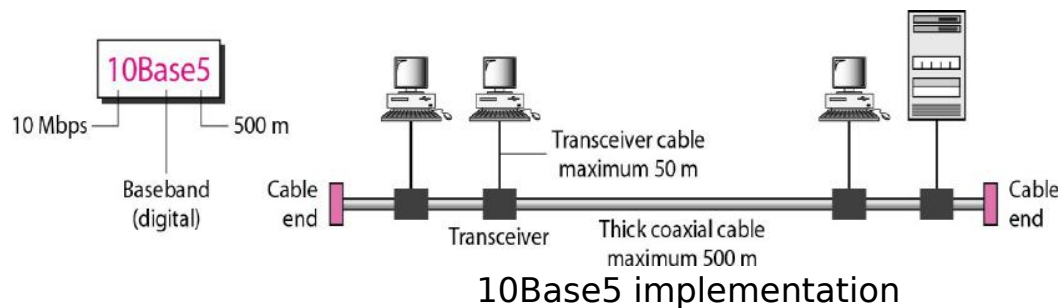
All standard implementations use digital signaling (baseband) at 10 Mbps. At the sender, data are converted to a digital signal using the Manchester scheme; at the receiver, the received signal is interpreted as Manchester and decoded into data. Manchester encoding is self-synchronous, providing a transition at each bit interval. Figure shows the encoding scheme for Standard Ethernet





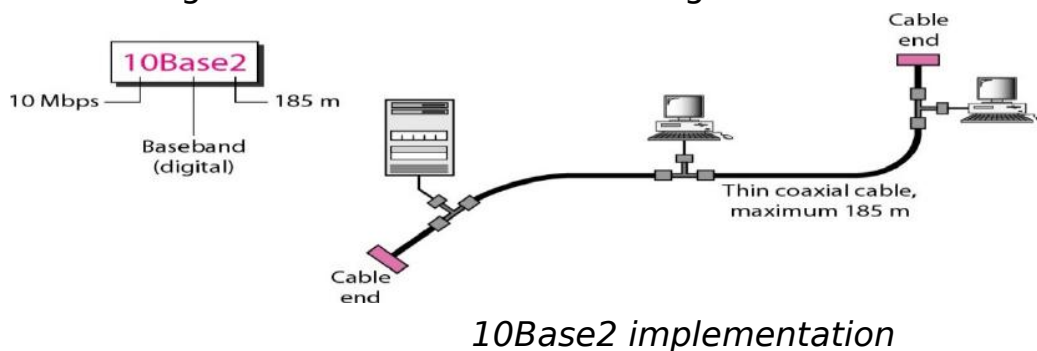
10Base5: Thick Ethernet

The first implementation is called **10Base5, thick Ethernet, or Thicknet**. 10Base5 was the first Ethernet specification to use a bus topology with an external **transceiver** (transmitter/receiver) connected via a tap to a thick coaxial cable. Figure shows a schematic diagram of a 10Base5 implementation



10Base2: Thin Ethernet

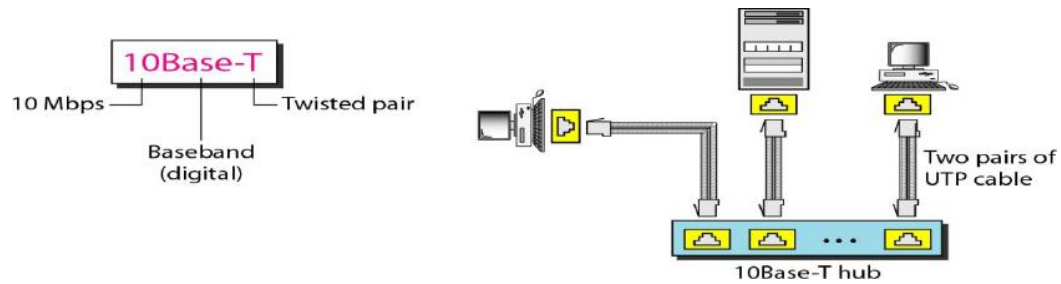
The second implementation is called 10 Base2, **thin** Ethernet, or Cheapernet. 10Base2 also uses a bus topology, but the cable is much thinner and more flexible. Figure shows the schematic diagram of a 10Base2 implementation.



thin coaxial cable is less expensive than thick coaxial. Installation is simpler because the thin coaxial cable is very flexible. However, the length of each segment cannot exceed 185 m (close to 200 m) due to the high level of attenuation in thin coaxial cable.

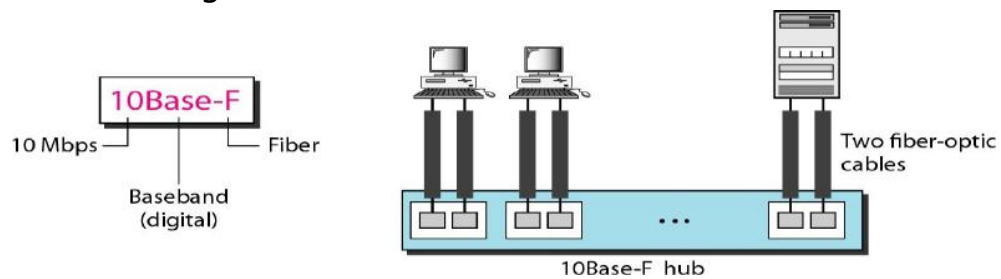
10Base-T: Twisted-Pair Ethernet

The maximum length of the twisted cable here is defined as 100 m, to minimize the effect of attenuation in the twisted cable



10Base-T implementation

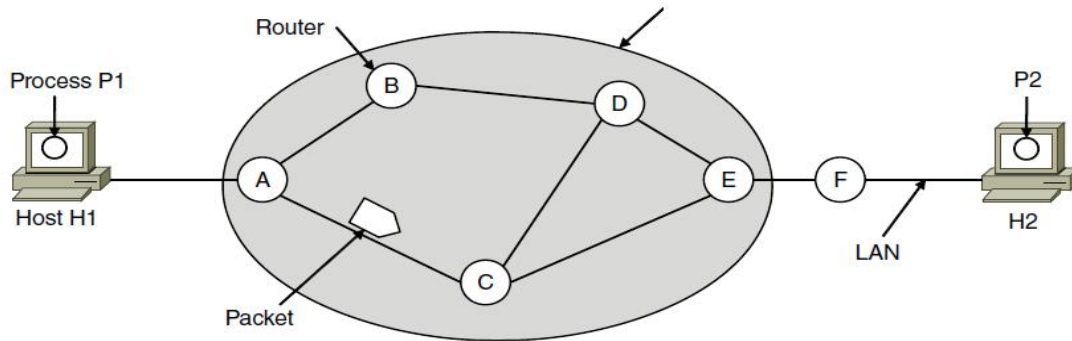
Although there are several types of optical fiber 10-Mbps Ethernet, the most common is called 10Base-F. 10Base-F uses a star topology to connect stations to a hub. The stations are connected to the hub using two fiber-optic cables, as shown in Figure



10Base-F implementation

1. Store-and-forward packet switching
2. Services provided to transport layer
3. Implementation of connectionless service
4. Implementation of connection-oriented service
5. Comparison of virtual-circuit and datagram networks

1 Store-and-forward packet switching



A host with a packet to send transmits it to the nearest router, either on its own LAN or over a point-to-point link to the ISP. The packet is stored there until it has fully arrived and the link has finished its processing by verifying the checksum. Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered. This mechanism is store-and-forward packet switching.

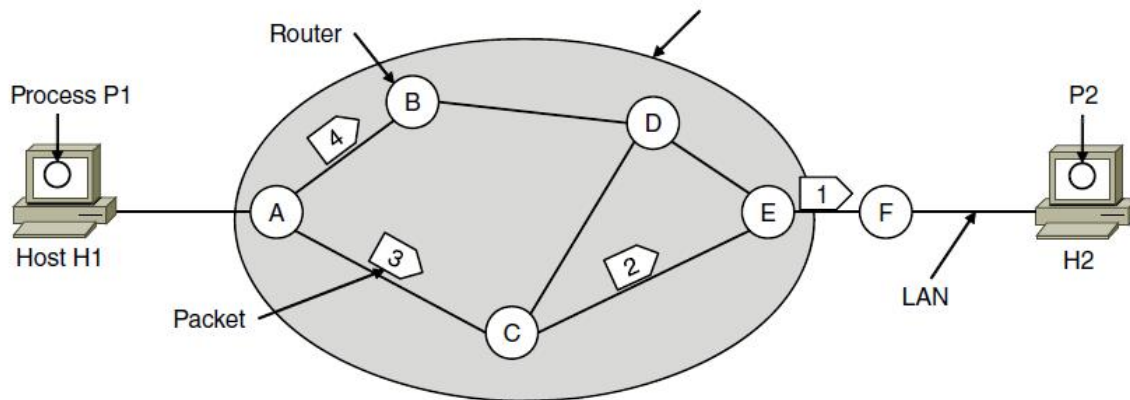
2 Services provided to transport layer

The network layer provides services to the transport layer at the network layer/transport layer interface. The services need to be carefully designed with the following goals in mind:

1. Services independent of router technology.
2. Transport layer shielded from number, type, topology of routers.
3. Network addresses available to transport layer use uniform numbering plan
 - even across LANs and WANs

3 Implementation of connectionless service

If connectionless service is offered, packets are injected into the network individually and routed independently of each other. No advance setup is needed. In this context, the packets



A's table (initially)

A	⊠
B	B
C	C
D	B
E	C
F	C

Dest. Line

A's table (later)

A	⊠
B	B
C	C
D	B
E	D
F	D

C's Table

A	A
B	A
C	⊠
D	E
E	E
F	E

E's Table

A	C
B	D
C	C
D	D
E	⊠
F	F

Let us assume for this example that the message is four times longer than the maximum packet size, so the network layer has to break it into four packets, 1, 2, 3, and 4, and send each of them in turn to router A.

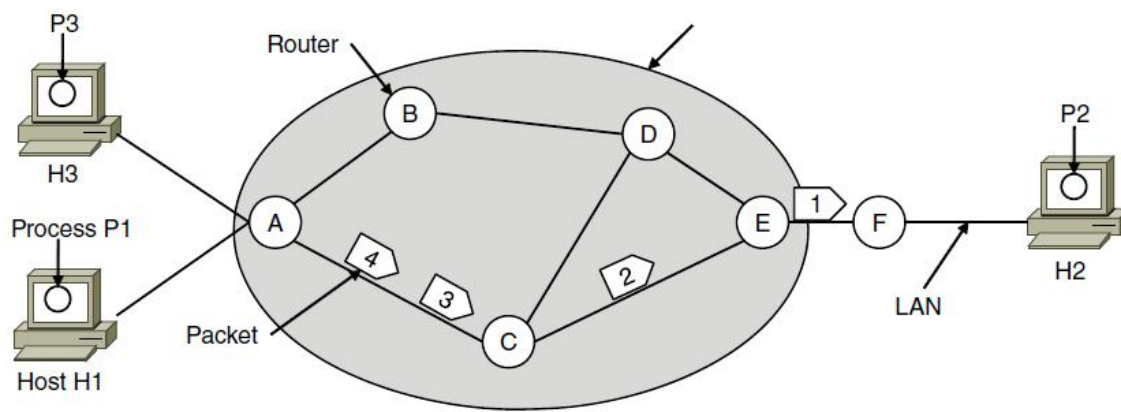
Every router has an internal table telling it where to send packets for each of the possible destinations. Each table entry is a pair(destination and the outgoing line). Only directly connected lines can be used.

A's initial routing table is shown in the figure under the label "initially."

At A, packets 1, 2, and 3 are stored briefly, having arrived on the incoming link. Then each packet is forwarded according to A's table, onto the outgoing link to C within a new frame. Packet 1 is then forwarded to E and then to F.

However, something different happens to packet 4. When it gets to A it is sent to router B, even though it is also destined for F. For some reason (traffic jam along ACE path), A decided to send packet 4 via a different route than that of the first three packets. Router A updated its routing table, as shown under the label "later."

The algorithm that manages the tables and makes the routing decisions is called the **routing algorithm**.



A's table

H1	1	C	1
H3	1	C	2
In		Out	

C's Table

A	1	E	1
A	2	E	2

E's Table

C	1	F	1
C	2	F	2

If connection-oriented service is used, a path from the source router all the way to the destination router must be established before any data packets can be sent. This connection is called a **VC (virtual circuit)**, and the network is called a **virtual-circuit network**.

When a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers. That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works. When the connection is released, the virtual circuit is also terminated. With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.

As an example, consider the situation shown in Figure. Here, host *H1* has established connection 1 with host *H2*. This connection is remembered as the first entry in each of the routing tables. The first line of *A*'s table says that if a packet bearing connection identifier 1 comes in from *H1*, it is to be sent to router *C* and given connection identifier 1. Similarly, the first entry at *C* routes the packet to *E*, also with connection identifier 1.

Now let us consider what happens if *H3* also wants to establish a connection to *H2*. It chooses connection identifier 1 (because it is initiating the connection and this is its only connection) and tells the network to establish the virtual circuit.

cannot do this. For this reason, A assigns a different connection identifier to the outgoing traffic for the second connection. Avoiding conflicts of this kind is why routers need the ability to replace connection identifiers in outgoing packets.

In some contexts, this process is called **label switching**. An example of a connection-oriented network service is **MPLS (Multi Protocol Label Switching)**.

5 Comparison of virtual-circuit and datagram networks

Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Routing Algorithms

The main function of NL (Network Layer) is routing packets from the source machine to the destination machine.

There are two processes inside router:

- One of them handles each packet as it arrives, looking up the outgoing line to use for it in the routing table. This process is forwarding.
- The other process is responsible for filling in and updating the routing tables. That is where the routing algorithm comes into play. This process is routing.

Regardless of whether routes are chosen independently for each packet or only when new connections are established, certain properties are desirable in a routing algorithm **correctness, simplicity, robustness, stability, fairness, optimality**

2) adaptive. (Dynamic Routing)

Nonadaptive algorithm do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get from I to J is computed in advance, off line, and downloaded to the routers when the network is booted. This procedure is sometimes called static routing.

Adaptive algorithm, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well.

Adaptive algorithms differ in

- 1) Where they get their information (e.g., locally, from adjacent routers, or from all routers),
- 2) When they change the routes (e.g., every ΔT sec, when the load changes or when the topology changes), and
- 3) What metric is used for optimization (e.g., distance, number of hops, or estimated transit time).

This procedure is called dynamic routing

Different Routing Algorithms

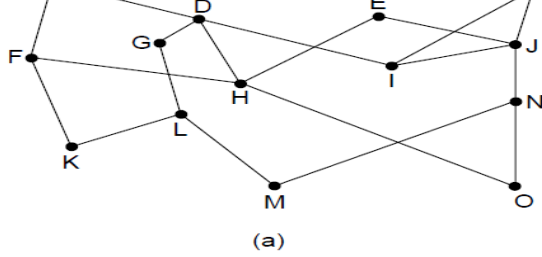
- Optimality principle
- Shortest path algorithm
- Flooding
- Distance vector routing
- Link state routing
- Hierarchical Routing

The Optimality Principle

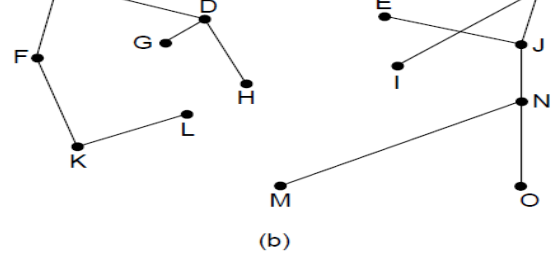
One can make a general statement about optimal routes without regard to network topology or traffic. This statement is known as the optimality principle.

It states that if router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same

As a direct consequence of the optimality principle, we can see that the set of optimal routes from all sources to a given destination form a tree rooted at the destination. Such a tree is called a **sink tree**. The goal of all routing algorithms is to discover and use the sink trees for all routers



(a) A network.



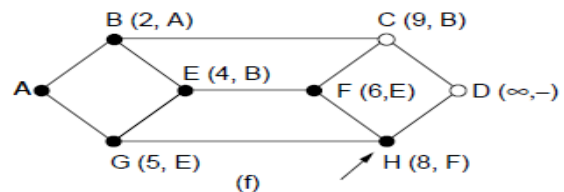
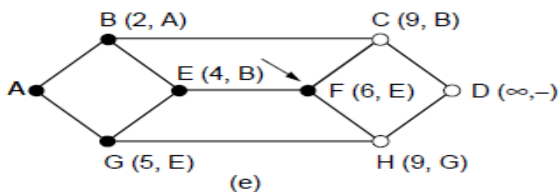
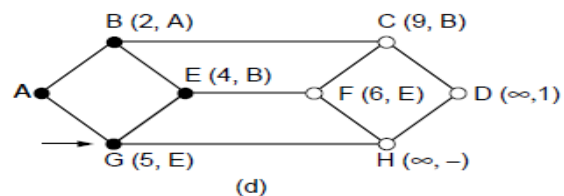
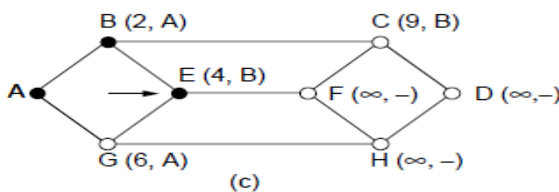
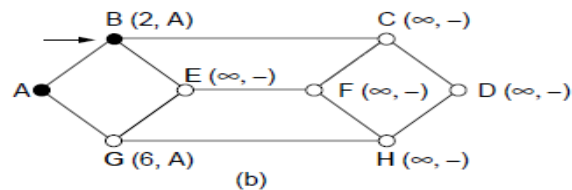
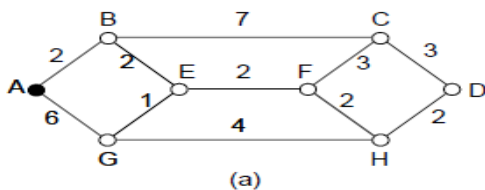
(b) A sink tree for router B.

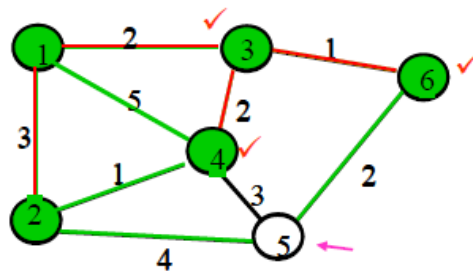
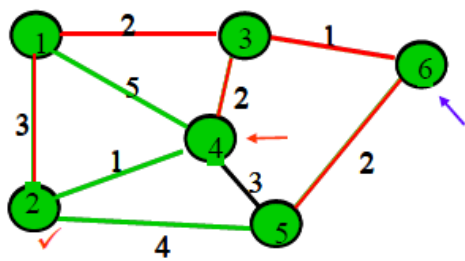
Shortest Path Routing (Dijkstra's)

The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line or link.

To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph

1. Start with the local node (router) as the root of the tree. Assign a cost of 0 to this node and make it the first permanent node.
2. Examine each neighbor of the node that was the last permanent node.
3. Assign a cumulative cost to each node and make it tentative
4. Among the list of tentative nodes
 - a. Find the node with the smallest cost and make it Permanent
 - b. If a node can be reached from more than one route then select the route with the shortest cumulative cost.
5. Repeat steps 2 to 4 until every node becomes permanent





Iteration	Permanent	tentative	D ₂	D ₃	D ₄	D ₅	D ₆
Initial	{1}	{2,3,4}	3	2 ✓	5	∞	∞
1	{1,3}	{2,4,6}	3 ✓	2	4	∞	3
2	{1,2,3}	{4,6,5}	3	2	4	7	3 ✓
3	{1,2,3,6}	{4,5}	3	2	4 ✓	5	3
4	{1,2,3,4,6}	{5}	3	2	4	5 ✓	3
5	{1,2,3,4,5,6}	{}	3	2	4	5	3

Flooding

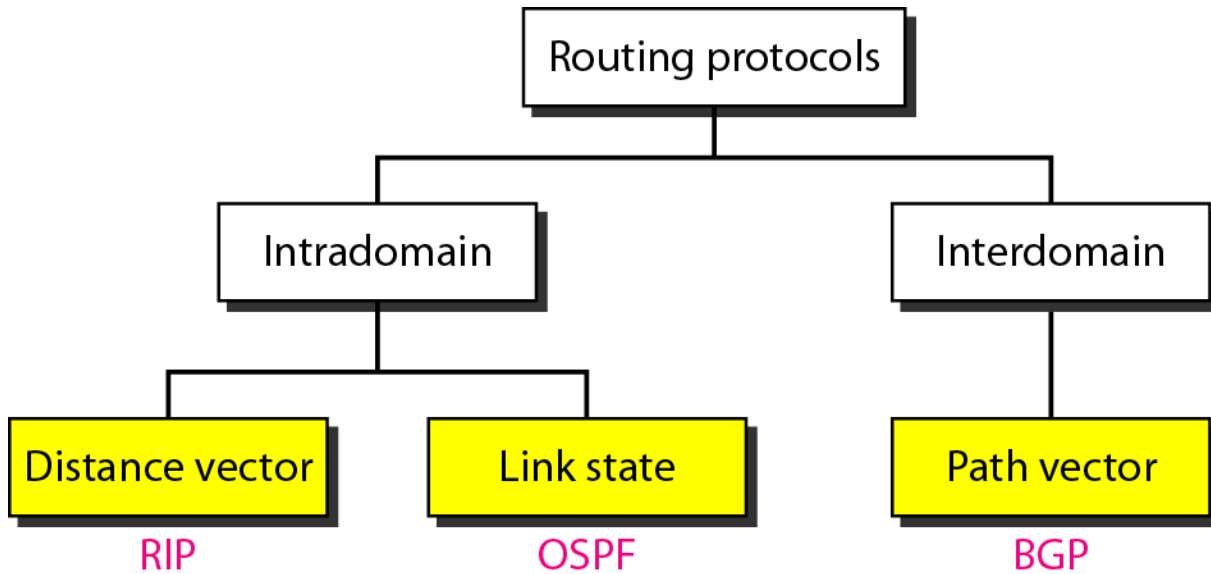
- Another static algorithm is flooding, in which every incoming packet is sent out on every outgoing line except the one it arrived on.
- Flooding obviously generates vast numbers of duplicate packets, in fact, an infinite number unless some measures are taken to damp the process.
- One such measure is to have a hop counter contained in the header of each packet, which is decremented at each hop, with the packet being discarded when the counter reaches zero. Ideally, the hop counter should be initialized to the length of the path from source to destination.
- A variation of flooding that is slightly more practical is **selective flooding**. In this algorithm the routers do not send every incoming packet out on every line, only on those lines that are going approximately in the right direction.
- Flooding is not practical in most applications.

Intra- and Inter domain Routing

An autonomous system (AS) is a group of networks and routers under the authority of a single administration.

Routing inside an autonomous system is referred to as intra domain routing. (DISTANCE VECTOR, LINK STATE)

routing inside the autonomous system. However, only one inter domain routing protocol handles routing between autonomous systems.



Distance Vector Routing

In distance vector routing, the least-cost route between any two nodes is the route with minimum distance. In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node.

Mainly 3 things in this

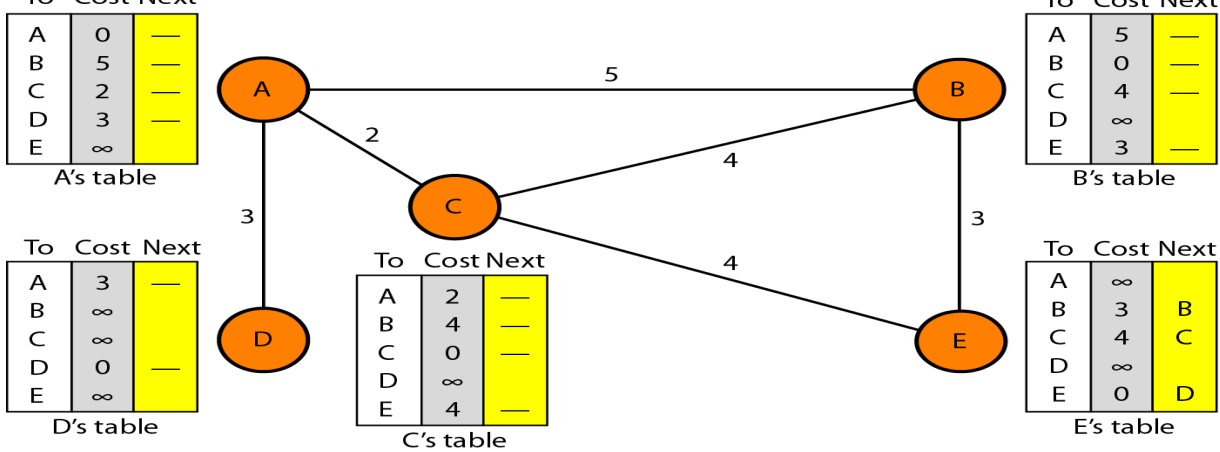
Initialization

Sharing

Updating

Initialization

Each node can know only the distance between itself and its immediate neighbors, those directly connected to it. So for the moment, we assume that each node can send a message to the immediate neighbors and find the distance between itself and these neighbors. Below fig shows the initial tables for each node. The distance for any entry that is not a neighbor is marked as infinite (unreachable).



Sharing

The whole idea of distance vector routing is the sharing of information between neighbors. Although node A does not know about node E, node C does. So if node C shares its routing table with A, node A can also know how to reach node E. On the other hand, node C does not know how to reach node D, but node A does. If node A shares its routing table with node C, node C also knows how to reach node D. In other words, nodes A and C, as immediate neighbors, can improve their routing tables if they help each other.

NOTE: In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change

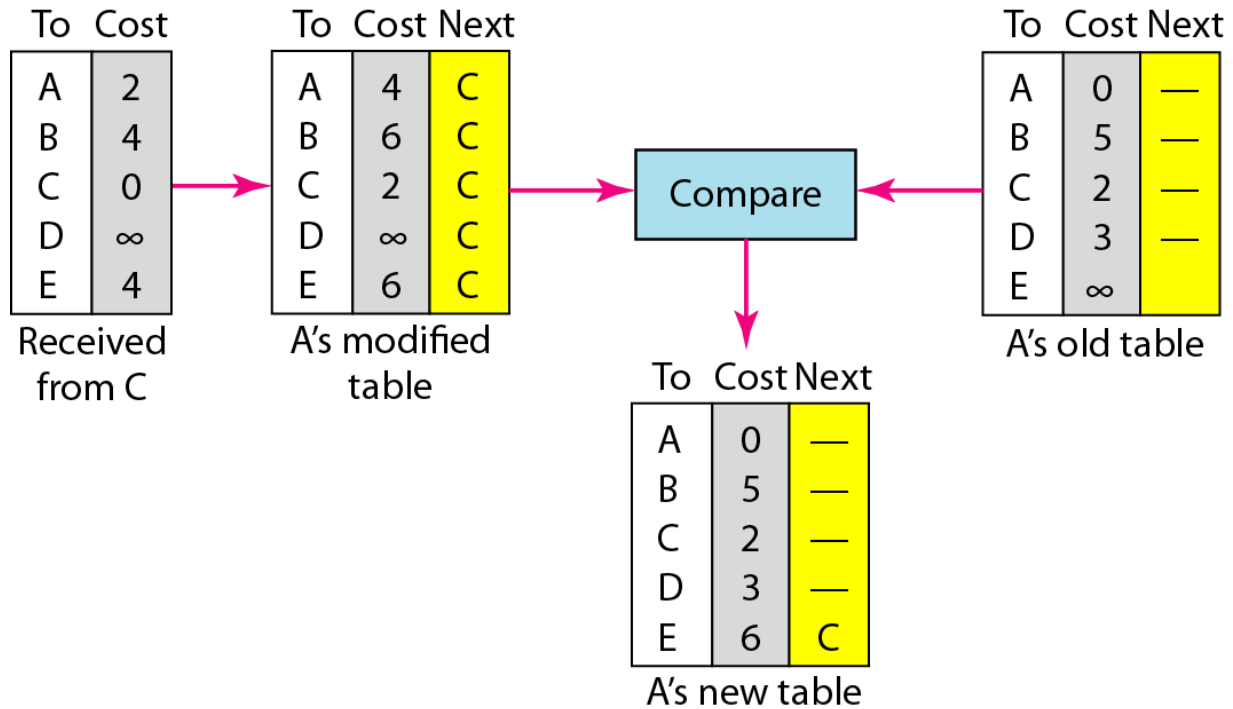
Updating

When a node receives a two-column table from a neighbor, it needs to update its routing table. Updating takes three steps:

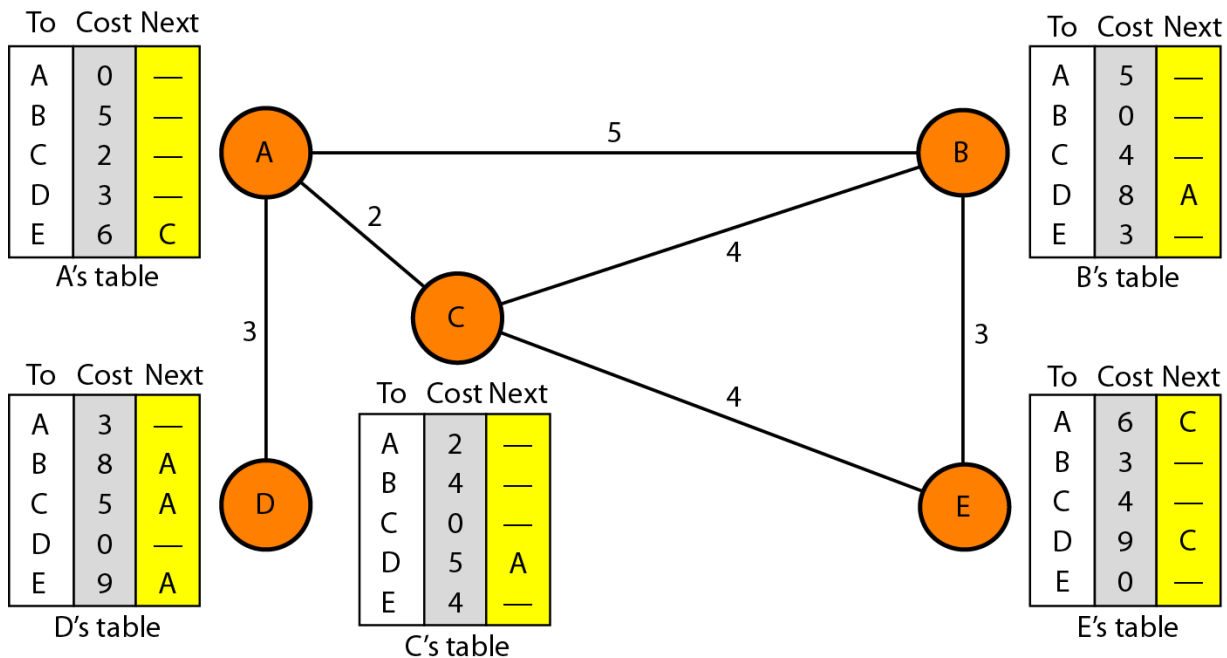
1. The receiving node needs to add the cost between itself and the sending node to each value in the second column. ($x+y$)
2. If the receiving node uses information from any row. The sending node is the next node in the route.
3. The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.
 - a. If the next-node entry is different, the receiving node chooses the row with the smaller cost. If there is a tie, the old one is kept.
 - b. If the next-node entry is the same, the receiving node chooses the new row.

distance of infinity. Node A must not ignore this value even though its old entry is smaller. The old route does not exist anymore. The new route has a distance of infinity.

Updating in distance vector routing



Final Diagram



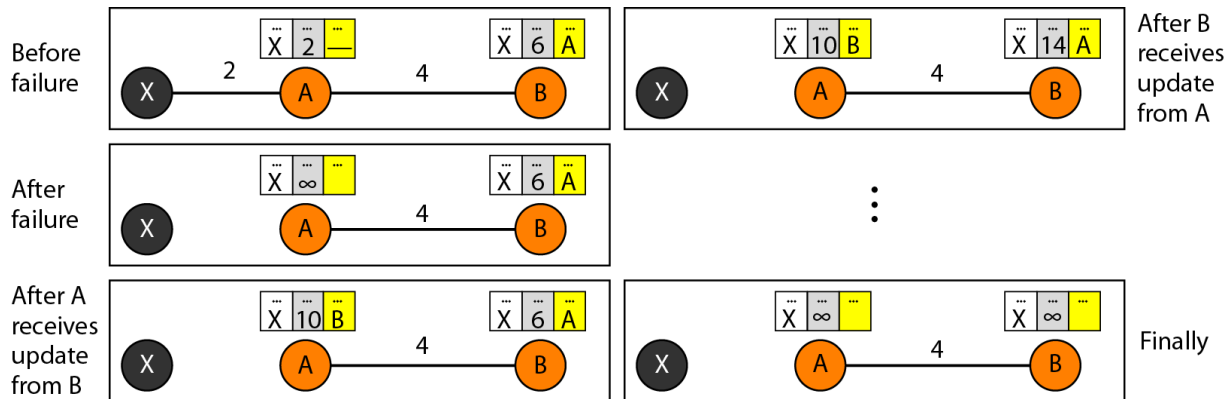
its immediate neighbors? The table is sent both periodically and when there is a change in the table.

Periodic Update A node sends its routing table, normally every 30 s, in a periodic update. The period depends on the protocol that is using distance vector routing.

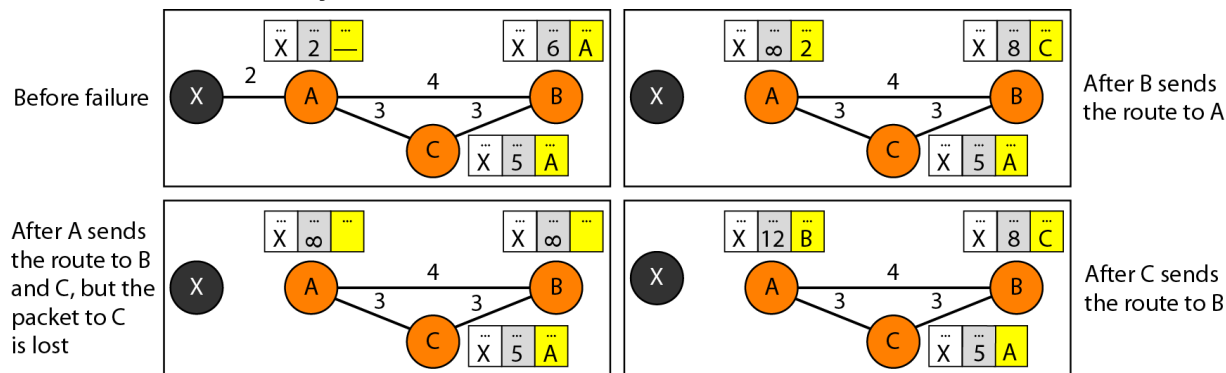
Triggered Update A node sends its two-column routing table to its neighbors anytime there is a change in its routing table. This is called a triggered update. The change can result from the following.

1. A node receives a table from a neighbor, resulting in changes in its own table after updating.
2. A node detects some failure in the neighboring links which results in a distance change to infinity.

Two-node instability



Three-node instability



SOLUTIONS FOR INSTABILITY

1. **Defining Infinity:** redefine infinity to a smaller number, such as 100. For our previous scenario, the system will be stable in less than 20 updates. As a matter of fact, most implementations of the distance vector protocol define the distance between each node to

2. **Split Horizon:** In this strategy, instead of flooding the table through each interface, each node sends **only part of its table** through each interface. If, according to its table, node B thinks that the optimum route to reach X is via A, it does not need to advertise this piece of information to A; the information has come from A (A already knows). Taking information from node A, modifying it, and sending it back to node A creates the confusion. In our scenario, node B eliminates the last line of its routing table before it sends it to A. In this case, node A keeps the value of infinity as the distance to X. Later when node A sends its routing table to B, node B also corrects its routing table. The system becomes stable after the first update: both node A and B know that X is not reachable.

3. **Split Horizon and Poison Reverse** Using the split horizon strategy has one drawback. Normally, the distance vector protocol uses a timer, and if there is no news about a route, the node deletes the route from its table. When node B in the previous scenario eliminates the route to X from its advertisement to A, node A cannot guess that this is due to the split horizon strategy (the source of information was A) or because B has not received any news about X recently. The split horizon strategy can be combined with the poison reverse strategy. Node B can still advertise the value for X, but if the source of information is A, it can replace the distance with infinity as a warning: "Do not use this value; what I know about this route comes from you."

The Count-to-Infinity Problem

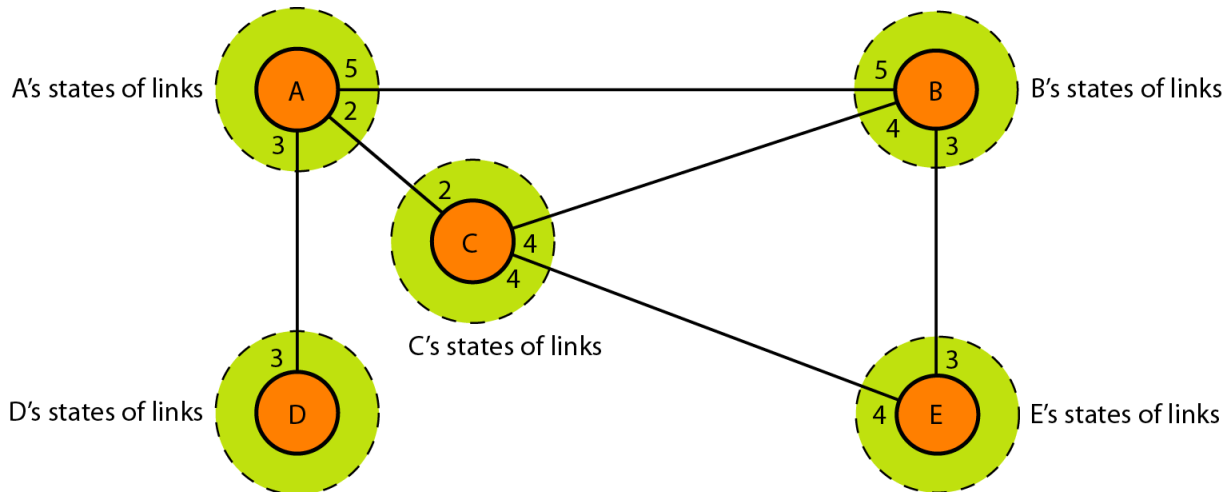
A	B	C	D	E	
•	•	•	•	•	Initially
	1	•	•	•	After 1 exchange
	1	2	•	•	After 2 exchanges
	1	2	3	•	After 3 exchanges
	1	2	3	4	After 4 exchanges

(a)

A	B	C	D	E	
•	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
		⋮			
	•	•	•	•	

(b)

topology is not clear, each node has partial knowledge: it knows the state (type, condition, and cost) of its links. **In other words, the whole topology can be compiled from the partial knowledge of each node**



Building Routing Tables

1. Creation of the states of the links by each node, called the link state packet (LSP).
2. Dissemination of LSPs to every other router, called **flooding, in an efficient and reliable way**.
3. Formation of a shortest path tree for each node.
4. Calculation of a routing table based on the shortest path tree

- I. **Creation of Link State Packet (LSP)** A link state packet can carry a large amount of information. For the moment, we assume that it carries a minimum amount of data: the node identity, the list of links, a sequence number, and age. The first two, node identity and the list of links, are needed to make the topology. The third, sequence number, facilitates flooding and distinguishes new LSPs from old ones. The fourth, age, prevents old LSPs from remaining in the domain for a long time.

LSPs are generated on two occasions:

1. When there is a change in the topology of the domain
2. on a periodic basis: The period in this case is much longer compared to distance vector.

The timer set for periodic dissemination is normally in the range of **60 min or 2 h** based on the implementation. A longer period ensures that flooding does not create too much traffic on the network.

- II. **Flooding of LSPs:** After a node has prepared an LSP, it must be disseminated to all other nodes, not only to its neighbors. The process is called flooding and based on the following
 1. The creating node sends a copy of the LSP out of each interface

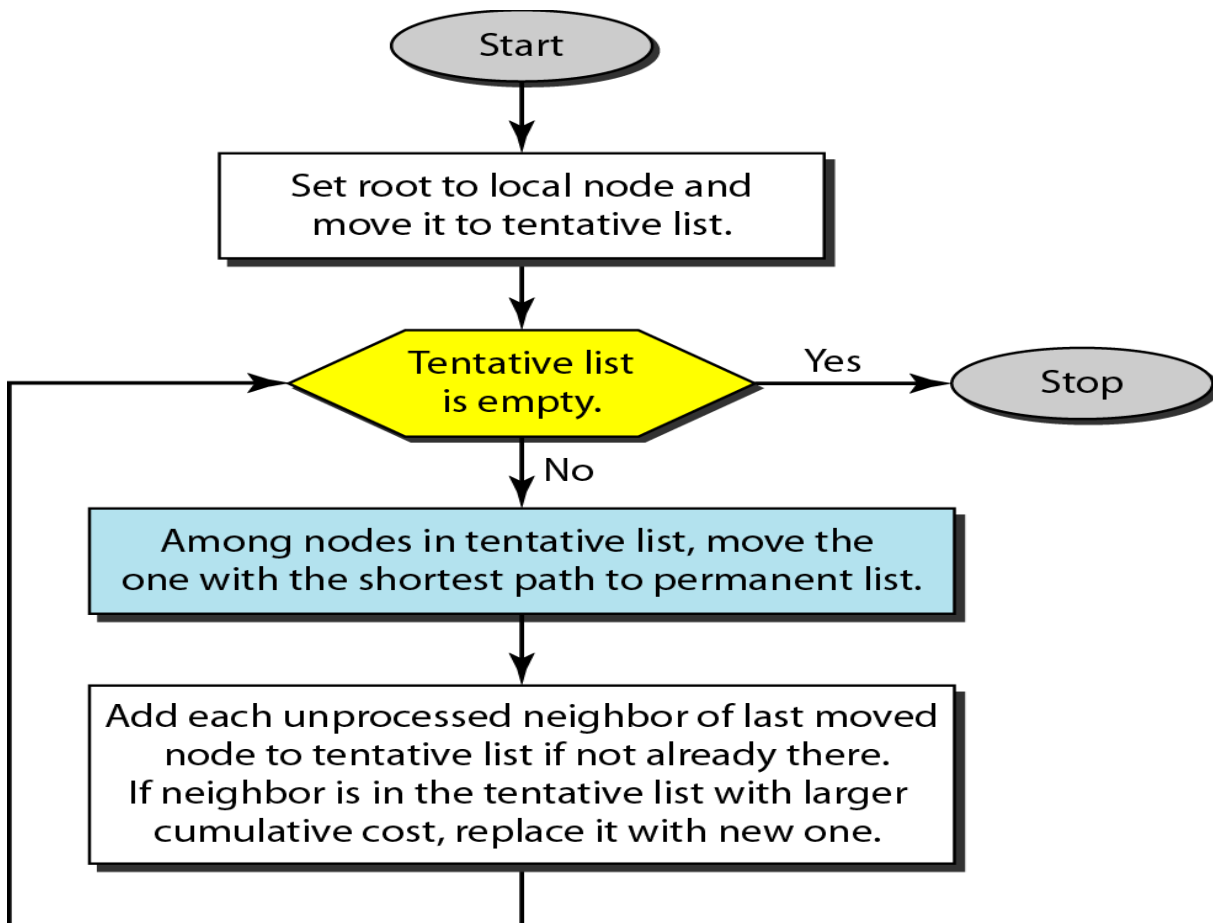
it discards the LSP. If it is newer, the node does the following:

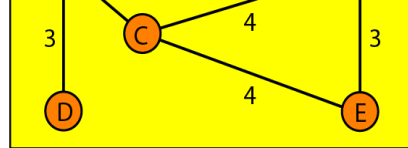
- a. It discards the old LSP and keeps the new one.
- b. It sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the domain (where a node has only one interface).

III. Formation of Shortest Path Tree: Dijkstra Algorithm

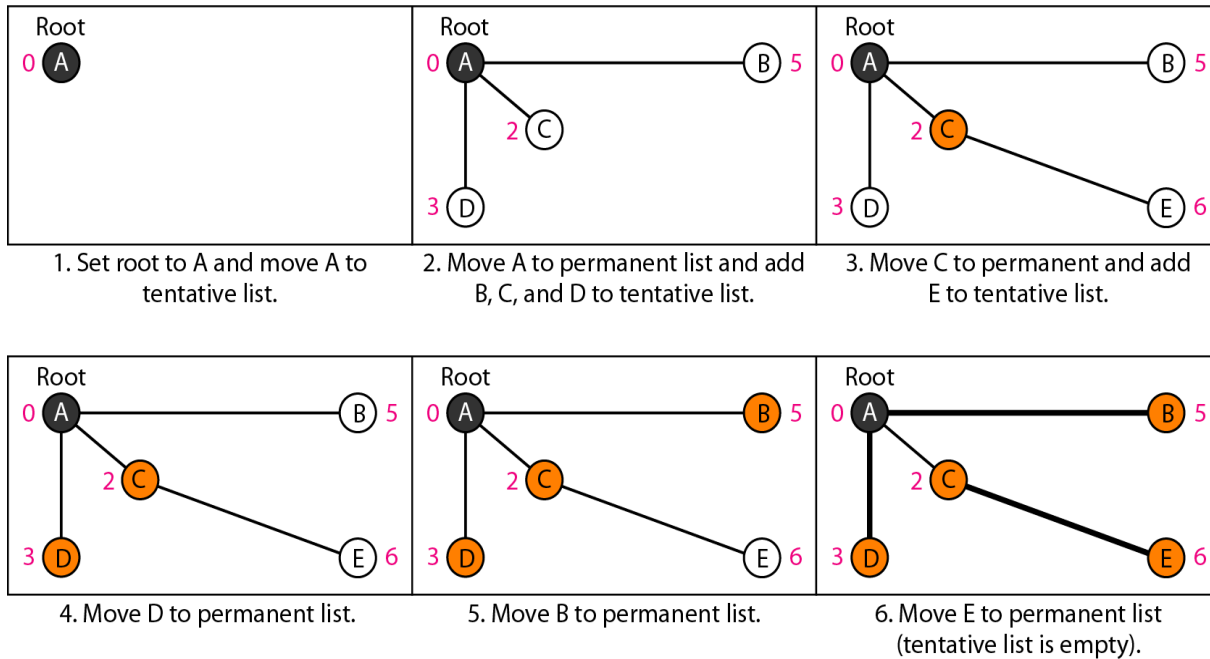
A shortest path tree is a tree in which the path between the root and every other node is the shortest.

The Dijkstra algorithm creates a shortest path tree from a graph. The algorithm divides the nodes into two sets: **tentative and permanent**. It finds the neighbors of a current node, makes them tentative, examines them, and if they pass the criteria, makes them permanent.





Topology



IV. Calculation of a routing table

routing table for node A

<i>Node</i>	<i>Cost</i>	<i>Next Router</i>
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

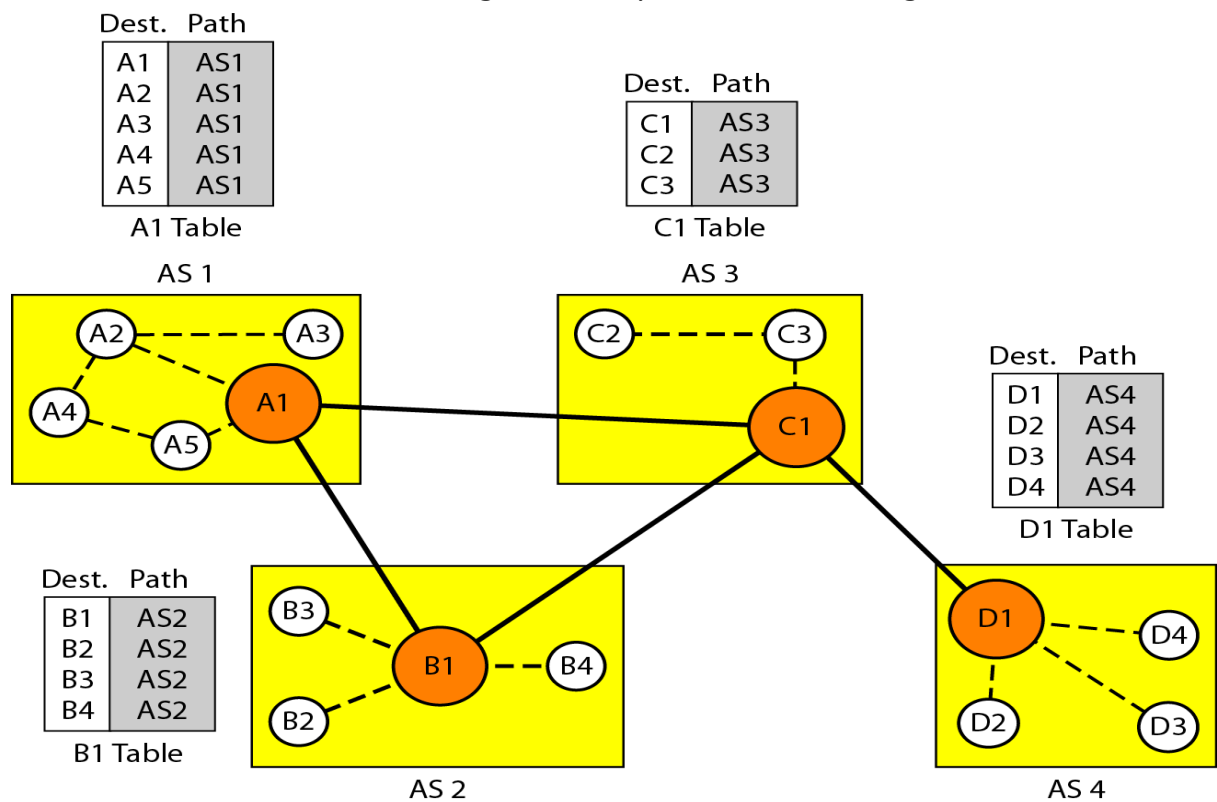
Path Vector Routing

Distance vector and link state routing are both intra domain routing protocols. They can be used inside an autonomous system, but not between autonomous systems. These two protocols are not suitable for inter domain routing mostly because of scalability. Both of these routing protocols become intractable when the domain of operation becomes large. **Distance vector routing is subject to instability** in the domain of operation. **Link state routing needs a**

Path vector routing proved to be useful for inter domain routing. The principle of path vector routing is similar to that of distance vector routing. **In path vector routing, we assume that there is one node** (there can be more, but one is enough for our conceptual discussion) **in each AS** that acts on behalf of the entire AS. Let us call it the **speaker node**. The speaker node in an AS creates a routing table and advertises it to speaker nodes in the neighboring ASs. The idea is the same as for distance vector routing except that only speaker nodes in each AS can communicate with each other. However, what is advertised is different. A speaker node advertises the path, not the metric of the nodes, in its autonomous system or other autonomous systems

Initialization

Initial routing tables in path vector routing



Sharing

Just as in distance vector routing, in path vector routing, a speaker in an autonomous system shares its table with immediate neighbors. In Figure, node A1 shares its table with nodes B1

Dest.	Path	Dest.	Path	Dest.	Path	Dest.	Path
A1	AS1	A1	AS2-AS1	A1	AS3-AS1	A1	AS4-AS3-AS1
...		
A5	AS1	A5	AS2-AS1	A5	AS3-AS1	A5	AS4-AS3-AS1
B1	AS1-AS2	B1	AS2	B1	AS3-AS2	B1	AS4-AS3-AS2
...
B4	AS1-AS2	B4	AS2	B4	AS3-AS2	B4	AS4-AS3-AS2
C1	AS1-AS3	C1	AS2-AS3	C1	AS3	C1	AS4-AS3
...
C3	AS1-AS3	C3	AS2-AS3	C3	AS3	C3	AS4-AS3
D1	AS1-AS2-AS4	D1	AS2-AS3-AS4	D1	AS3-AS4	D1	AS4
...
D4	AS1-AS2-AS4	D4	AS2-AS3-AS4	D4	AS3-AS4	D4	AS4

A1 Table

B1 Table

C1 Table

D1 Table

Updating When a speaker node receives a two-column table from a neighbor, it updates its own table by adding the nodes that are not in its routing table and adding its own autonomous system and the autonomous system that sent the table. After a while each speaker has a table and knows how to reach each node in other Ass

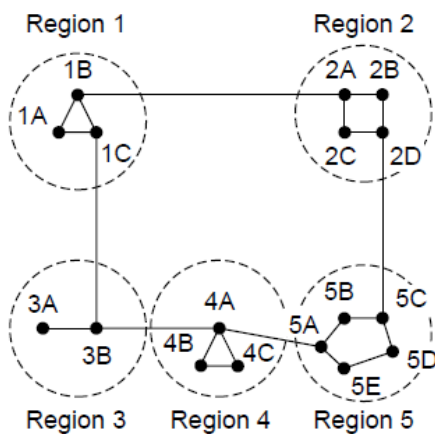
- Loop prevention.** The instability of distance vector routing and the creation of loops can be avoided in path vector routing. When a router receives a message, it checks to see if its AS is in the path list to the destination. If it is, looping is involved and the message is ignored.
 - Policy routing.** Policy routing can be easily implemented through path vector routing. When a router receives a message, it can check the path. If one of the AS listed in the path is against its policy, it can ignore that path and that destination. It does not update its routing table with this path, and it does not send this message to its neighbors.
 - Optimum path.** What is the optimum path in path vector routing? We are looking for a path to a destination that is the best for the organization that runs the AS. One system may use RIP, which defines hop count as the metric; another may use OSPF with minimum delay defined as the metric. In our previous figure, each AS may have more than one path to a destination. For example, a path from AS4 to AS1 can be AS4-AS3-AS2-AS1, or it can be AS4-AS3-AS1. For the tables, **we chose the one that had the smaller number of ASs**, but this is not always the case. Other criteria, such as security, safety, and reliability, can also be applied
-

memory consumed by ever-increasing tables, but more CPU time is needed to scan them and more bandwidth is needed to send status reports about them.

At a certain point, the network may grow to the point where it is no longer feasible for every router to have an entry for every other router, so the routing will have to be done hierarchically, as it is in the telephone network.

When hierarchical routing is used, the routers are divided into what we will call regions. Each router knows all the details about how to route packets to destinations within its own region but knows nothing about the internal structure of other regions.

For huge networks, a two-level hierarchy may be insufficient; it may be necessary to group the regions into clusters, the clusters into zones, the zones into groups, and so on, until we run out of names for aggregations



(a)

Full table for 1A		
Dest.	Line	Hops
1A	—	—
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A		
Dest.	Line	Hops
1A	—	—
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

When a single network becomes very large, an interesting question is “how many levels should the hierarchy have?”

For example, consider a network with 720 routers. If there is no hierarchy, each router needs 720 routing table entries.

If the network is partitioned into 24 regions of 30 routers each, each router needs 30 local entries plus 23 remote entries for a total of 53 entries.

own cluster, and 7 entries for distant clusters, for a total of 25 entries

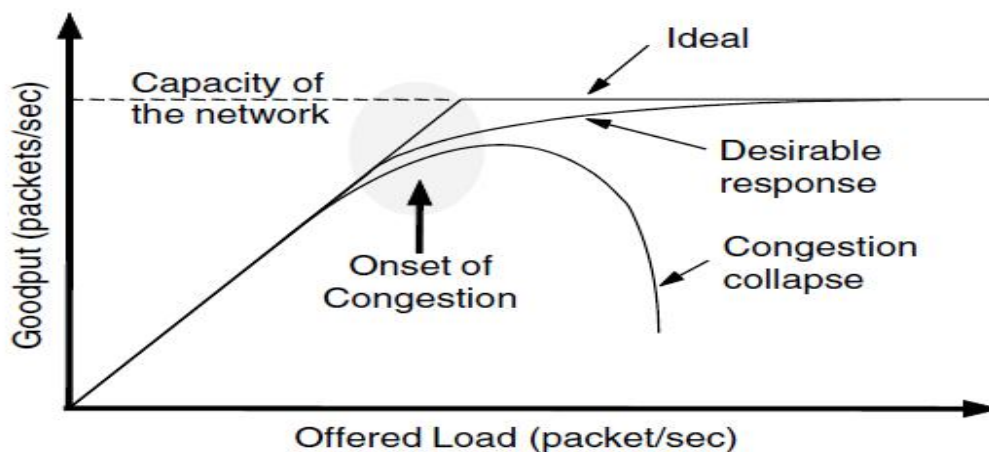
Kamoun and Kleinrock (1979) discovered that the optimal number of levels for an N router network is $\ln N$, requiring a total of $e \ln N$ entries per router

CONGESTION CONTROL ALGORITHMS

Too many packets present in (a part of) the network causes packet delay and loss that degrades performance. This situation is called **congestion**.

The network and transport layers share the responsibility for handling congestion. Since congestion occurs within the network, it is the network layer that directly experiences it and must ultimately determine what to do with the excess packets.

However, the most effective way to control congestion is to reduce the load that the transport layer is placing on the network. This requires the network and transport layers to work together. In this chapter we will look at the network aspects of congestion.



When too much traffic is offered, congestion sets in and performance degrades sharply

Above Figure depicts the onset of congestion. When the number of packets hosts send into the network is well within its carrying capacity, the number delivered is proportional to the number sent. If twice as many are sent, twice as many are delivered. However, as the offered load approaches the carrying capacity, bursts of traffic occasionally fill up the buffers inside routers and some packets are lost. These lost packets consume some of the capacity, so the number of delivered packets falls below the ideal curve. The network is now congested. Unless the network is well designed, it may experience a **congestion collapse**

It is a global issue, involving the behavior of all the hosts and routers.

Flow control, in contrast, relates to the traffic between a particular sender and a particular receiver. Its job is to make sure that a fast sender cannot continually transmit data faster than the receiver is able to absorb it.

To see the difference between these two concepts, consider a network made up of 100-Gbps fiber optic links on which a supercomputer is trying to force feed a large file to a personal computer that is capable of handling only 1 Gbps. Although there is no congestion (the network itself is not in trouble), flow control is needed to force the supercomputer to stop frequently to give the personal computer a chance to breathe.

At the other extreme, consider a network with 1-Mbps lines and 1000 large computers, half of which are trying to transfer files at 100 kbps to the other half. Here, the problem is not that of fast senders overpowering slow receivers, but that the total offered traffic exceeds what the network can handle.

The reason congestion control and flow control are often confused is that the best way to handle both problems is to get the host to slow down. Thus, a host can get a “slow down” message either because the receiver cannot handle the load or because the network cannot handle it.

Several techniques can be employed. These include:

1. Warning bit
2. Choke packets
3. Load shedding
4. Random early discard
5. Traffic shaping

The first 3 deal with congestion detection and recovery. The last 2 deal with congestion avoidance

Warning Bit

1. A special bit in the packet header is set by the router to warn the source when congestion is detected.
2. The bit is copied and piggy-backed on the ACK and sent to the sender.
3. The sender monitors the number of ACK packets it receives with the warning bit set and adjusts its transmission rate accordingly.

2. A choke packet is a control packet generated at a congested node and transmitted to restrict traffic flow.
3. The source, on receiving the choke packet must reduce its transmission rate by a certain percentage.
4. An example of a choke packet is the ICMP Source Quench Packet.

Hop-by-Hop Choke Packets

1. Over long distances or at high speeds choke packets are not very effective.
2. A more efficient method is to send to choke packets hop-by-hop.
3. This requires each hop to reduce its transmission even before the choke packet arrive at the source

Load Shedding

1. When buffers become full, routers simply discard packets.
2. Which packet is chosen to be the victim depends on the application and on the error strategy used in the data link layer.
3. For a file transfer, for, e.g. cannot discard older packets since this will cause a gap in the received data.
4. For real-time voice or video it is probably better to throw away old data and keep new packets.
5. Get the application to mark packets with discard priority.

Random Early Discard (RED)

1. This is a proactive approach in which the router discards one or more packets *before* the buffer becomes completely full.
2. Each time a packet arrives, the RED algorithm computes the average queue length, **avg**.
3. If *avg* is lower than some lower threshold, congestion is assumed to be minimal or non-existent and the packet is queued.
4. If *avg* is greater than some upper threshold, congestion is assumed to be serious and the packet is discarded.
5. If *avg* is between the two thresholds, this might indicate the onset of congestion. The probability of congestion is then calculated.

network.

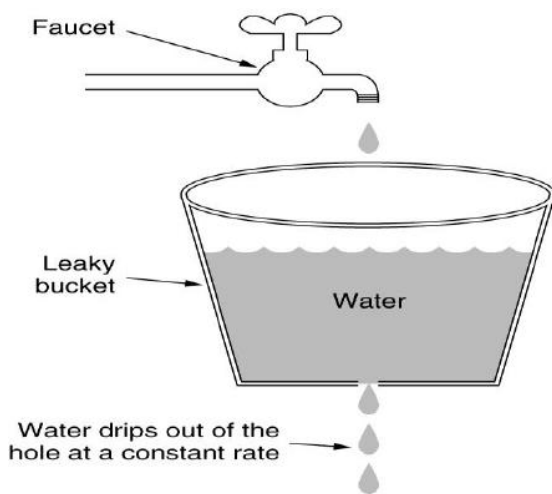
2. Traffic shaping controls the *rate* at which packets are sent (not just how many). Used in ATM and Integrated Services networks.
3. At connection set-up time, the sender and carrier negotiate a traffic pattern (shape).

Two traffic shaping algorithms are:

Leaky Bucket

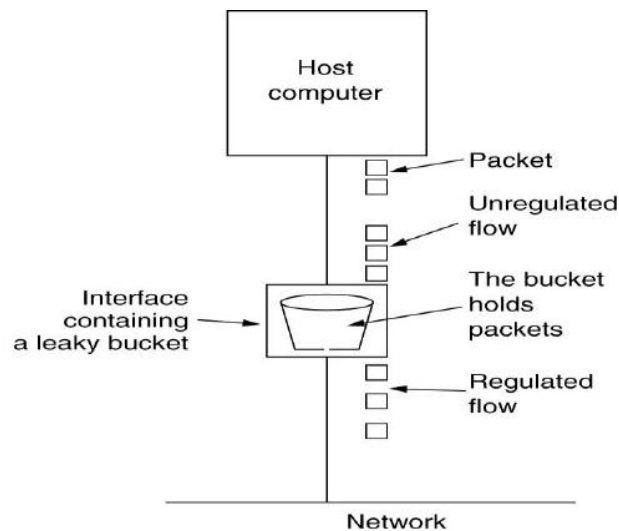
Token Bucket

The **Leaky Bucket Algorithm** used to control rate in a network. It is implemented as a single-server queue with constant service time. If the bucket (buffer) overflows then packets are discarded.



(a)

(a) A leaky bucket with water.

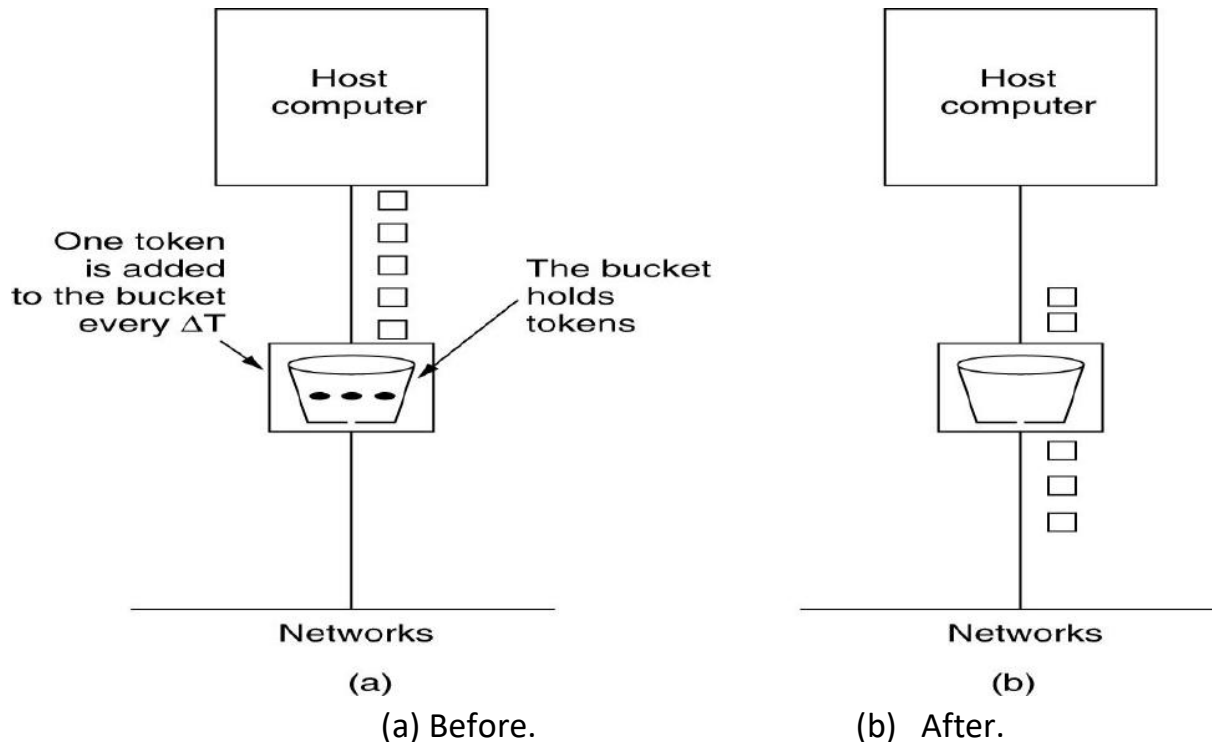


(b)

(b) a leaky bucket with packets.

1. The leaky bucket enforces a constant output rate (average rate) regardless of the burstiness of the input. Does nothing when input is idle.
2. The host injects one packet per clock tick onto the network. This results in a uniform flow of packets, smoothing out bursts and reducing congestion.
3. When packets are the same size (as in ATM cells), the one packet per tick is okay. For variable length packets though, it is better to allow a fixed number of bytes per tick. E.g. 1024 bytes per tick will allow one 1024-byte packet or two 512-byte packets or four 256-byte packets on 1 tick

1. In contrast to the LB, the Token Bucket Algorithm, allows the output rate to vary, depending on the size of the burst.
2. In the TB algorithm, the bucket holds tokens. To transmit a packet, the host must capture and destroy one token.
3. Tokens are generated by a clock at the rate of one token every Δt sec.
4. Idle hosts can capture and save up tokens (up to the max. size of the bucket) in order to send larger bursts later.



Leaky Bucket vs. Token Bucket

1. LB discards packets; TB does not. TB discards tokens.
2. With TB, a packet can only be transmitted if there are enough tokens to cover its length in bytes.
3. LB sends packets at an average rate. TB allows for large bursts to be sent faster by speeding up the output.
4. TB allows saving up tokens (permissions) to send large bursts. LB does not allow saving.

the transport layer that does the work is called the **transport entity**.

header

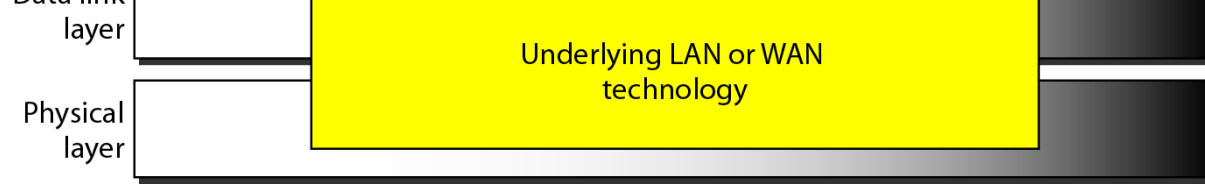




23.7

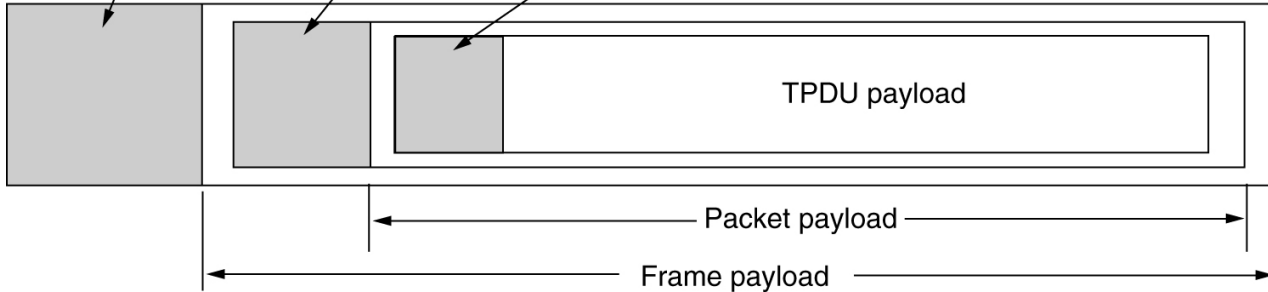


23.8



The primitives for a simple transport
service.

the simplest form, either party can do a (blocking) RECEIVE to wait for the other party to do a SEND. When the segment arrives, the receiver is unblocked. It can then process the segment and send a reply. As long as both sides can keep track of whose turn it is to send, this scheme works fine.



A state diagram for a simple connection management scheme. Transitions labeled in *italics* are caused by packet arrivals. The solid lines show the client's state sequence. The dashed lines show the server's state sequence.

- (a) Environment of the data link layer.
- (b) Environment of the transport layer.

packets can sometimes be disastrous and can require the use of special protocols to correctly transport information.

4. Buffering and flow control are needed in both layers, but the presence in the transport layer of a large and varying number of connections with bandwidth that fluctuates as the connections compete with each other may require a different approach than we need in the data link layer.



TSAPs, NSAPs and transport
connections.

special process called a **portmapper**

sequence number, y : Finally, host Y acknowledges host X 's choice of an initial sequence number in the first data segment that it sends.

- (b) Old CONNECTION REQUEST appearing out of nowhere.
- (c) Duplicate CONNECTION REQUEST and duplicate ACK.

segment arrives at host 2, the fact that *z has been* acknowledged rather than *y tells host 2 that this, too, is an old duplicate*. The important thing to realize here is that there is no combination of old segments that can cause the protocol to fail and have a connection set up by accident when no one wants it.



a disconnect
request



Abrupt disconnection with loss of data.

Symmetric release does the job when each process has a fixed amount of data to send and clearly knows when it has sent it. One can envision a protocol in which host 1 says “I am done. Are you done too?” If host 2 responds: “I am done too. Goodbye, the connection can be safely released.”



Four protocol scenarios for releasing a connection. (a) Normal case of a three-way handshake. (b) final ACK lost.

(c)

(d)

(c) Response lost. (d) Response lost and subsequent DRs lost.

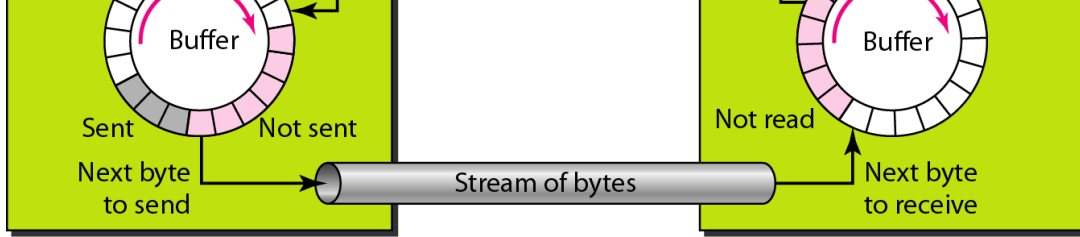
no segments are lost and all segments are delivered correctly and on time.

Last scenario, Fig.(d), is the same as Fig. (c) except that now we assume all the repeated attempts to retransmit the DR also fail due to lost segments. *After N retries, the sender just gives up and releases the connection. Meanwhile, the receiver times out and*

Error Control

25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call





area contains empty chambers to be filled by bytes received from the network. The colored sections contain received bytes that can be read by the receiving process. When a byte is read by the receiving process, the chamber is recycled and added to the pool of empty chambers.

the send operation is transparent to the receiving process. Later we will see that segments may be received out of order, lost, or corrupted and resent. All these are handled by TCP with the receiving process unaware of any activities. Above fig shows how segments are created from the bytes in the buffers

23.34

Acknowledgment Number The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive. The acknowledgment number is cumulative.

is also determined by the level of congestion in the network

Options and Padding

23.38

can be piggybacked together.

Header length. This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field can be between 5 ($5 \times 4 = 20$) and 15 ($15 \times 4 = 60$).

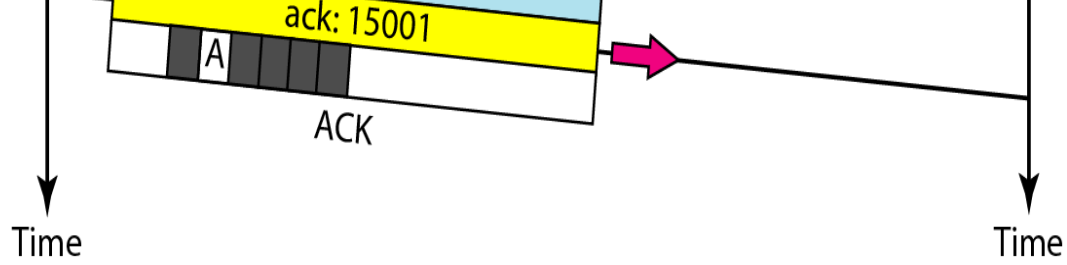
Reserved. This is a 6-bit field reserved for future use.

Control. This field defines 6 different control bits or flags as shown in Figure. One or more of these bits can be set at a time.

checksum for TCP follows the same procedure as the one described for UDP. However, the inclusion of the checksum in the UDP datagram is optional, whereas the inclusion of the checksum for TCP is mandatory. The same pseudoheader, serving the same purpose, is added to the segment. For the TCP pseudoheader, the value for the protocol field is 6.

acknowledgment number field. Note that the sequence number in this segment is the same as the one in the SYN segment; the ACK segment does not consume any sequence numbers.

NOTE: An ACK segment, if carrying no data, consumes no sequence number

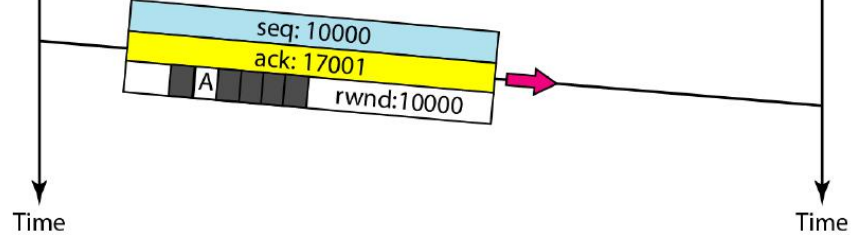


1 Some have imposed a limit on connection requests during a specified period of time.

2 Others filter out datagrams coming from unwanted source addresses.

3 One recent strategy is to postpone resource allocation until the entire connection is set up using what is called a cookie

are received.

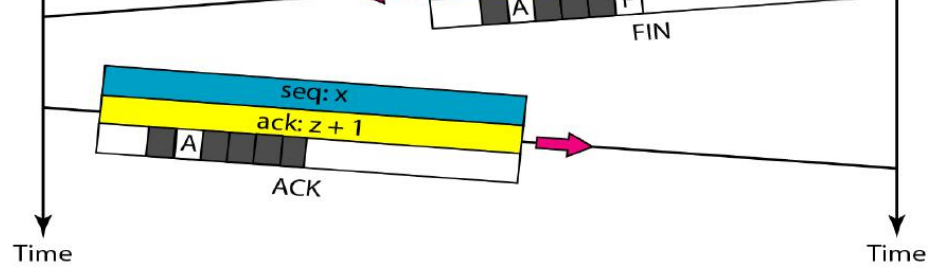


to announce the closing of the connection in the other direction. This segment can also contain the last chunk of data from the server. If it does not carry data, it consumes only one sequence number.

NOTE: The FIN +ACK segment consumes one sequence number if it does not carry data.

needs time for setting, its outboard direction must remain open.





the data link layer was of fixed size

end can accept before its buffer overflows and data are discarded.

The congestion window is a value determined by the network to avoid congestion



Window management in TCP

data until another 2 KB came in, to be able to transmit a segment with a 4-KB payload. This freedom can be used to improve performance

Client □ **Server:** Send typed character

Server □ **Client:** Echo of character (or user output) and
acknowledgement for first packet

Client □ **Server:** Acknowledgement for second packet

41-byte packets containing 1 byte of data) is still operating inefficiently. A way to reduce this usage is known as **Nagle's algorithm (Nagle, 1984)**.

Because of these problems, Nagle's algorithm can be disabled (which is called the *TCP NODELAY option*).



The receiver will buffer the data until it can be passed up to the application in order (handling out of order segments)

Cumulative acknowledgements

corrupted segment. If the segment is corrupted, it is discarded by the destination TCP and is considered as lost. TCP uses a 16-bit checksum that is mandatory in every segment

10010110 11101011 → Sum

01101001 00010100 → Checksum

23.64

retransmission)

Out-of-Order Segments

Data may arrive out of order and be temporarily stored by the receiving TCP, but yet guarantees that no out-of-order segment is delivered to the process

The window size at the sender is set as follows:

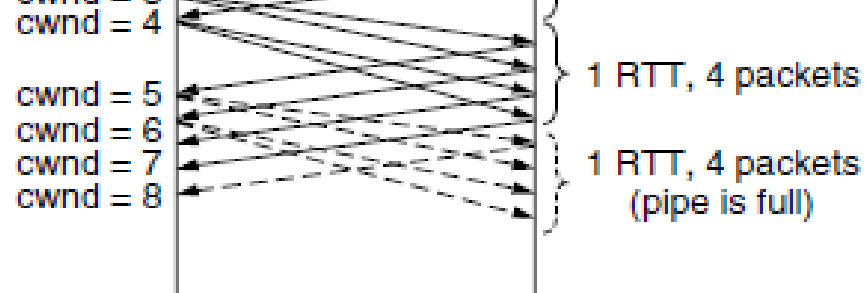
**Send Window = MIN (flow control window,
congestion window)**

where

flow control window is advertised by the receiver (rwnd)

congestion window is adjusted based on feedback from the

slow link permits, but they will not queue up and congest any router along the path. This timing is known as an **ack clock**. It is an essential part of TCP. By using an ack clock, TCP smoothes out traffic and avoids unnecessary queues at routers. This is first consideration



Congestion window by 2—one for each ACK—and next sends four packets.

- The end result is that TCP effectively doubles the number of packets it has in transit every RTT.

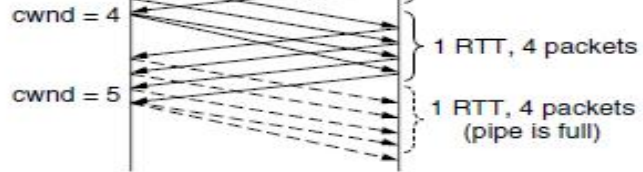


Figure 6-45. Additive increase from an initial congestion window of one segment.

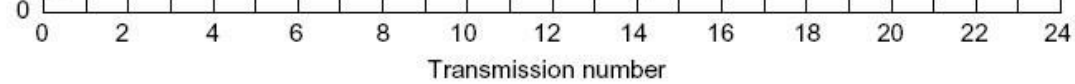
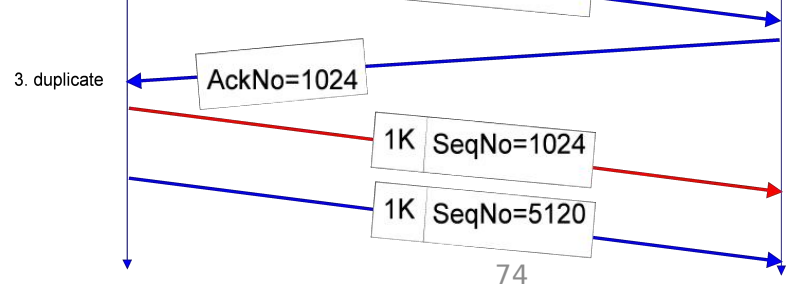


Fig. 6-37. An example of the Internet congestion algorithm.

- ssthresh is set to half the current size of the congestion window:
 $ssthresh = cwnd / 2$
- and slow-start is entered

- Enter slow start:
 $\text{ssthresh} = \text{cwnd}/2$
 $\text{cwnd} = 1$



- **RED** (Floyd and Jacobson 1993)

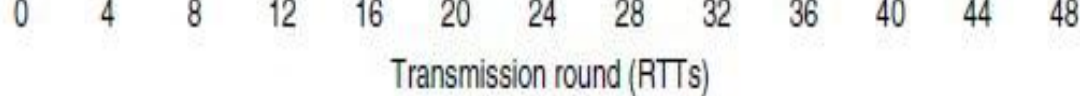


Figure 6-47. Fast recovery and the sawtooth pattern of TCP Reno.

User Datagram

Checksum

UDP Operation

Use of UDP

111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

23.80

(Padding must be added to make the data a multiple of 16 bits)

23.82

Encapsulation and Decapsulation

To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages in an IP datagram.

23.84

10010110 11101011 → Sum

01101001 00010100 → Checksum

23.85

23.86

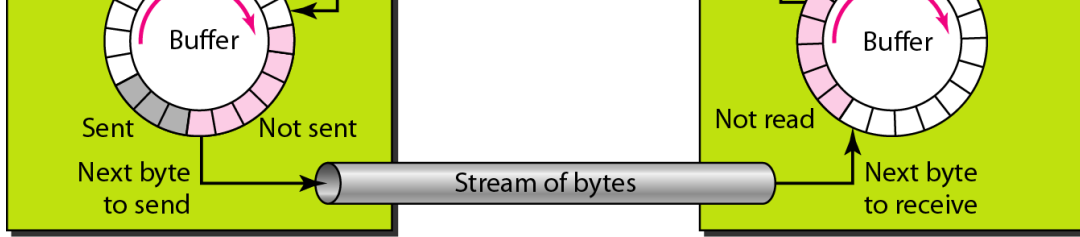
procedure call from the client to the server is not local

Figure 6-29. Steps in making a remote procedure call. The stubs are shaded.

Error Control

25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call





area contains empty chambers to be filled by bytes received from the network. The colored sections contain received bytes that can be read by the receiving process. When a byte is read by the receiving process, the chamber is recycled and added to the pool of empty chambers.

the send operation is transparent to the receiving process. Later we will see that segments may be received out of order, lost, or corrupted and resent. All these are handled by TCP with the receiving process unaware of any activities. Above fig shows how segments are created from the bytes in the buffers

Acknowledgment Number The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive. The acknowledgment number is cumulative.

is also determined by the level of congestion in the network

Options and Padding

23.10

can be piggybacked together.

Header length. This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field can be between 5 ($5 \times 4 = 20$) and 15 ($15 \times 4 = 60$).

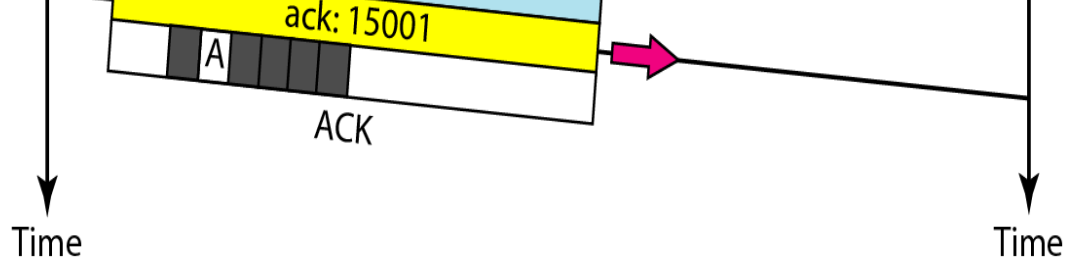
Reserved. This is a 6-bit field reserved for future use.

Control. This field defines 6 different control bits or flags as shown in Figure. One or more of these bits can be set at a time.

checksum for TCP follows the same procedure as the one described for UDP. However, the inclusion of the checksum in the UDP datagram is optional, whereas the inclusion of the checksum for TCP is mandatory. The same pseudoheader, serving the same purpose, is added to the segment. For the TCP pseudoheader, the value for the protocol field is 6.

acknowledgment number field. Note that the sequence number in this segment is the same as the one in the SYN segment; the ACK segment does not consume any sequence numbers.

NOTE: An ACK segment, if carrying no data, consumes no sequence number

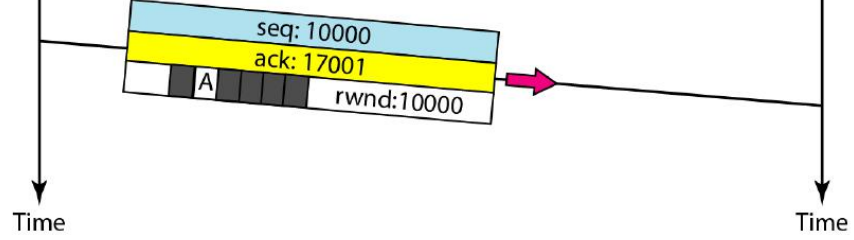


1 Some have imposed a limit on connection requests during a specified period of time.

2 Others filter out datagrams coming from unwanted source addresses.

3 One recent strategy is to postpone resource allocation until the entire connection is set up using what is called a cookie

are received.

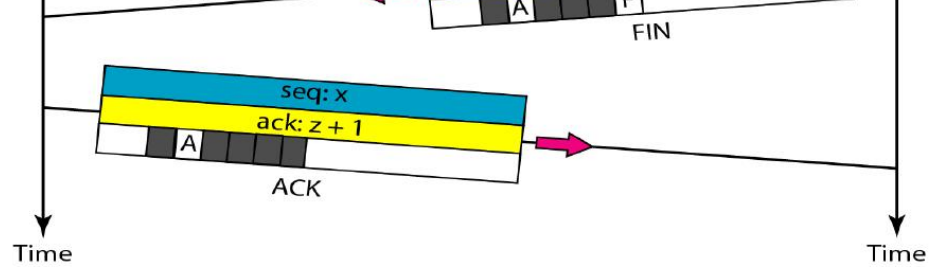


to announce the closing of the connection in the other direction. This segment can also contain the last chunk of data from the server. If it does not carry data, it consumes only one sequence number.

NOTE: The FIN +ACK segment consumes one sequence number if it does not carry data.

needs time for setting, its outboard direction must remain open.





the data link layer was of fixed size

end can accept before its buffer overflows and data are discarded.

The congestion window is a value determined by the network to avoid congestion



Window management in TCP

data until another 2 KB came in, to be able to transmit a segment with a 4-KB payload. This freedom can be used to improve performance

Client □ **Server:** Send typed character

Server □ **Client:** Echo of character (or user output) and
acknowledgement for first packet

Client □ **Server:** Acknowledgement for second packet

41-byte packets containing 1 byte of data) is still operating inefficiently. A way to reduce this usage is known as **Nagle's algorithm (Nagle, 1984)**.

Because of these problems, Nagle's algorithm can be disabled (which is called the *TCP NODELAY option*).



The receiver will buffer the data until it can be passed up to the application in order (handling out of order segments)

Cumulative acknowledgements

corrupted segment. If the segment is corrupted, it is discarded by the destination TCP and is considered as lost. TCP uses a 16-bit checksum that is mandatory in every segment

10010110 11101011 → Sum

01101001 00010100 → Checksum

23.36

retransmission)

Out-of-Order Segments

Data may arrive out of order and be temporarily stored by the receiving TCP, but yet guarantees that no out-of-order segment is delivered to the process

The window size at the sender is set as follows:

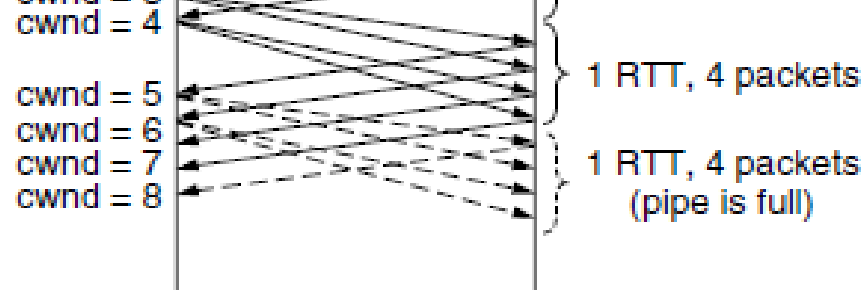
$$\text{Send Window} = \text{MIN} (\text{flow control window}, \text{congestion window})$$

where

flow control window is advertised by the receiver (rwnd)

congestion window is adjusted based on feedback from the

slow link permits, but they will not queue up and congest any router along the path. This timing is known as an **ack clock**. It is an essential part of TCP. By using an ack clock, TCP smoothes out traffic and avoids unnecessary queues at routers. This is first consideration



Congestion window by 2—one for each ACK—and next sends four packets.

- The end result is that TCP effectively doubles the number of packets it has in transit every RTT.

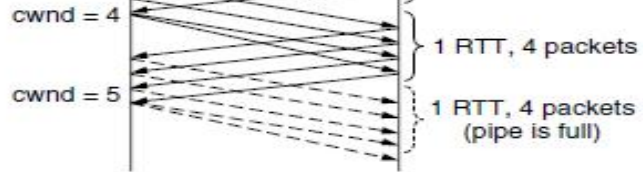


Figure 6-45. Additive increase from an initial congestion window of one segment.

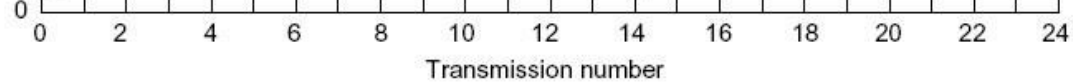
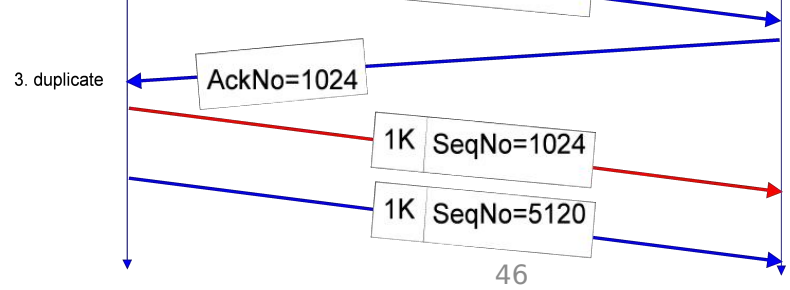


Fig. 6-37. An example of the Internet congestion algorithm.

- ssthresh is set to half the current size of the congestion window:
 $ssthresh = cwnd / 2$
- and slow-start is entered

- Enter slow start:
 $ssthresh = cwnd/2$
 $cwnd = 1$



- **RED** (Floyd and Jacobson 1993)

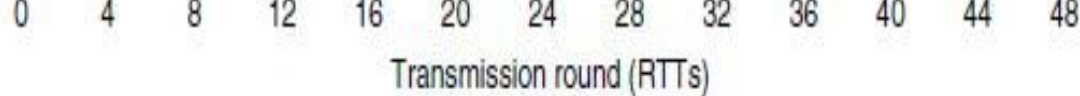


Figure 6-47. Fast recovery and the sawtooth pattern of TCP Reno.

User Datagram

Checksum

UDP Operation

Use of UDP

111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

23.52

(Padding must be added to make the data a multiple of 16 bits)

23.54

Encapsulation and Decapsulation

To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages in an IP datagram.

23.56

10010110 11101011 → Sum

01101001 00010100 → Checksum

23.57

23.58

procedure call from the client to the server is not local

Figure 6-29. Steps in making a remote procedure call. The stubs are shaded.

RFC 3550 and is now in widespread use for multimedia applications. We will describe two aspects of real-time transport.

The first is the RTP protocol for transporting audio and video data in packets.

The second is the processing that takes place, mostly at the receiver, to play out the audio and video at the right time..



Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Figure 6.54: (a) The position of R11 in the protocol stack. (b) Packet numbering.

Each RTP payload may contain multiple samples, and they may be coded any way that the application wants. To allow for interworking, RTP defines several profiles (e.g., a single audio stream), and for each profile, multiple encoding formats may be allowed

Another facility many real-time applications need is time stamping. Not only does time stamping reduce the effects of variation in network delay, but it

Figure 6-31. The RTP header.

jitter at the receiver by decoupling the playback from the packet arrival time.

The *Synchronization source identifier* tells which stream the packet belongs to. It is the method used to multiplex and demultiplex multiple data streams onto a single stream of UDP packets.

Finally, the *Contributing source identifiers*, if any, are used when

granularities and different unit rates. RTCP can be used to keep them in sync.

Finally, RTCP provides a way for naming the various sources (e.g., in ASCII text). This information can be displayed on the receiver's screen to indicate who is talking at the moment.

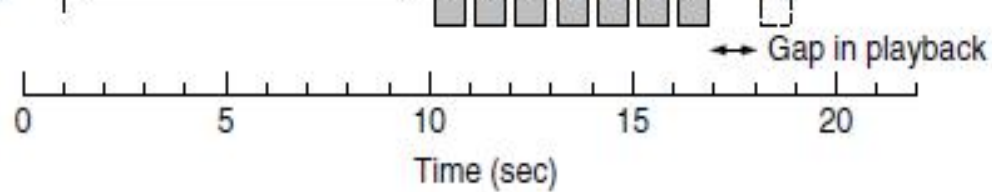


Figure 6-32. Smoothing the output stream by buffering packets.

One way to avoid this problem for audio is to adapt the playback point between **talk spurts**, in the gaps in a conversation. No one will notice the difference between a short and slightly longer silence



b. Remote log-in

at the TCP/IP stack at the remote machine. Here the characters are delivered to the operating system and passed to the TELNET server, which changes the characters to the corresponding characters understandable by the remote computer. However, the characters cannot be passed directly to the operating system because the remote operating system is not designed to receive characters from a TELNET server: It is designed to receive characters from a terminal driver. The solution is to add a piece of

Data character

Control character



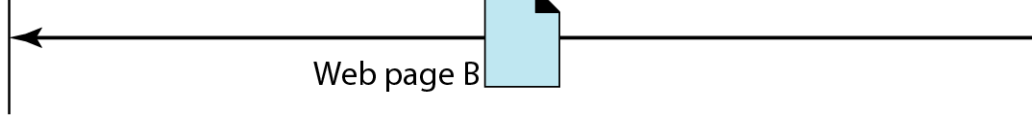
26.75

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display. 27.76

Server

Uniform Resource Locator
Cookies

27.77



27.78

The Web page is stored at the server. Each time a client request arrives, the corresponding document is sent to the client. To improve efficiency, servers normally store requested files in a cache in memory; memory is faster to access than disk. A server can also become more efficient through multithreading or multiprocessing. In this case, a server can answer more than one

27.80

is separated from the host by a colon.

Path is the pathname of the file where the information is located. Note that the path can itself contain slashes that, in the UNIX operating system, separate the directories from the subdirectories and files.

27.82

Static Documents

Dynamic Documents

Active Documents

27.83



Static HTML document

27.84



27.85

27.86

Server Pages (JSP), which uses the Java language for scripting; Active Server Pages (ASP), a Microsoft product which uses Visual Basic language for scripting; and ColdFusion, which embeds SQL database queries in the HTML document

27.88

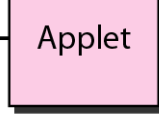
27.89

27.90

Run the applet
to get the result



Result



Applet

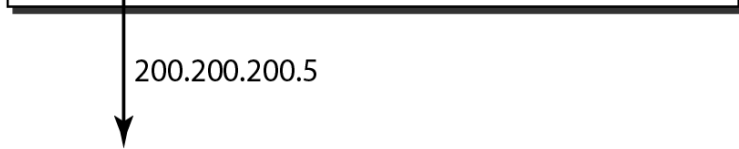
27.91

Result

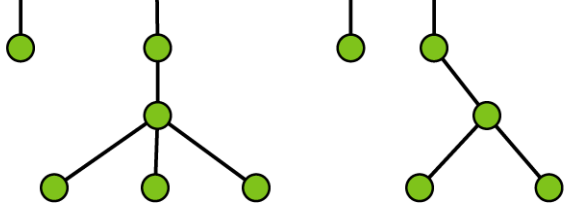
27.92

27.93

27.94

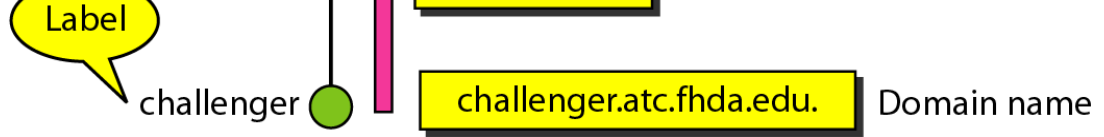


Transport layer

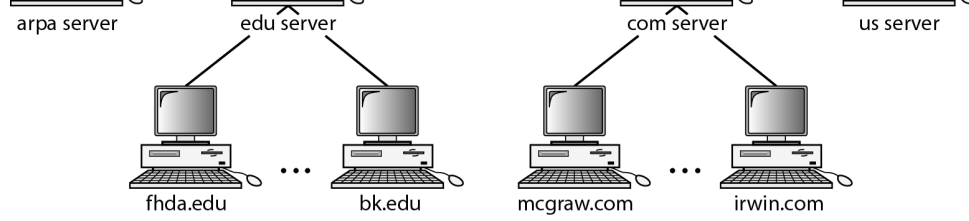


cs.hmme.com.
www.funny.int.

cs.hmme
www





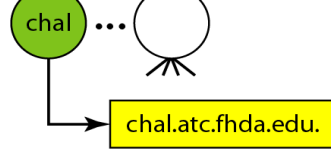




Inverse
domain

Generic
domains

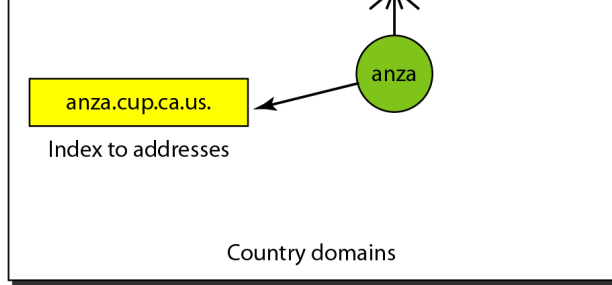
Country
domains

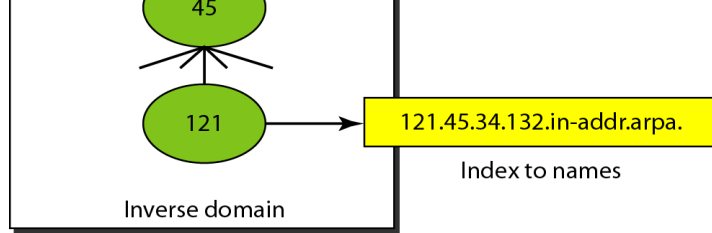


Index to addresses

Generic domains

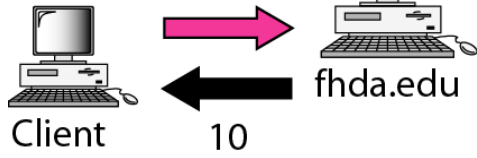
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

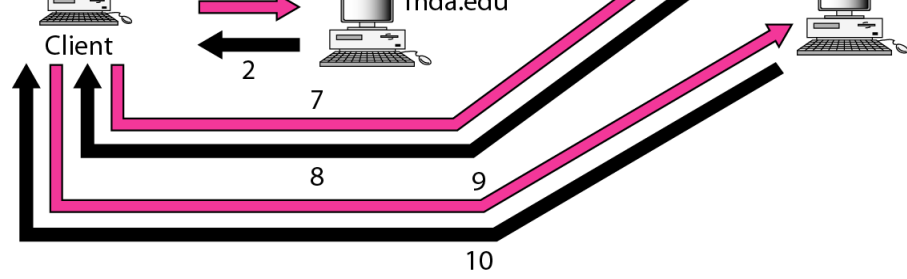




Recursive Resolution

The client (resolver) can ask for a recursive answer from a name server. This means that the resolver expects the server to supply the final answer.. When the query is finally resolved, the response travels back until it finally reaches the requesting client. This is called recursive resolution and is shown in FIG





and additional records

new are new domains added to DNS. This is done through a registrar, a commercial entity accredited by ICANN. A registrar first verifies that the requested domain name is unique and then enters it into the DNS database. A fee is charged. Today, there are many registrars; their names and addresses can be found at <http://www.intenic.net>

