LECTURE NOTES

ON

DATA COMMUNICATION & COMPUTER NETWORK

DIPLOMA II Year IV Semester

DEPARTMENT OF CSE

NM INSTITUTE OF ENGINEERING & TECHNOLOGY

Sijua, Patrapada, Near AIIMS , Bhubaneswar-751019,Odisha

NETWORKS

A network is a set of devices (often referred to as *nodes)* connected by 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, issharing 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 graphicalcomputer screen
- doing business electronically, especially with customers and suppliers. This new model is called **e-commerce** (**electronic commerce**) and it has grownrapidly in recent years.

2 Home Applications

- **peer-to-peer** communication

- electronic commerce
- entertainment.(game playing,)

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 manyunsolved 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 clipsare 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.

BOTNET ATTACK: Botnets can be used to perform distributed denial-of-service attack (DDoS attack), steal data, send spam, and allows the attacker to access the device and its connection.

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

I. **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 unevendelay 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.

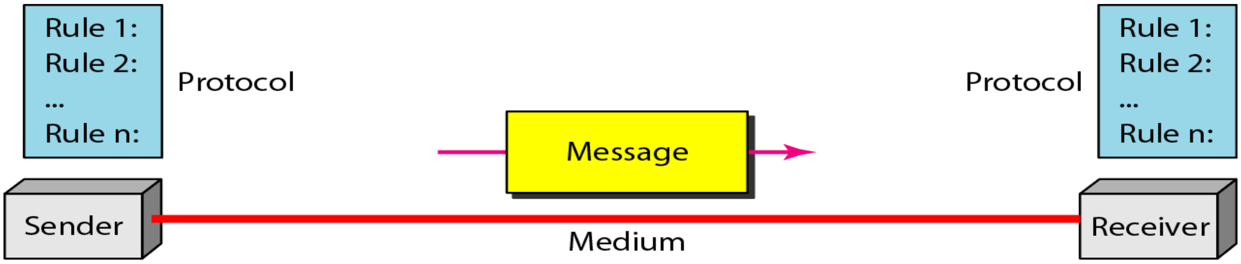
A data communications system has five components

I. **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 acomputer, workstation, telephone handset, video camera, and so on.

3. **Receiver.** The receiver is the device that receives the message. It can be acomputer, 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.

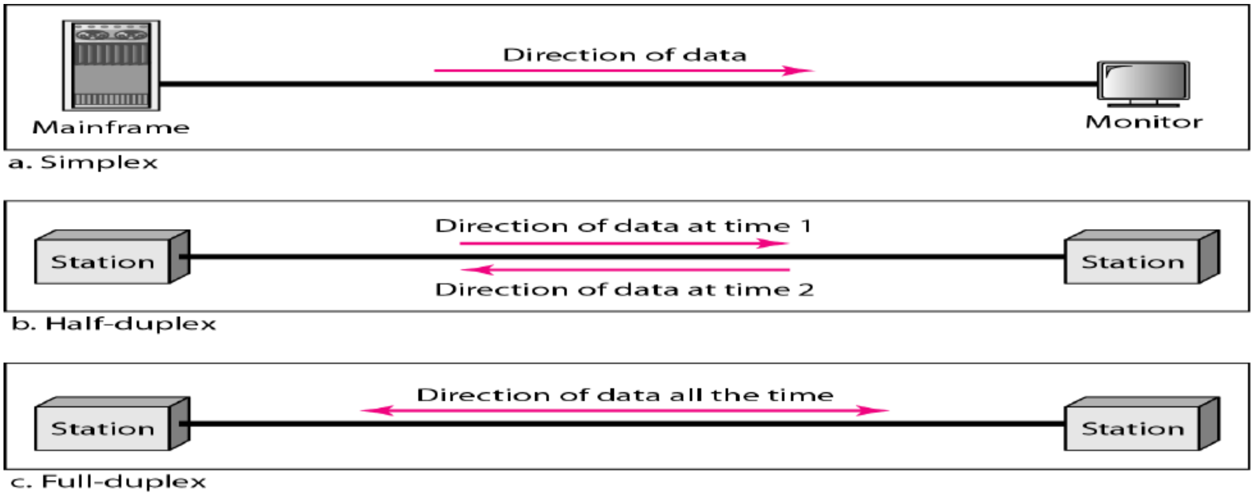


Data Representation

Text
Numbers
Images
Audio
Video

Data Flow

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


Direction of data
Mainframe
Monitor
Direction of data at time 1
Station
Station
Direction of data at time 2
Direction of data all the time
Station
Station

a. Simplex

b. Half-duplex

c. Full-duplex

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 andlisten at the same time. The full-duplex mode is used when communication inboth directions is required all the time.

Network Criteria

A network must be able to meet a certain number of criteria. The mostimportant 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 an inquiry and a response. The performance of a network depends on a number of factors, including the number of users, the type of transmission medium, the capabilities of the connected hardware, and the efficiency of the software.

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

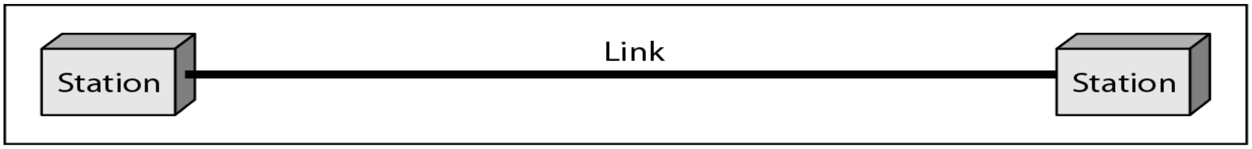
A network is two or more devices connected through links. A link is a communications pathway that transfers data from one device to another.

There are two possible types of connections: point-to-point and multipoint. **Point-to-Point** A point-to-point connection provides a dedicated link betweentwo devices. The entire capacity of the link is reserved for transmissionbetween those two devices. Most point-to-point connections use an actuallength of wire or cable to connect the two ends, but other options, such as microwave or satellite links, are also possible

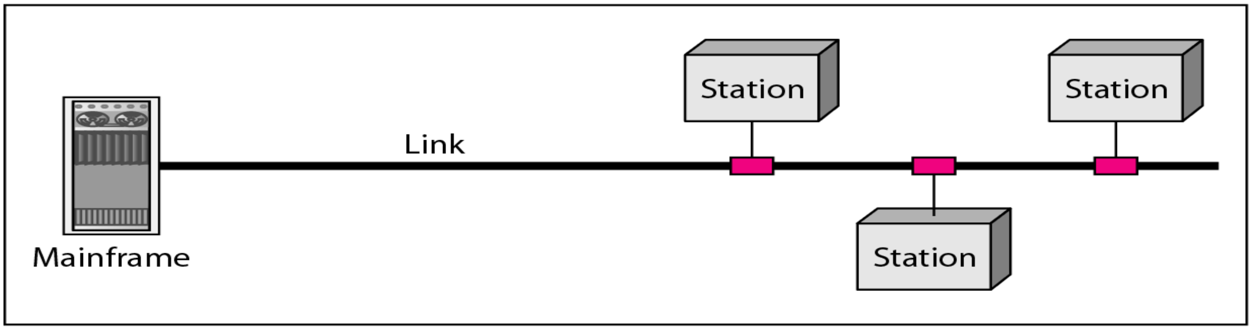
When you change television channels by infrared remote control, you are establishing a point-to-point connection between the remote control and the television's control system.

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, eitherspatially 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



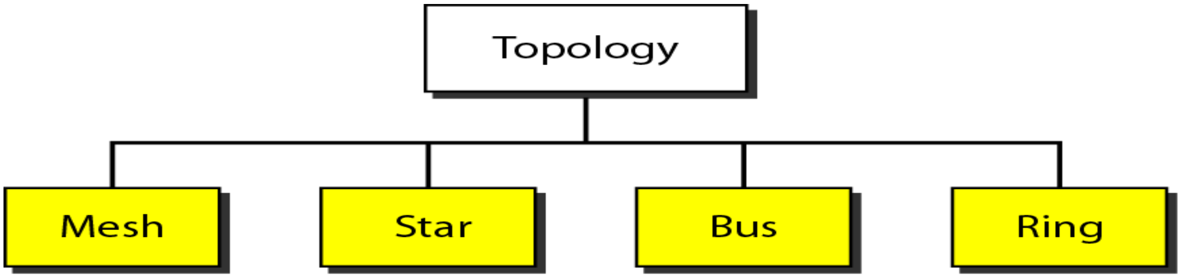
b. Multipoint

Physical Topology

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

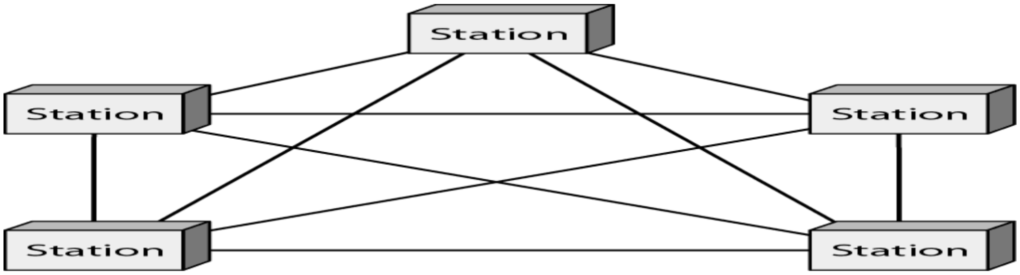
Two or more devices connect to a link; two or more links form a topology. The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to one another.

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 nodein 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 can be calculated using the following formula (*n* is the number of computers in the network): **n(n-1)/2**

In a *partially connected mesh topology*, at least two of the computers in the 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.

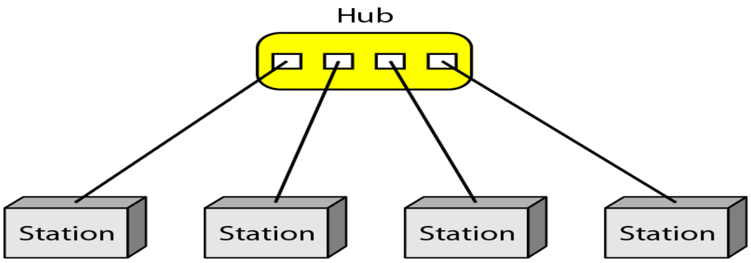
Advantages of a mesh topology

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

Disadvantages of a mesh topology

- The cost to implement is higher than other network topologies, making it aless desirable option.
- Building and maintaining the topology is difficult and time consuming.
- The chance of redundant connections is high, which adds to the high costsand 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, likea 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 usedin each computer of the star topology, a coaxial cable or a RJ-45 network cableis used to connect computers together.

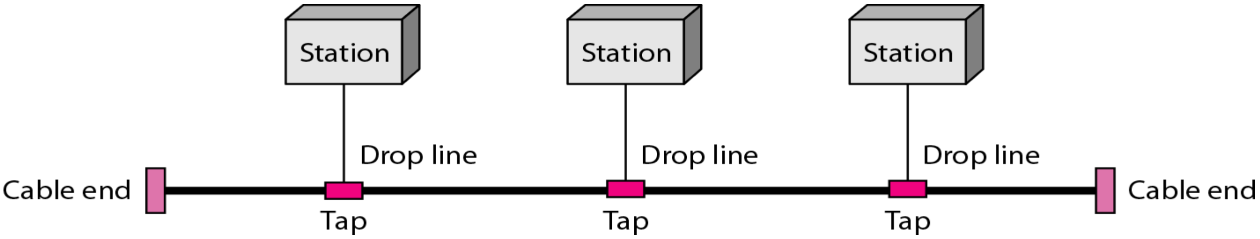
Advantages of star topology

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

Disadvantages of star topology

- Can have a higher cost to implement, especially when using a switch orrouter as the central network device.
- The central network device determines the performance and number ofnodes the network can handle.
- If the central computer, hub, or switch fails, the entire network goes downand all computers are disconnected from the network

BUS:



a **line topology**, a **bus topology** is a network setup in which each computerand network device are connected to a single cable or backbone.

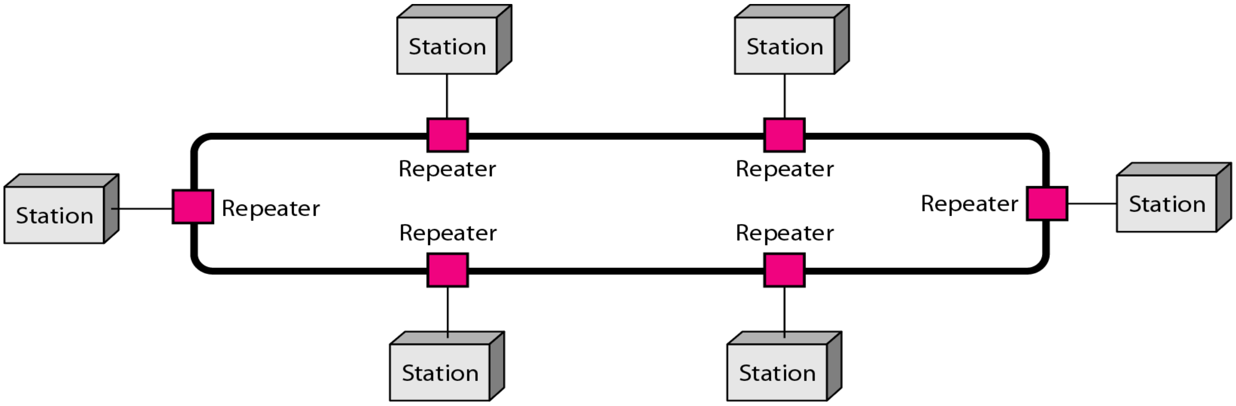
Advantages of bus topology

- It works well when you have a small network.
- It's the easiest network topology for connecting computers or peripheralsin 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:



A **ring topology** is a network configuration in which device connections createa circular data path. In a ring network, packets of data travel from one device tothe next until they reach their destination. Most ring topologies allow packets to 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).

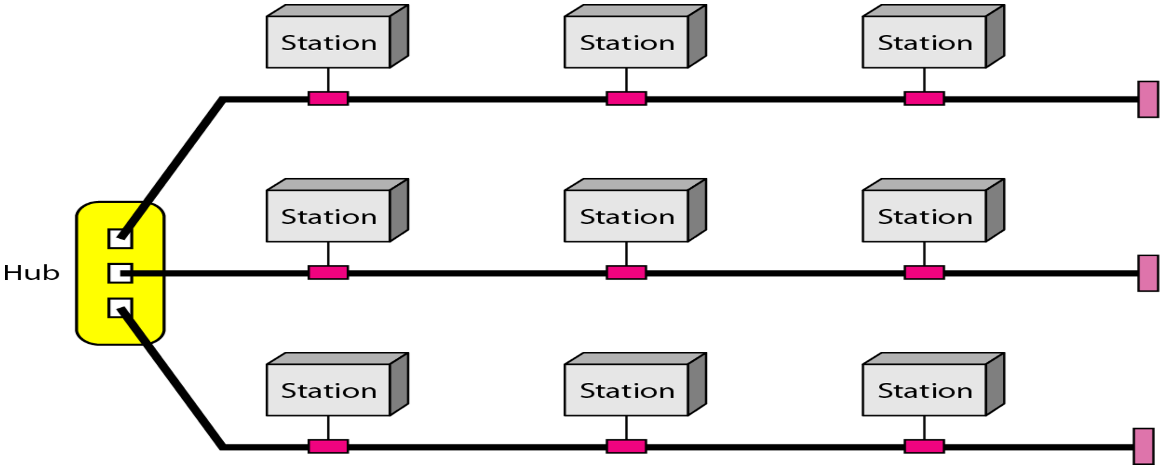
Advantages of ring topology

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

Disadvantages of ring topology

- All data being transferred over the network must pass through eachworkstation 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 moreexpensive 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 asshown in Figure



Types of Network based on size

The types of network are classified based upon the size, the area it covers andits 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

LAN (Local Area Network)

Group of interconnected computers within a small area. (room, building, campus)

Two or more pc's can from a LAN to share files, folders, printers, applicationsand 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 beshared.

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)**

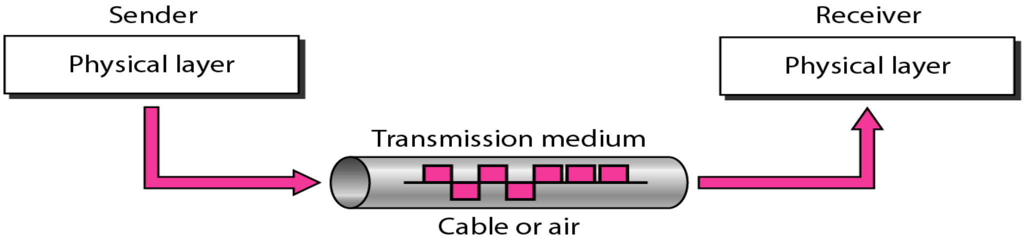
Network organized by the individual user for its personal use.

SAN (Storage Area Network)

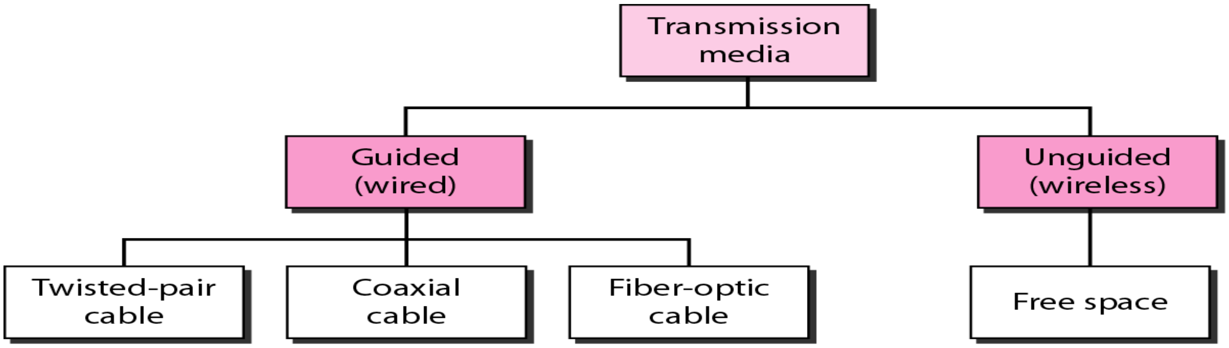
Connects servers to data storage devices via fiber-optic cables.E.g.: Used for daily backup of organization or a mirror copy

---

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

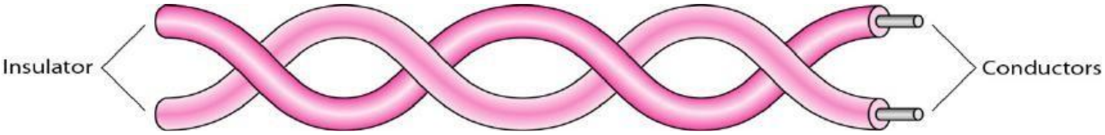


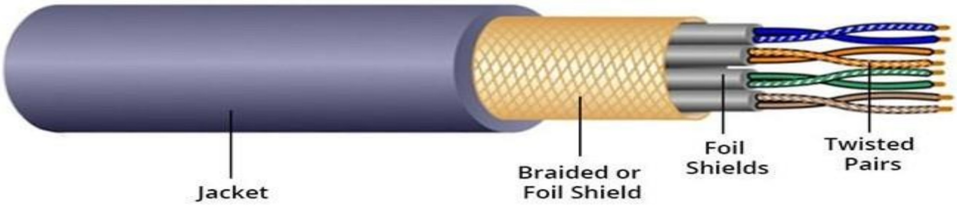
Classes of transmission media



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

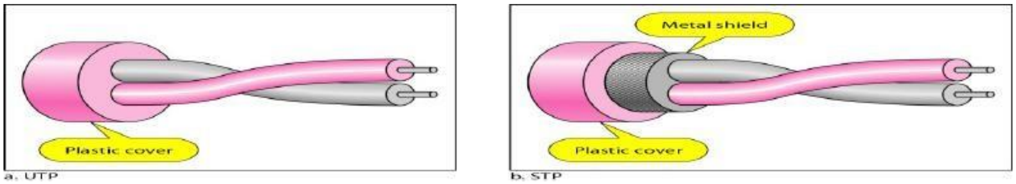
Twisted-Pair Cable: A twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together. One of the wires is used to carry signals to the receiver, and the other is used only as a ground reference.





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 casingimproves the quality of cable by preventing the penetration of noise or crosstalk, it is bulkier and more expensive.



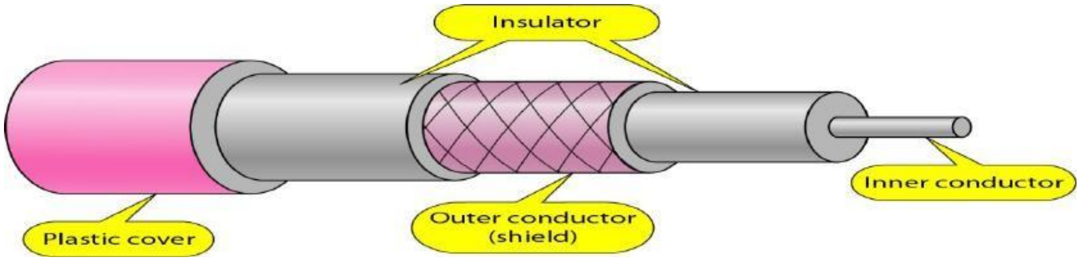
The most common UTP connector is RJ45 (RJ stands for registered jack)

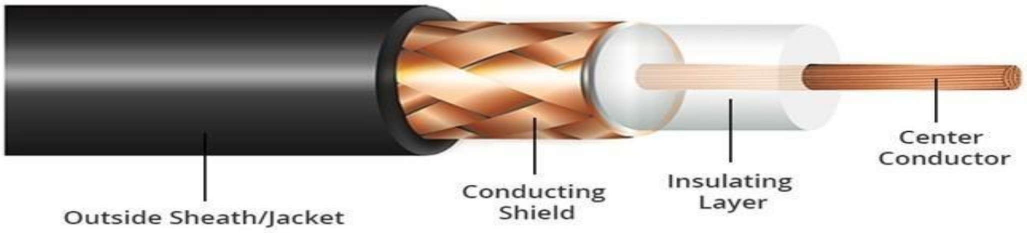
Applications

Twisted-pair cables are used in telephone lines to provide voice and datachannels.
Local-area networks, such as l0Base-T and l00Base-T, also use twisted-paircables.

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 (usuallycopper) 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 Bayone-Neill-Concelman(BNe), connector.

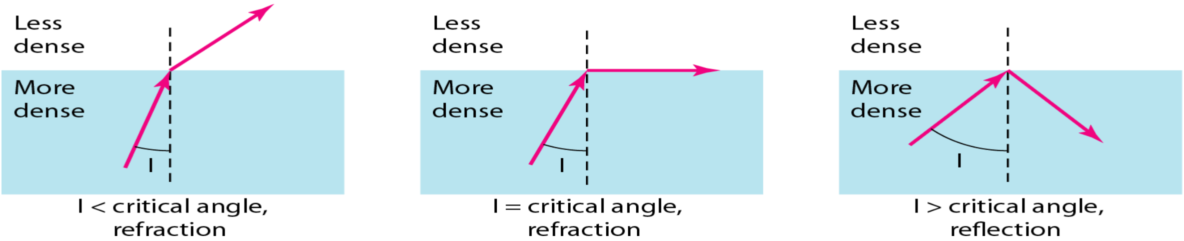
Applications

Coaxial cable was widely used in analog telephone networks,digital telephonenetworks
Cable TV networks also use coaxial cables.
Another common application of coaxial cable is in traditional Ethernet LANs

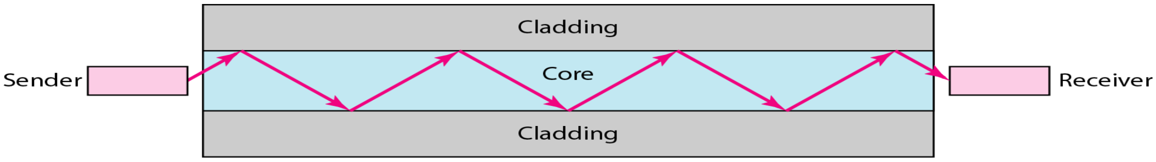
Fiber-Optic Cable

A fiber-optic cable is made of glass or plastic and transmits signals in the formof 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.

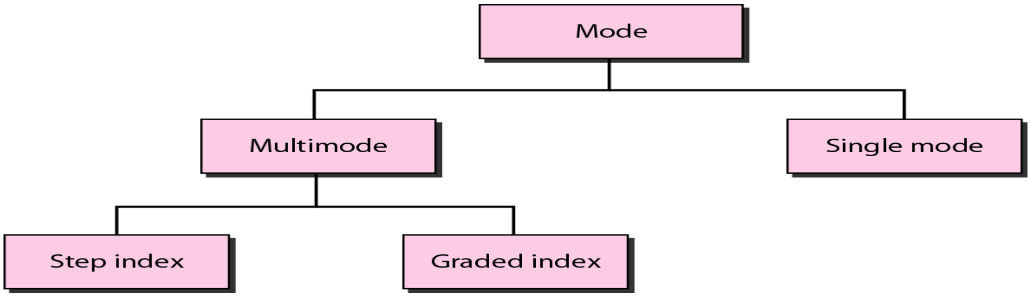
Bending of light ray



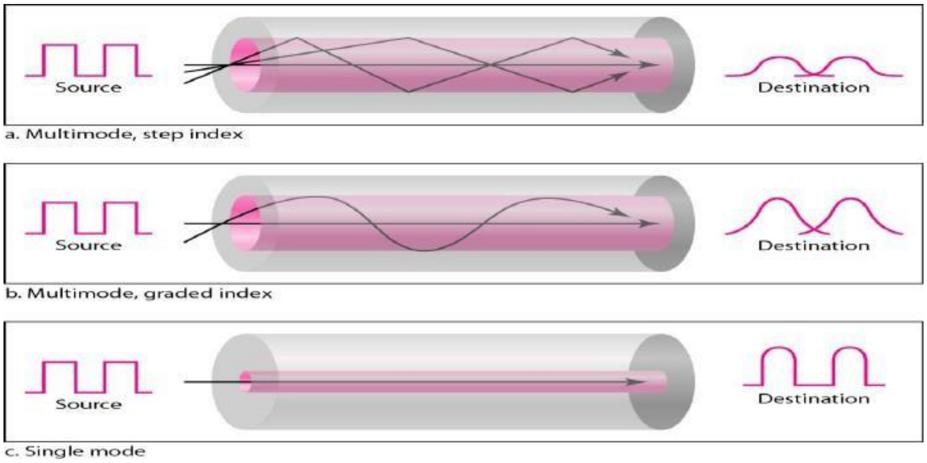
Optical fibers use reflection to guide light through a channel. A glass or plastic core is surrounded by a cladding of less dense glass or plastic.



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.

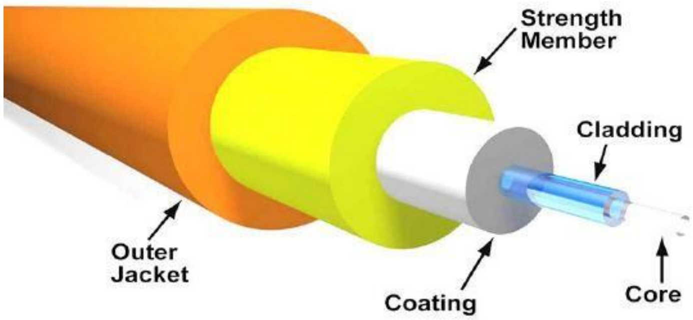


In **multimode step-index fiber**, the density of the core remains constant from the center to the edges. A beam of light moves through this constant density ina straight line until it reaches the interface of the core and the cladding. The term *step index* refers to the suddenness of this change, which 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 sourceof light that limits beams to a small range of angles, all close to the horizontal.

Fiber Construction



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 coaxialcable,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 greaterthan 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.1Installation and maintenance

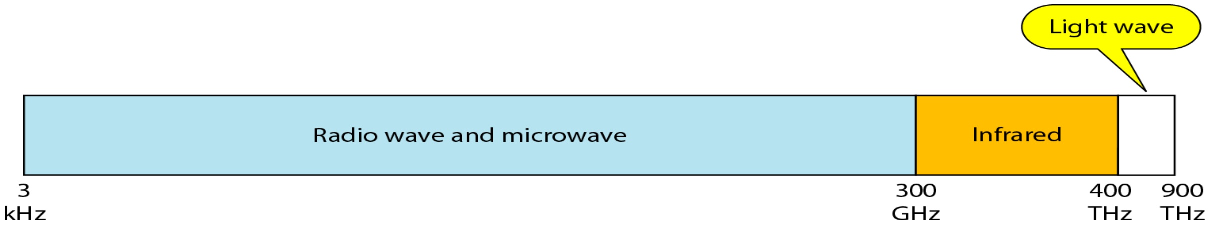
2 Unidirectional light propagation. Propagation of light is unidirectional. If weneed bidirectional communication, two fibers are needed.

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.

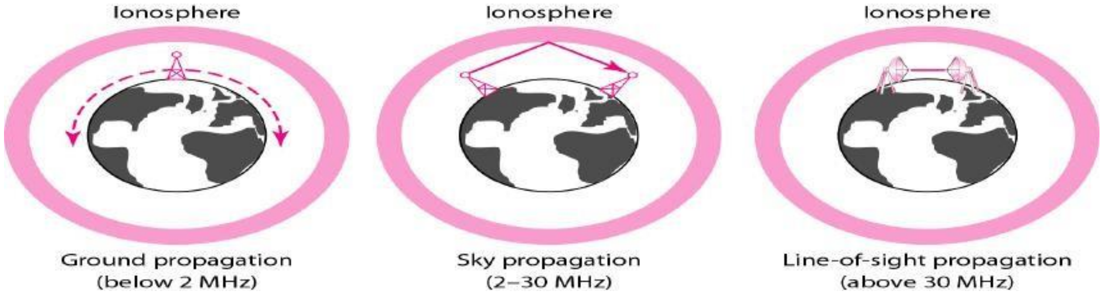
UNGUIDED MEDIA: WIRELESS

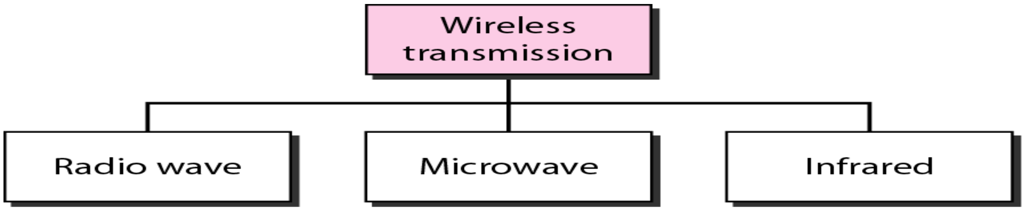
Unguided media transport electromagnetic waves without using a physical conductor. This type of communication is often referred to as wireless communication.

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


Ionosphere
Ionosphere
Ionosphere
Ground propagation (below 2 MHz)
Sky propagation (2–30 MHz)
Line-of-sight propagation (above 30 MHz)

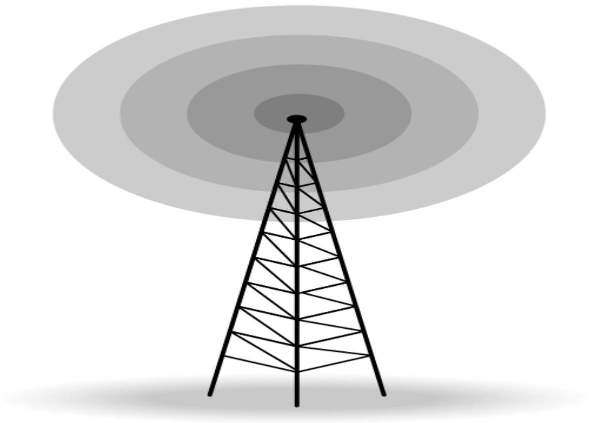


Radio Waves

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

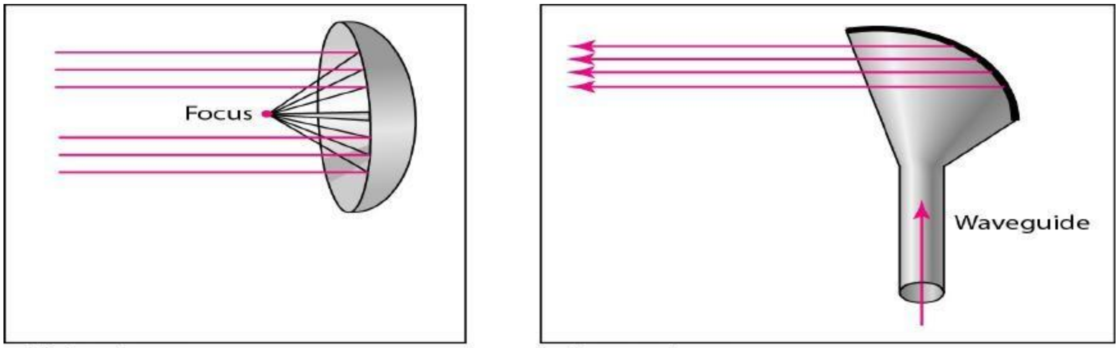
The Omni directional characteristics of radio waves make them useful for multicasting, in which there is one sender but many receivers. AM and FM radio, television, maritime radio, cordless phones, and paging are examples of multicasting.

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

characteristic prevents interference between one system and another; a short- range communication system in one room cannot be affected by another system in the next room.
When we use our infrared remote control, we do not interfere with the use ofthe remote by our neighbors. Infrared signals 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:

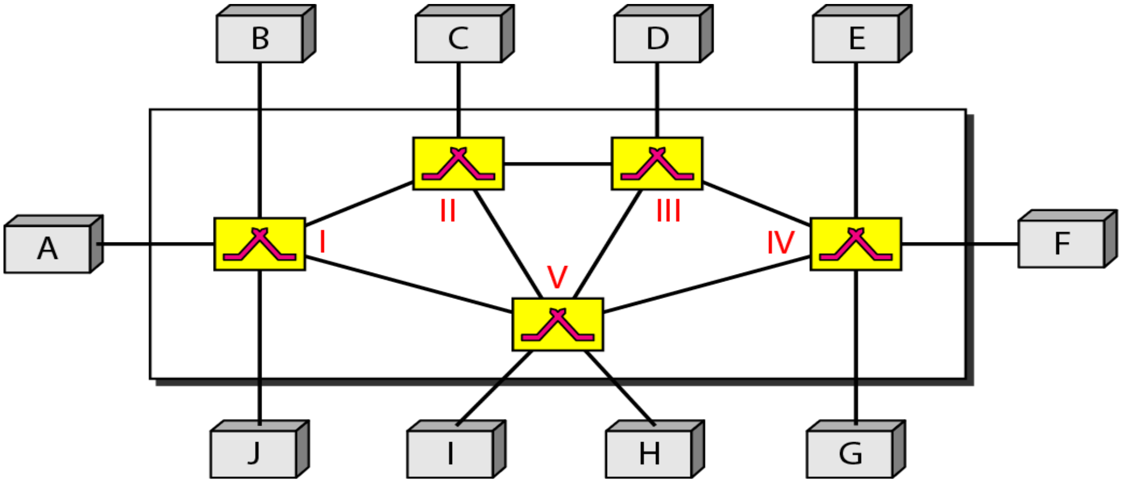
Infrared signals can be used for short-range communication in a closedarea using line-of-sight propagation.

Switching

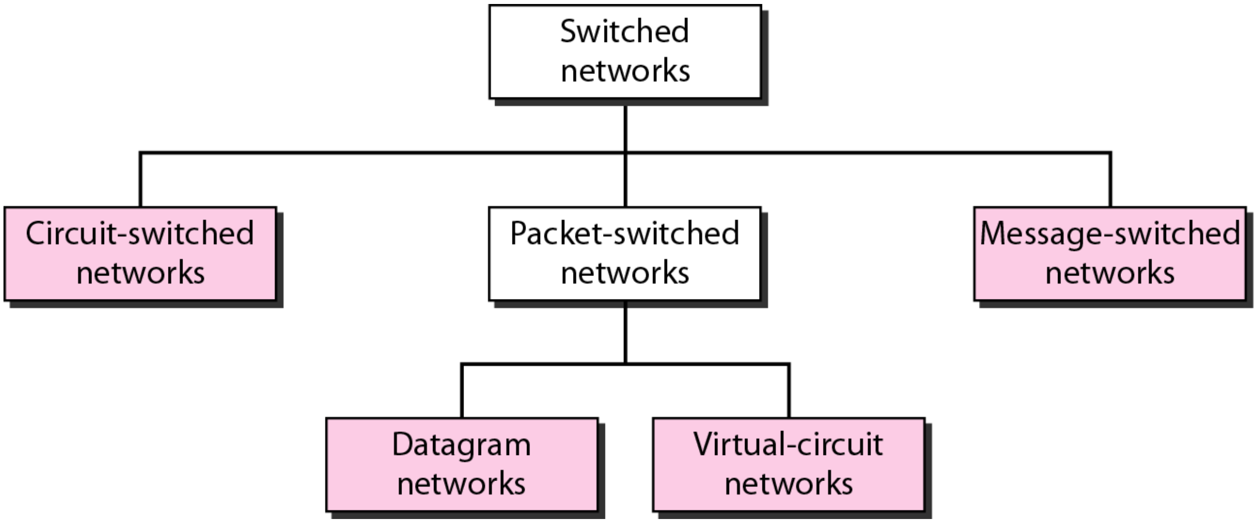
A network is a set of connected devices. Whenever we have multiple devices, we have the problem of how to connect them to make one-to-one communication possible. One solution is to make a point-to-point connection between each pair of devices (a mesh topology) or between a central deviceand every other 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.



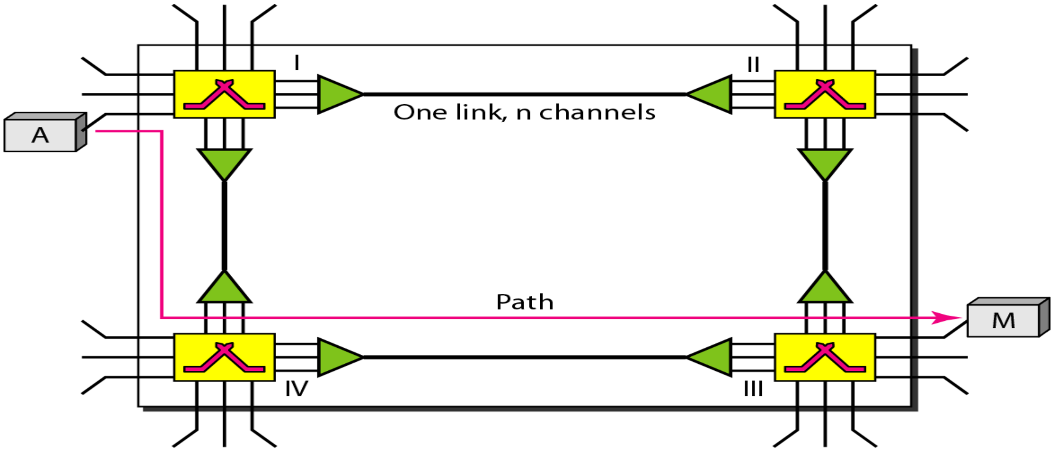
We can then divide today's networks into three broad categories: circuit- switched networks, packet-switched networks, and message-switched. Packet- switched networks can further be divided into two subcategories-virtual-circuit networks and datagram networks as shown in Figure.



CIRCUIT-SWITCHED NETWORKS

A circuit-switched network consists of a set of switches connected byphysical links. A connection between two stations is a dedicated path made ofone or more links. However, each connection uses only one dedicated channelon 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 theteardown 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 dedicated channel between itself and switch III. Switch III informs system M of system A's intention at this time.

In the next step to making a connection, an acknowledgment from system M needs to be sent in the opposite direction to system A. Only after system Areceives this acknowledgment is the connection established.

Data Transfer Phase

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

Teardown Phase

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

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

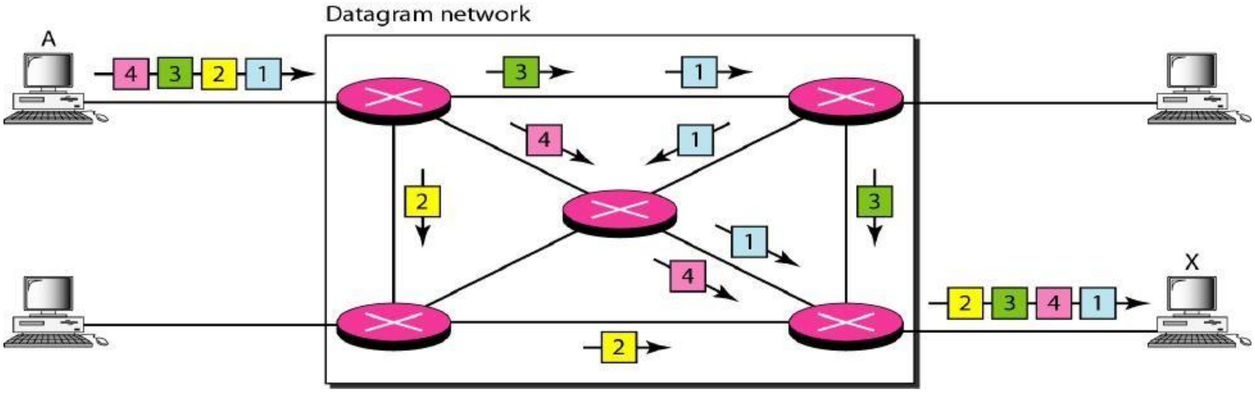
In a packet-switched network, there is no resource reservation; resources are allocated on demand. The allocation is done on a first come, first-served basis. When a switch receives a packet, no matter what is the source or destination, the packet must wait if there are other packets being processed. 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 traditionallyreferred 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.

A switch in a datagram network uses a routing table that is based on the destination address. The destination address in the header of a packet in a datagram network remains the same during the entire journey of the packet.



Efficiency

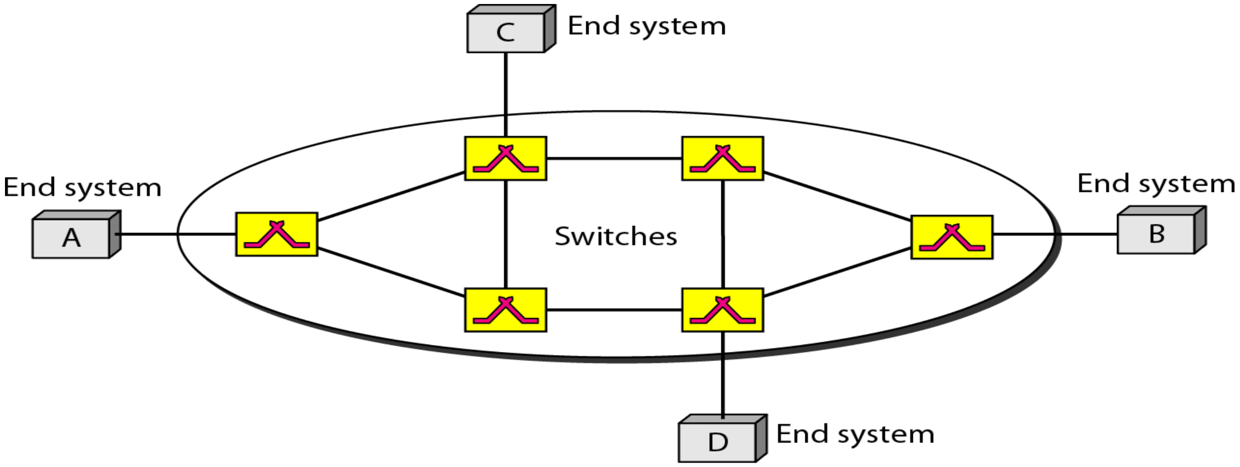
The efficiency of a datagram network is better than that of a circuit-switchednetwork; resources are allocated only when there are packets to be transferred. **Delay**

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 topacket 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.



1. As in a circuit-switched network, there are setup and teardown phases in addition to the data transfer phase.
2. Resources can be allocated during the setup phase, as in a circuit-switched 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 andlocal (virtual-circuit identifier).

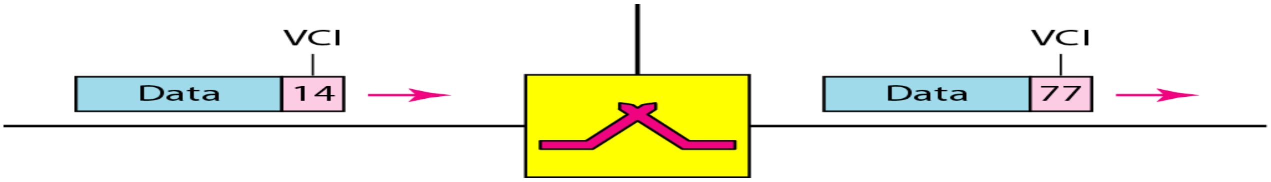
Global Addressing

A source or a destination needs to have a global address-an address that can beunique 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 VCl.

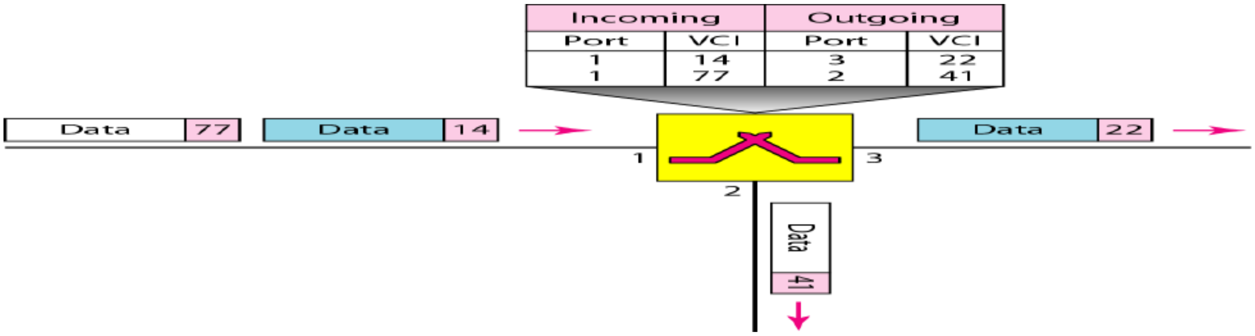
Figure shows how the VCI in a data frame changes from one switch to another. Note that a VCI does not need to be a large number since each switch can useits own unique set of VCls.



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 thentalk about the setup and teardown phases.

Data Transfer Phase



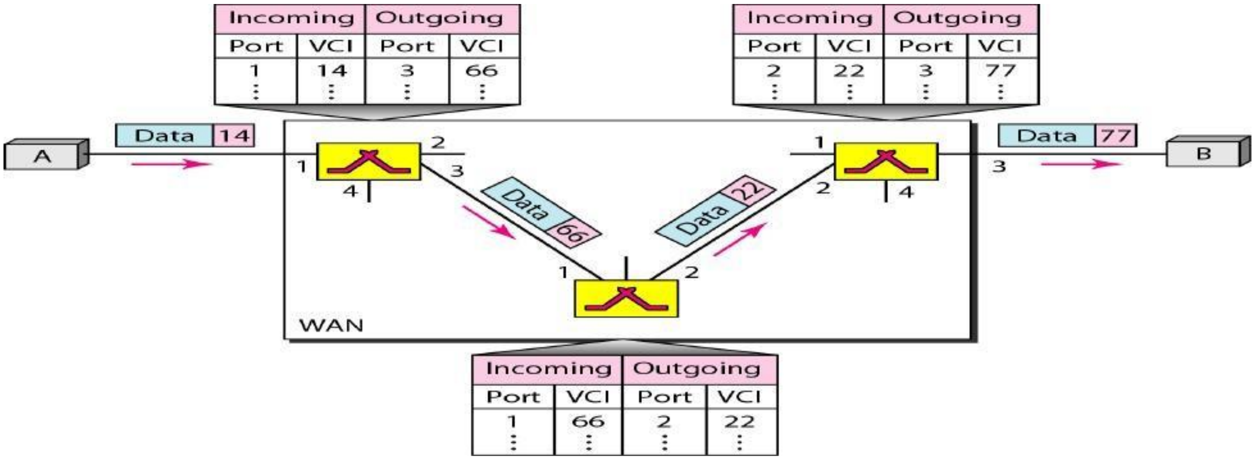
Incoming		Outgoing	
Port	VCI	Port	VCI
1	14	3	22
1	77	2	41

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 fourcolumns.

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.



Incoming		Outgoing	
Port	VCI	Port	VCI
1	14	3	66
⋮	⋮	⋮	⋮

Incoming		Outgoing	
Port	VCI	Port	VCI
2	22	3	77
⋮	⋮	⋮	⋮

Incoming		Outgoing	
Port	VCI	Port	VCI
1	66	2	22
⋮	⋮	⋮	⋮

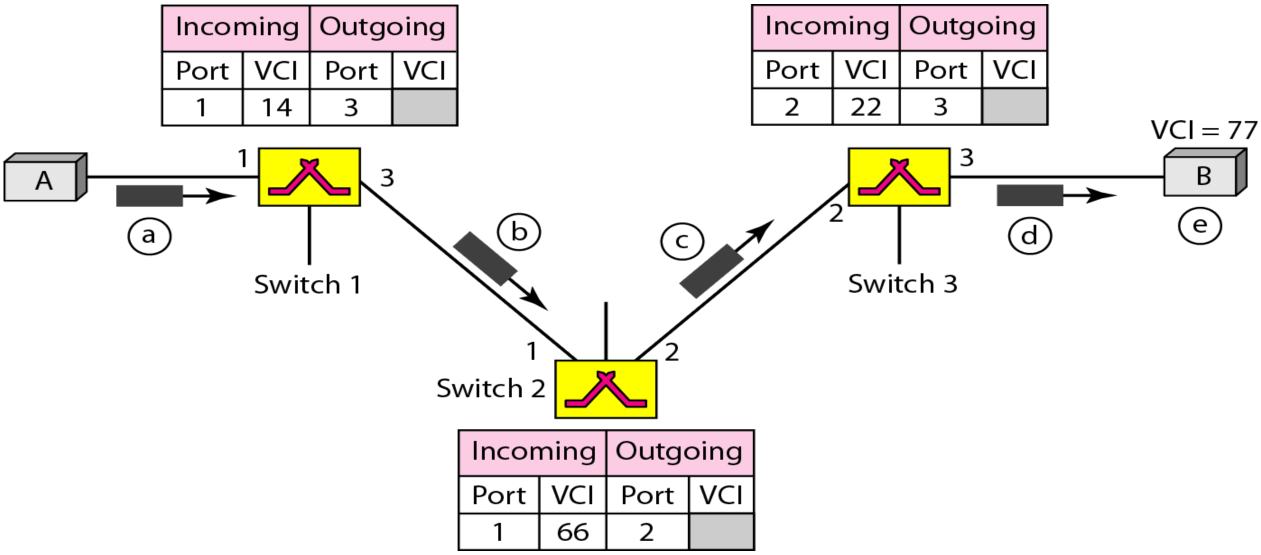
Each switch changes the VCI and routes the frame.

The data transfer phase is active until the source sends all its frames tothe 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.



Incoming		Outgoing	
Port	VCI	Port	VCI
1	14	3	

Incoming		Outgoing	
Port	VCI	Port	VCI
2	22	3	

Incoming		Outgoing	
Port	VCI	Port	VCI
1	66	2	

a. Source A sends a setup frame to switch 1.

b. 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.

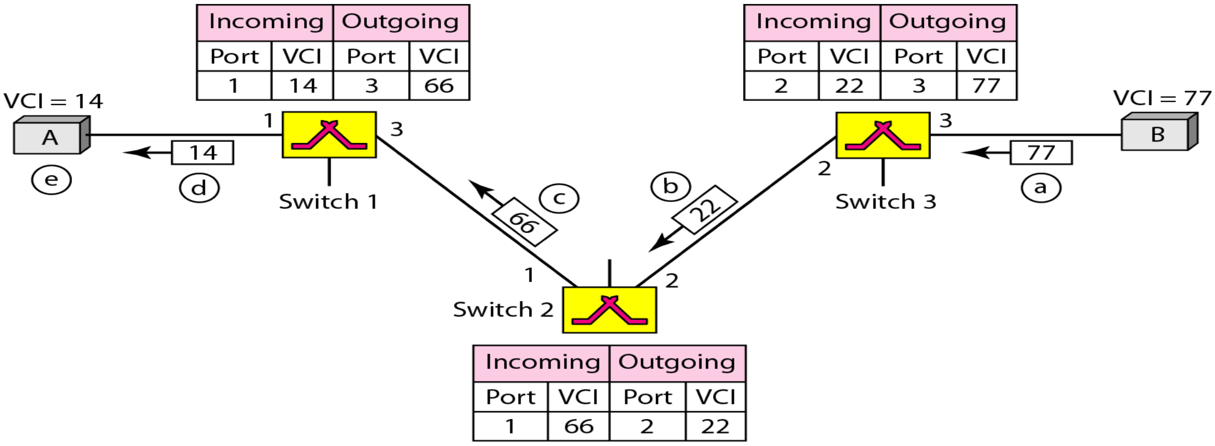
c. Switch 2 receives the setup request frame. The same events happen here as at switch 1; three columns of the table are completed: in this case, incoming port (l), incoming VCI (66), and outgoing port (2).

d. Switch 3 receives the setup request frame. Again, three columns are completed: incoming port (2), incoming VCI (22), and outgoing port (3).

e. 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.



Incoming		Outgoing	
Port	VCI	Port	VCI
1	14	3	66

Incoming		Outgoing	
Port	VCI	Port	VCI
2	22	3	77

Incoming		Outgoing	
Port	VCI	Port	VCI
1	66	2	22

a. The destination sends an acknowledgment to switch 3. The acknowledgment carries the global source and destination addresses so the switch knows whichentry in the table is to be completed. The frame also carries VCI 77, chosen by the destination as the incoming VCI for frames from A. Switch 3 uses this VCI tocomplete 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 VCIin the table, chosen in the previous step. Switch 2 uses this as the outgoing VCIin the table.

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

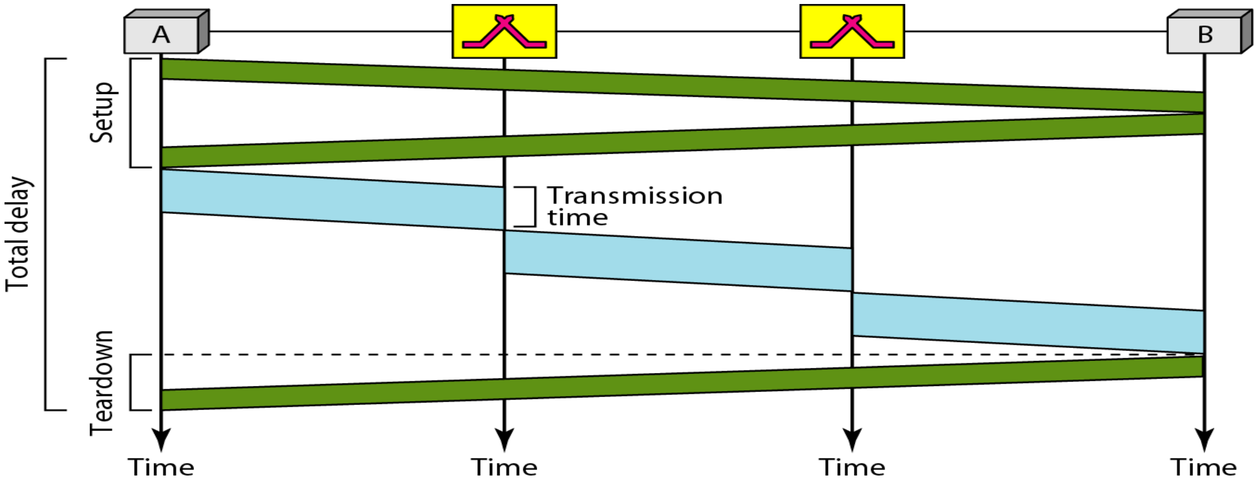
In this phase, source A, after sending all frames to B, sends a special frame called a *teardown request.* Destination B responds with a teardown confirmation frame. All switches delete the corresponding entry from their tables.

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.

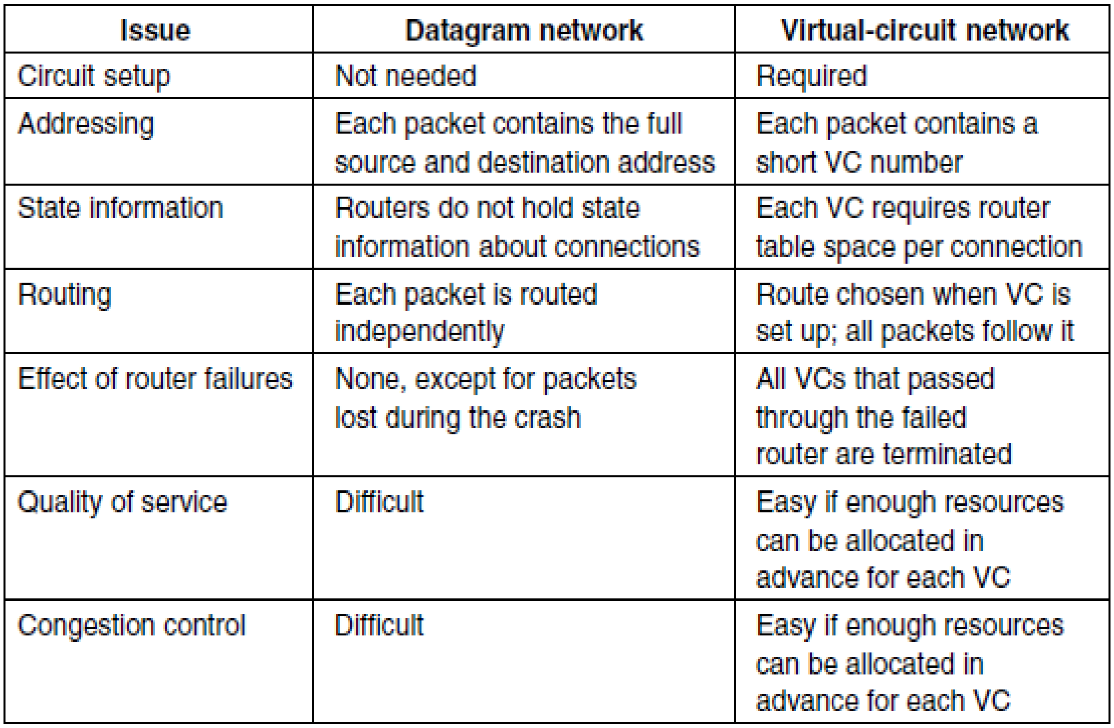
Delay

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 byusing 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

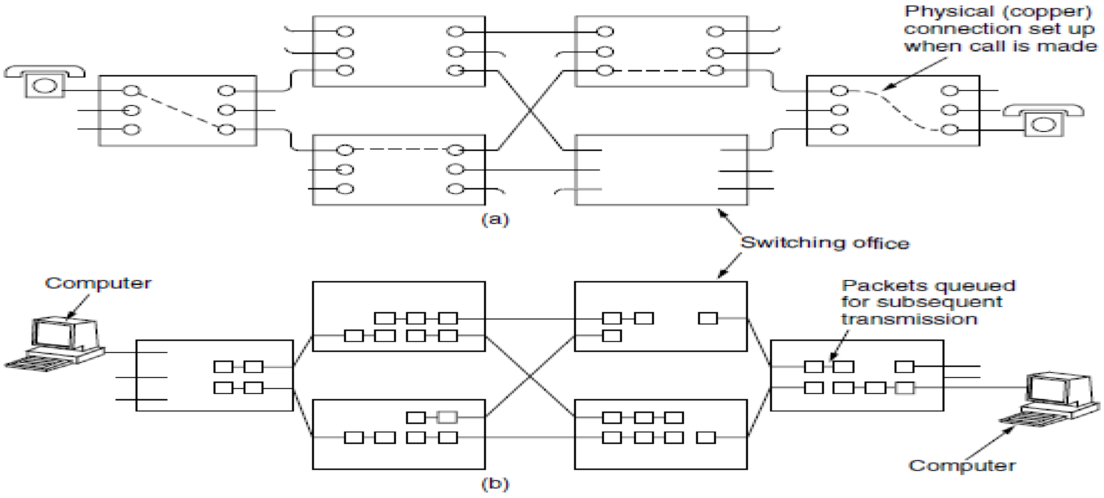

Physical (copper) connection set up when call is made
(a)
Switching office
Computer
Packets queued for subsequent transmission
Computer
(b)

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

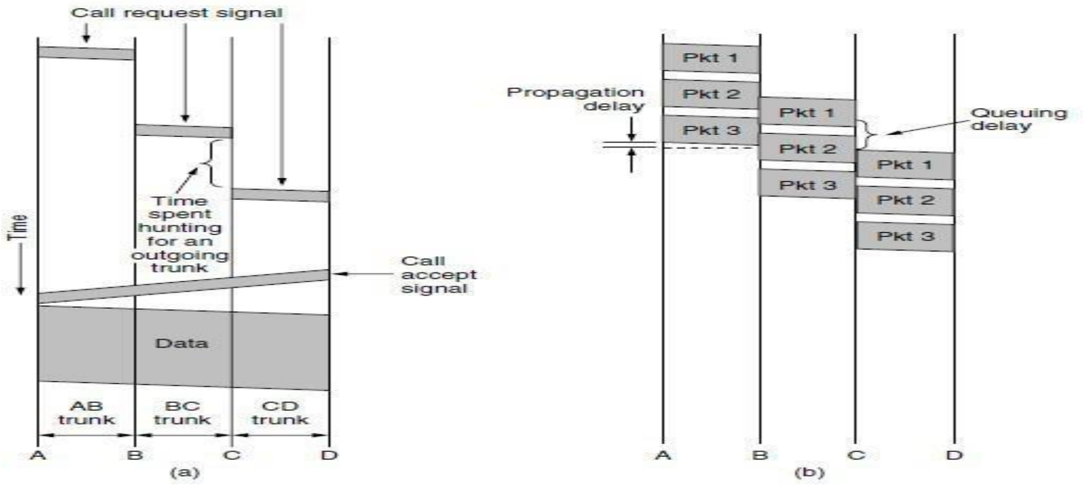
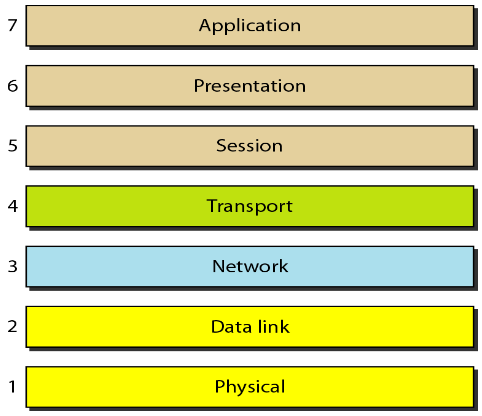


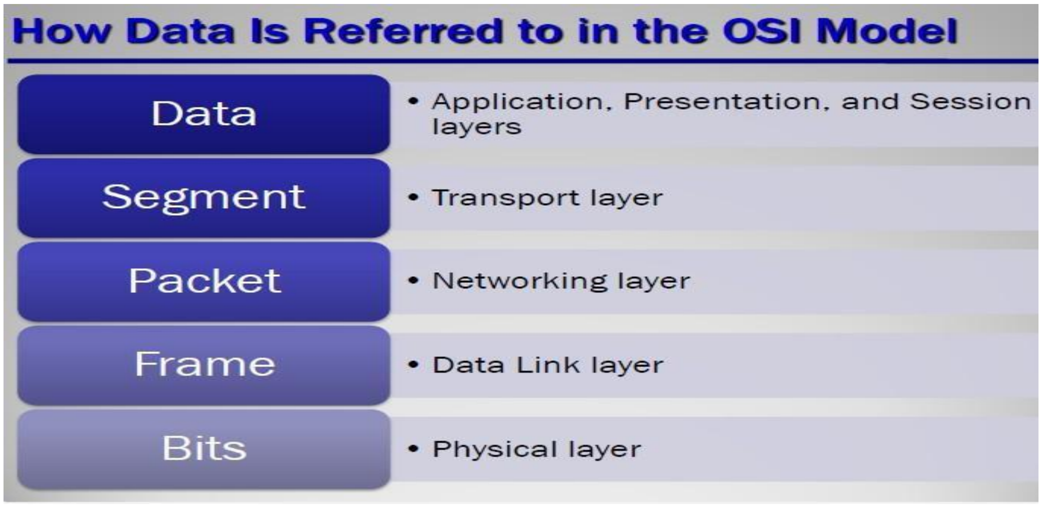
Figure 2-43. Timing of events in (a) circuit switching. (b) packet switching.

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



Top to bottom
–All People Seem To Need Data ProcessingBottom to top
–Please Do Not Throw Sausage Pizza Away



How Data Is Referred to in the OSI Model

Data	• Application, Presentation, and Session layers
Segment	• Transport layer
Packet	• Networking layer
Frame	• Data Link layer
Bits	• Physical layer

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 opticstandards. Copper wiring, fiber optic cable, radio frequencies, anything that can be used totransmit 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 framesacross 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 giventime
    - –Determines where one frame ends and the next one starts, called frame synchronization

Network Layer

- Responsible for moving packets (data) from one end of the network to theother, called *end-to-end communications*
- Requires *logical addresses* such as IP addresses
- Device example: Router
- –Routing is the ability of various network devices and their related software tomove data packets from source to destination

Transport Layer

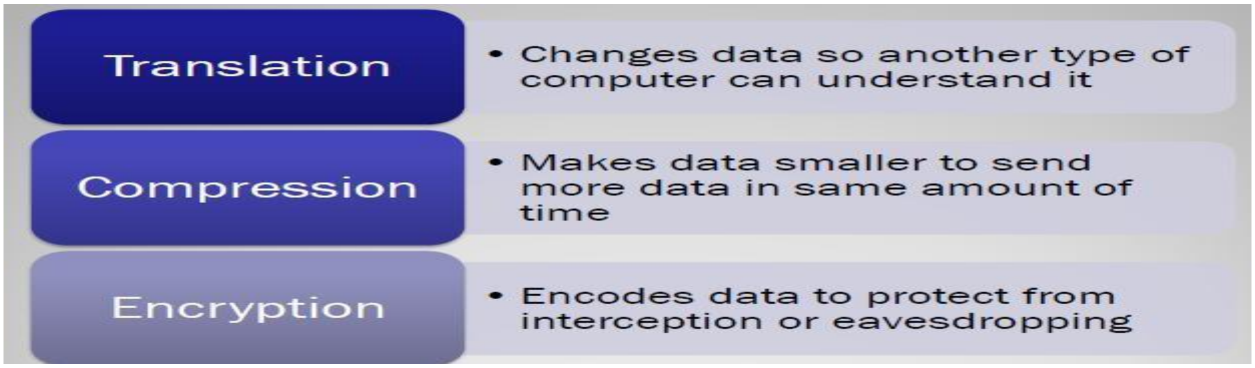
- Takes data from higher levels of OSI Model and breaks it into segments that canbe sent to lower-level layers for data transmission
- Conversely, reassembles data segments into data that higher-level protocols and applications can use
- Also puts segments in correct order (called sequencing) so they can bereassembled 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 destinationreceived 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



Translation
- Changes data so another type of computer can understand it

Compression
- Makes data smaller to send more data in same amount of time

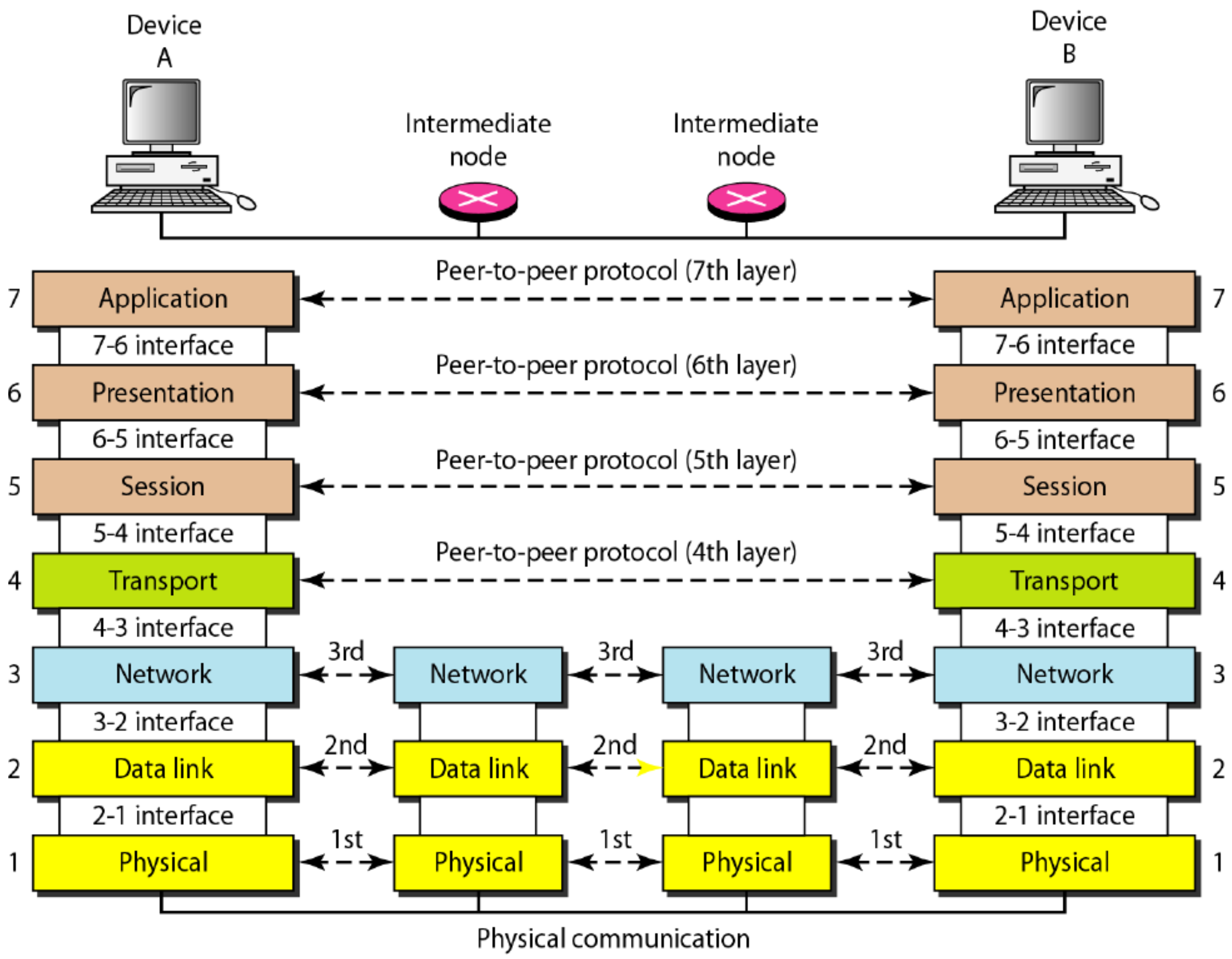
Encryption
- Encodes data to protect from interception or eavesdropping

Application Layer

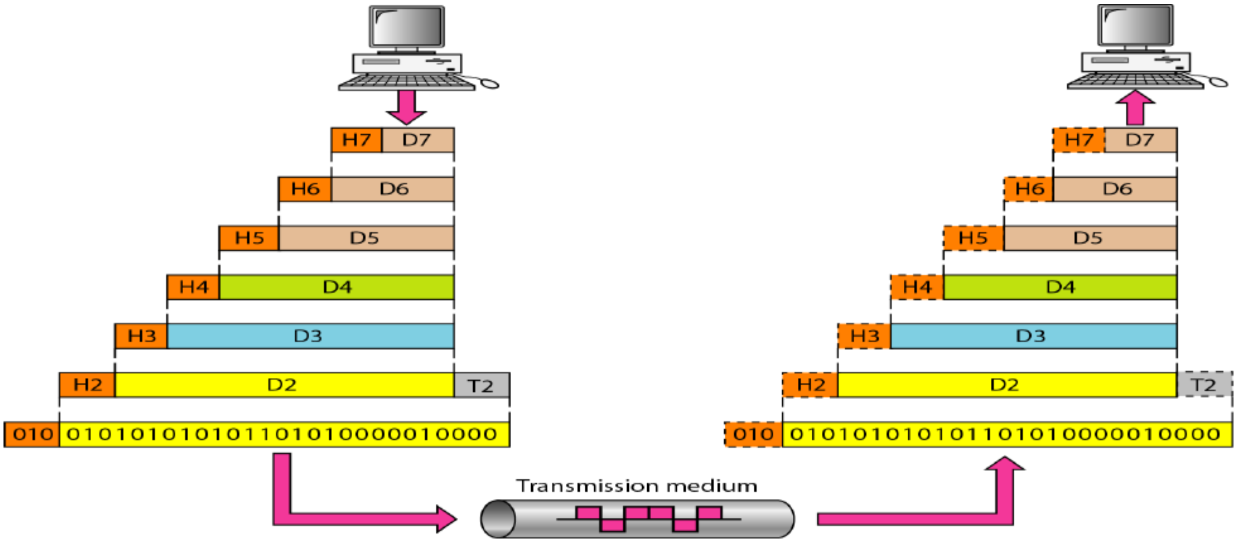
- Contains all services or protocols needed by application software or operatingsystem to communicate on the network
- Examples
  - –Firefox web browser uses HTTP (Hyper-Text Transport Protocol)

- –E-mail program may use POP3 (Post Office Protocol version 3) to read e-mailsand SMTP (Simple Mail Transport Protocol) to send e-mails

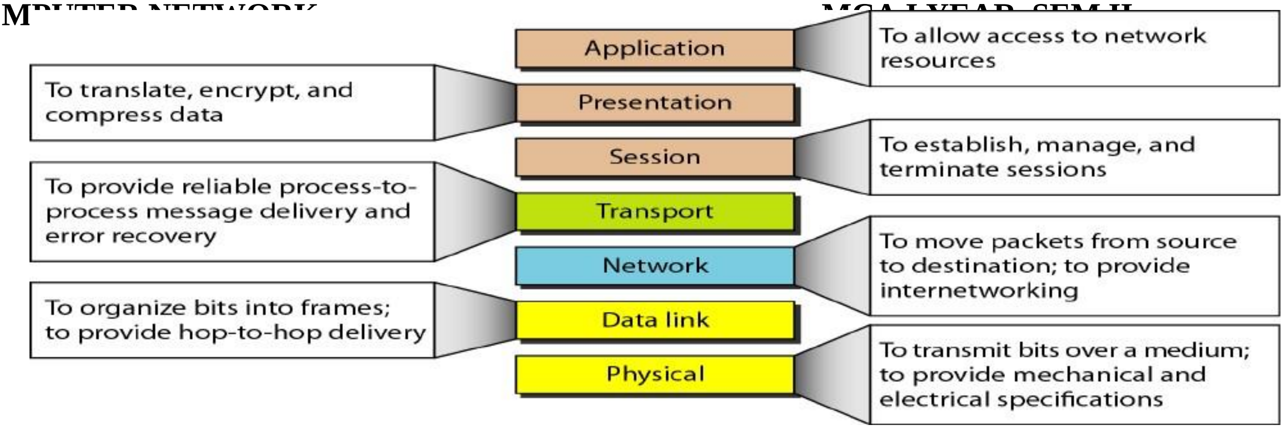
The interaction between layers in the OSI model



An exchange using the OSI model

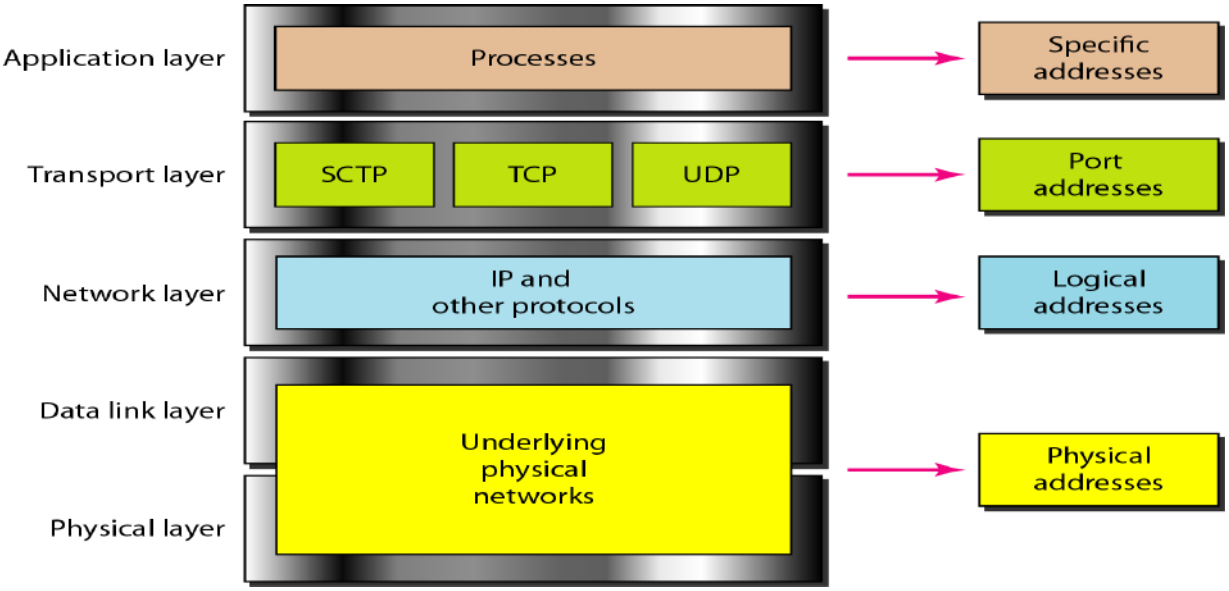


SUMMAR


Application
To allow access to network resources
Presentation
To translate, encrypt, and compress data
Session
To establish, manage, and terminate sessions
Transport
To provide reliable process-to-process message delivery and error recovery
Network
To move packets from source to destination; to provide internetworking
Data link
To organize bits into frames; to provide hop-to-hop delivery
Physical
To transmit bits over a medium; to provide mechanical and electrical specifications

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

–A *protocol suite* is a large number of related protocols that work together toallow 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

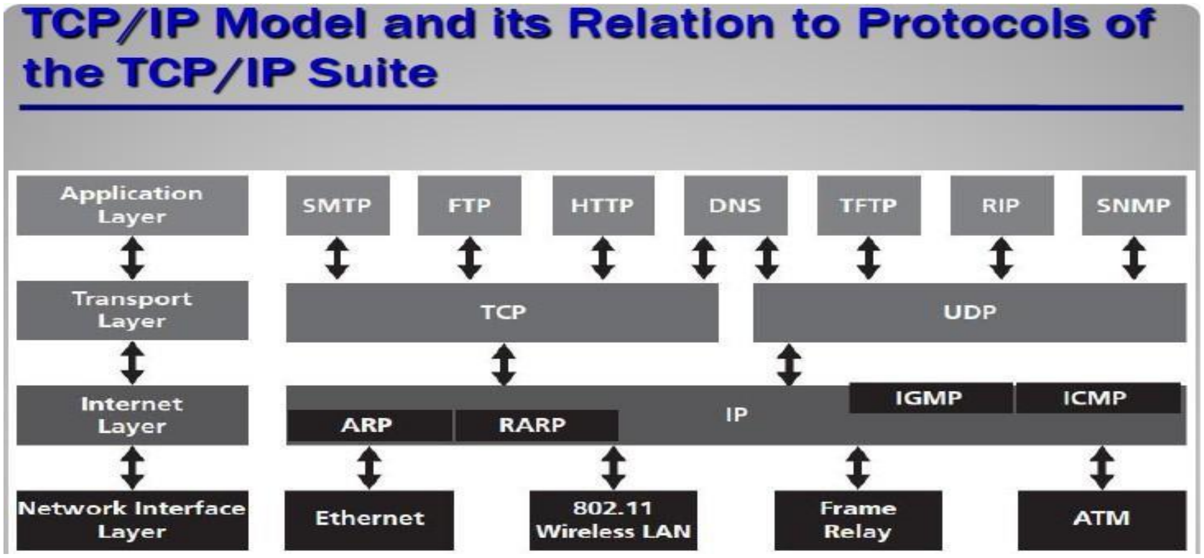
- TCP is a connection-oriented protocol
  - Does not mean it has a physical connection between sender and receiver
  - TCP provides the function to allow a connection virtually exists – also calledvirtual circuit
- UDP provides the functions:
  - Dividing a chunk of data into segments
  - Reassembly 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 betweenthe respective computers

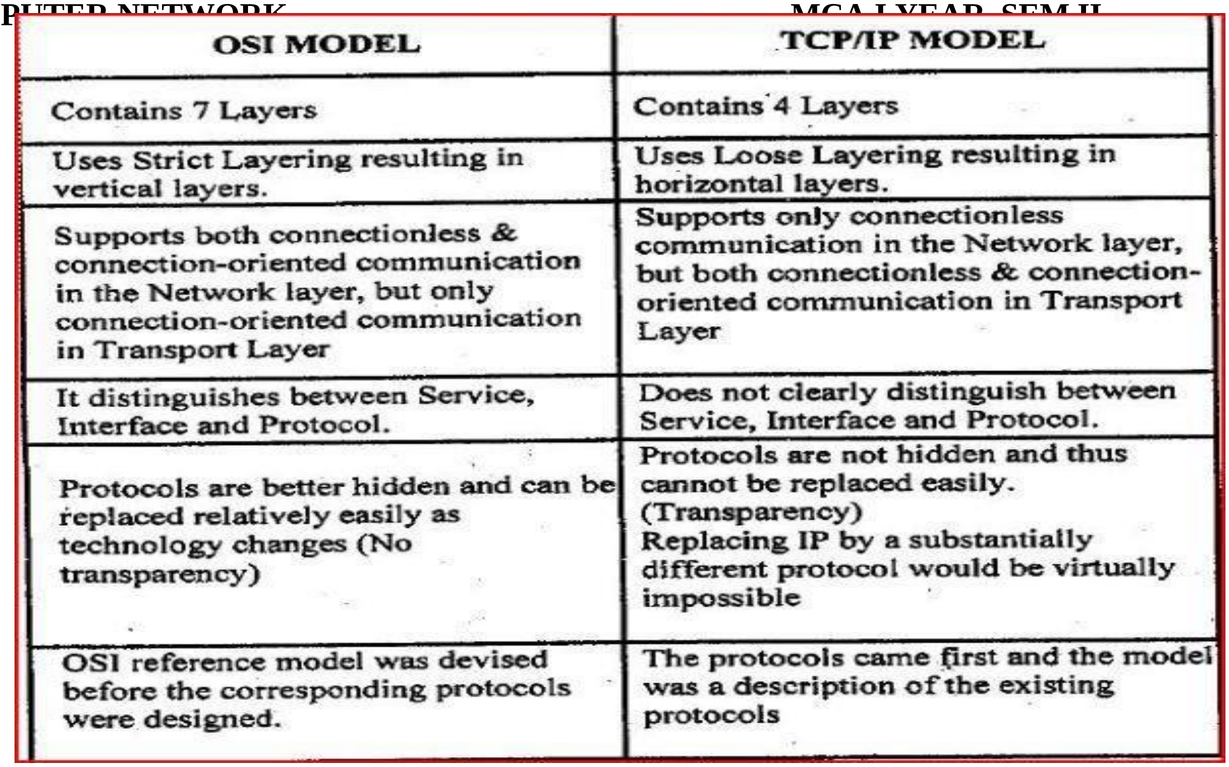
Internet Layer

- 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.





OSI MODEL	TCP/IP MODEL
Contains 7 Layers	Contains 4 Layers
Uses Strict Layering resulting in vertical layers.	Uses Loose Layering resulting in 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

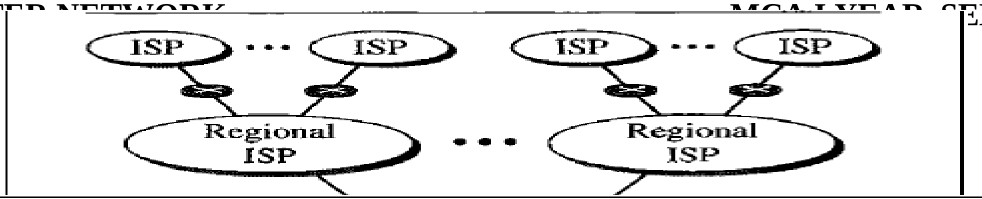
(ARPA) in the Department of Defense (DoD) was interested in finding a way to connect computers so that the researchers they funded could share their findings, thereby reducing costs and eliminating duplication of effort.

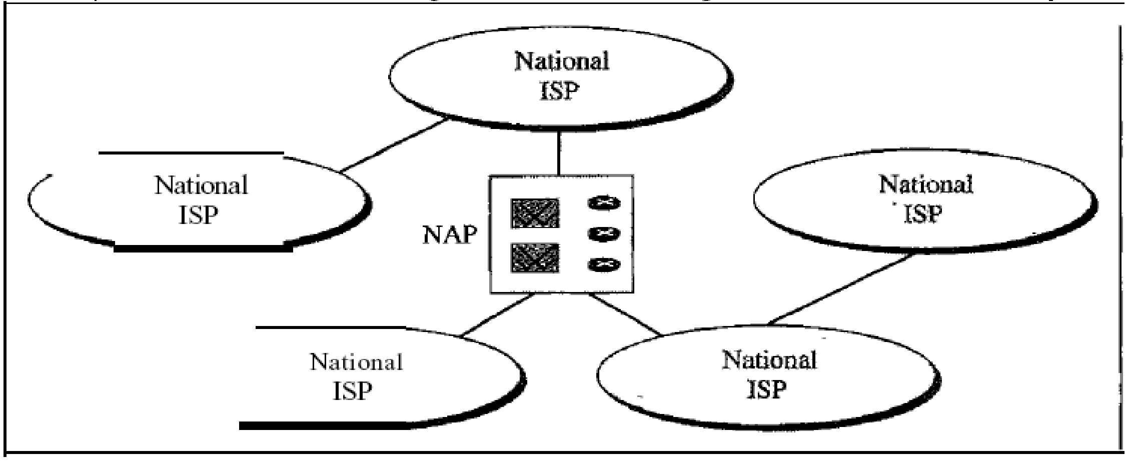
In 1967, at an Association for Computing Machinery (ACM) meeting, ARPA presented its ideas for ARPANET, a small network of connected computers. The idea was that each host computer (not necessarily from the same manufacturer) would be attached to a specialized computer, called an *inteiface message processor* (IMP). The IMPs, in tum, 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 Projec1.* 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 (lP). 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 TCPIIP.

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 (lSPs). 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 witha 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.

UNIT- II

DATA LINK LAYER FUNCTIONS (SERVICES)

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 DataLink 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 ofthe 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 ofthe 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 thenreceiving 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 wherethe 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

service over the links as they have higher error rates and they can be corrected locally, link at which an error occurs rather than forcing toretransmit the data.

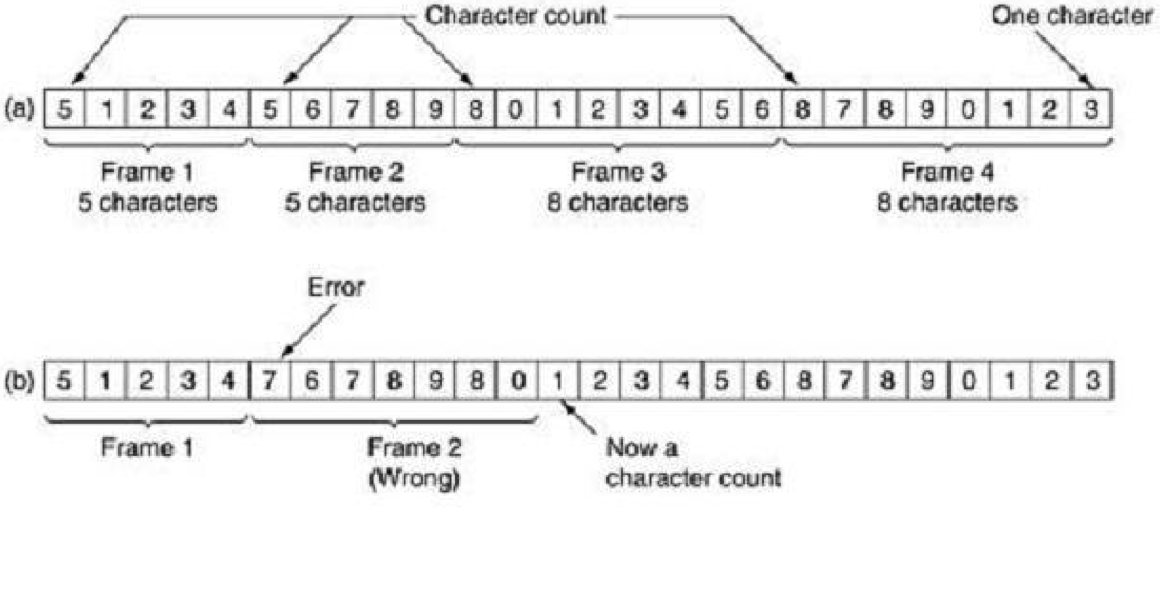
8. **Half-Duplex & Full-Duplex**: In a Full-Duplex mode, both the nodes can transmit the data at the same time. In a Half-Duplex mode, only one node can transmit the data at the same time.

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.

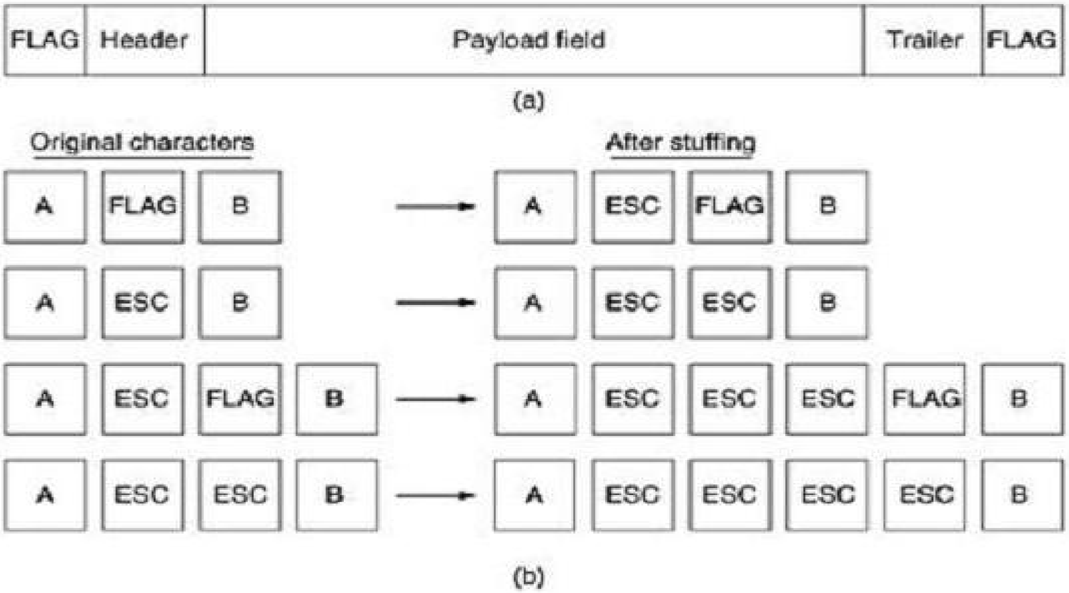
Character count method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is. This technique is shown in Fig. (a) For four frames of sizes 5, 5, 8, and 8 characters, respectively.



A character stream. (a) Without errors. (b) With one error

The trouble with this algorithm is that the count can be garbled by a transmission error. For example, if the character count of 5 in the second 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 wayof 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 layeron the receiving end removes the escape byte before the data are given tothe network layer. This technique is called byte stuffing or character stuffing.

Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it.

What happens if an escape byte occurs in the middle of the data? The answer is that, it too is stuffed with an escape byte. Thus, any single escapebyte 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 exactlythe same as the original byte sequence.

A major disadvantage of using this framing method is that it is closelytied 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 endswith 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.

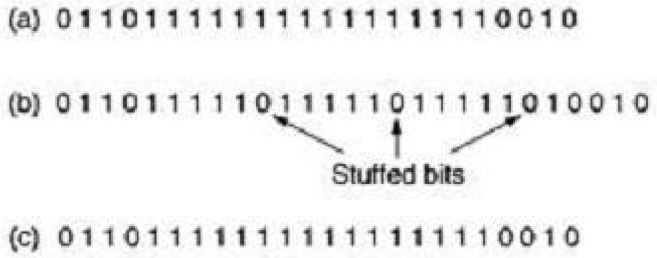
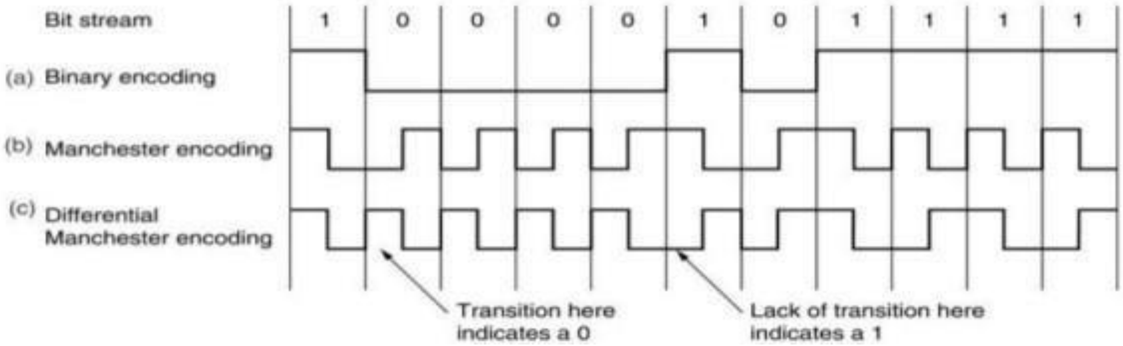

(a) 011011111111111111110010
(b) 011011111011111011111010010
Stuffed bits
(c) 011011111111111111110010

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.

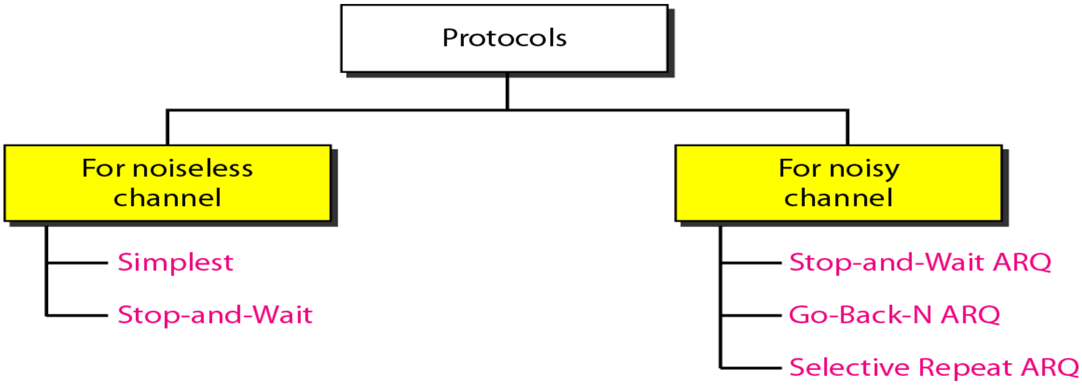
With bit stuffing, the boundary between two frames can be unambiguously recognized by the flag pattern. Thus, if the receiver loses track of where it is, all it has to do is scan the input for flag sequences, since they can only occurat frame boundaries and never within the data.

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 physicalbits. 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-

high and low-low are not used for data but are used for delimiting frames insome protocols.

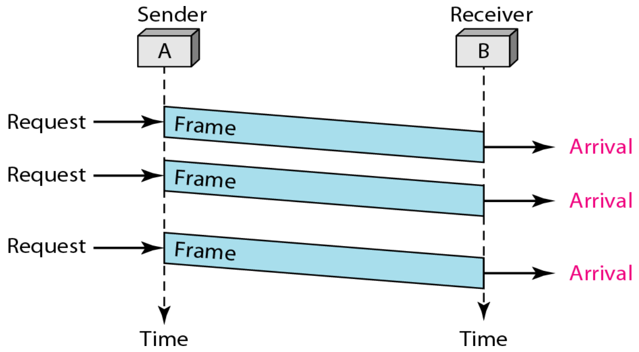

Bit stream
1
0
0
0
0
1
0
1
1
1
1
(a) Binary encoding
(b) Manchester encoding
(c) Differential Manchester encoding
Transition here indicates a 0
Lack of transition here indicates a 1

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 isthe frame accepted as valid. Otherwise, the input stream is scanned for the next delimiter



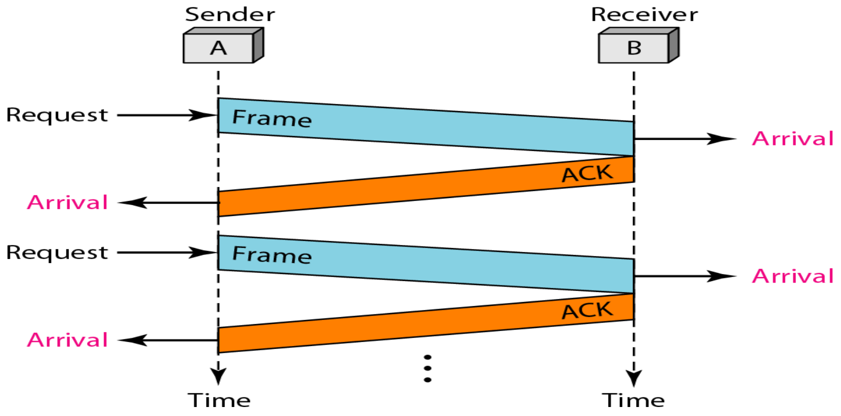
ELEMENTARY DATA LINK PROTOCOLS

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 sendsthe 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 ignorethe error (as we sometimes do), or we need to add error control to ourprotocols. We discuss three protocols in this section that use error control.

Sliding Window Protocols:

1 Stop-and-Wait Automatic Repeat Request

2 Go-Back-N Automatic Repeat Request

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.

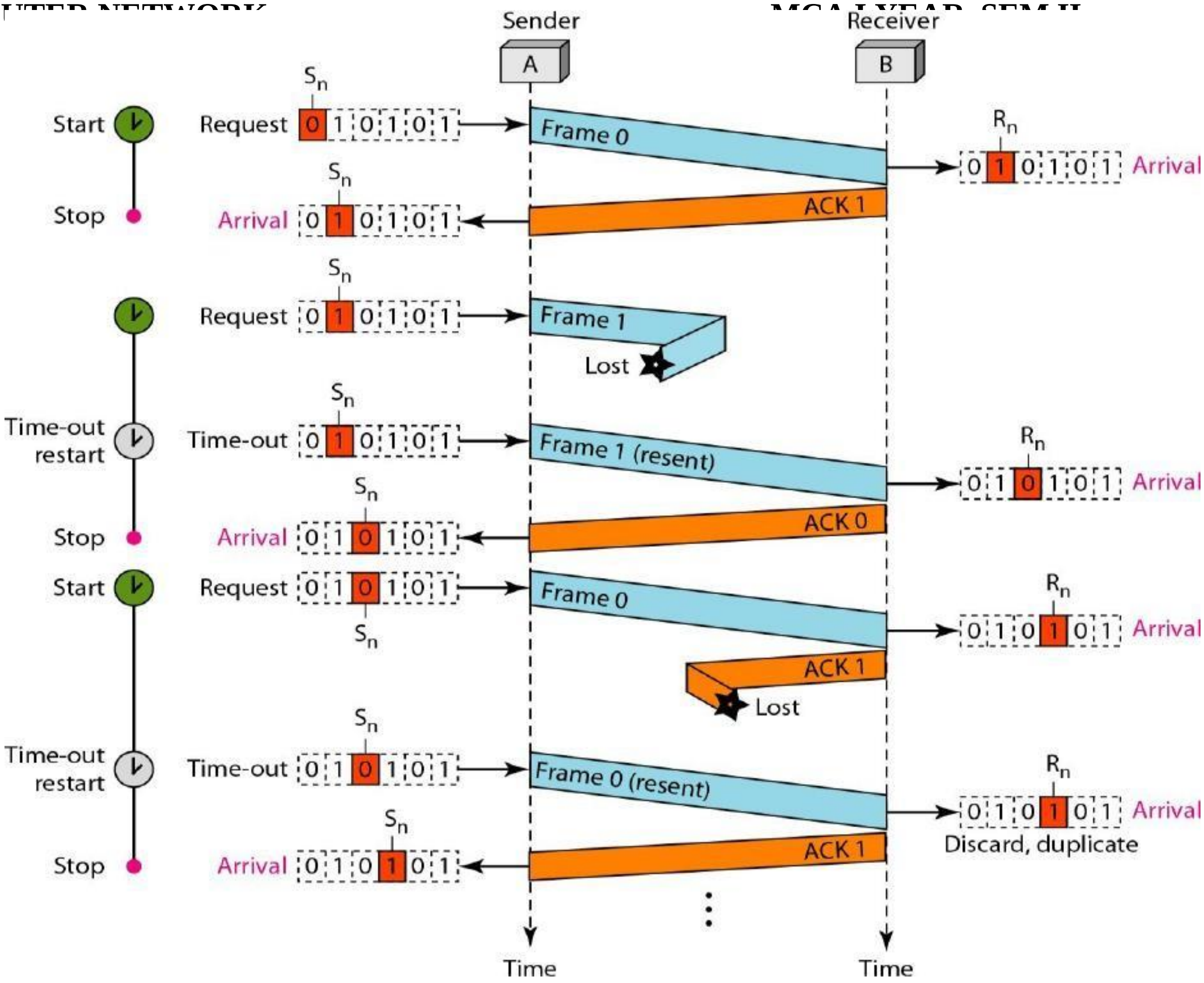
Lost frames are more difficult to handle than corrupted ones. In our previous protocols, there was no way to identify a frame. The received frame could be the correct one, or a duplicate, or a frame out of order. The solution is to number the frames. When the receiver receives a data frame that is out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

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

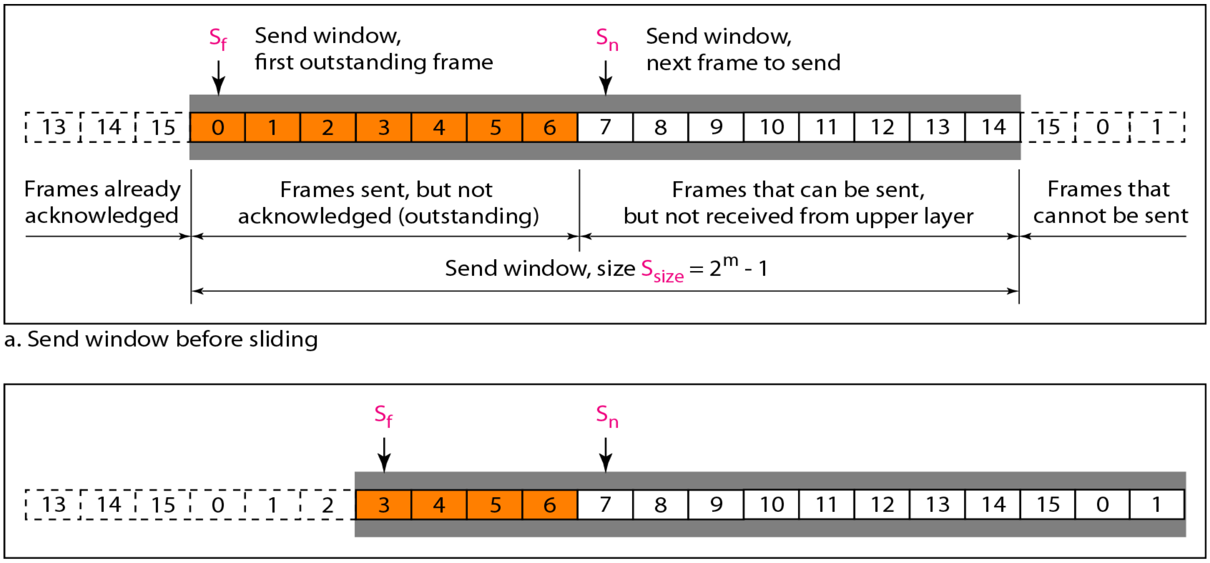
The link utilization is only 1000/20,000, or 5 percent. For this reason, for a link with a high bandwidth or long delay, the use of Stop-and-Wait ARQ wastes the capacity of the link.

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 thechannel busy while the sender is waiting for acknowledgment.

The first is called Go-Back-N Automatic Repeat. In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive.

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 power m*- 1. For example, if *m* is 4, the only sequence numbers are 0 through 15 inclusive.


S_f
Send window, first outstanding frame
S_n
Send window, next frame to send
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already acknowledged
Frames sent, but not acknowledged (outstanding)
Frames that can be sent, but not received from upper layer
Frames that cannot be sent
Send window, size $S_{size} = 2^m - 1$
S_f
S_n
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1

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 sequencenumbers for frames that can be sent; however, the corresponding data packets have not yet been received from the network layer.

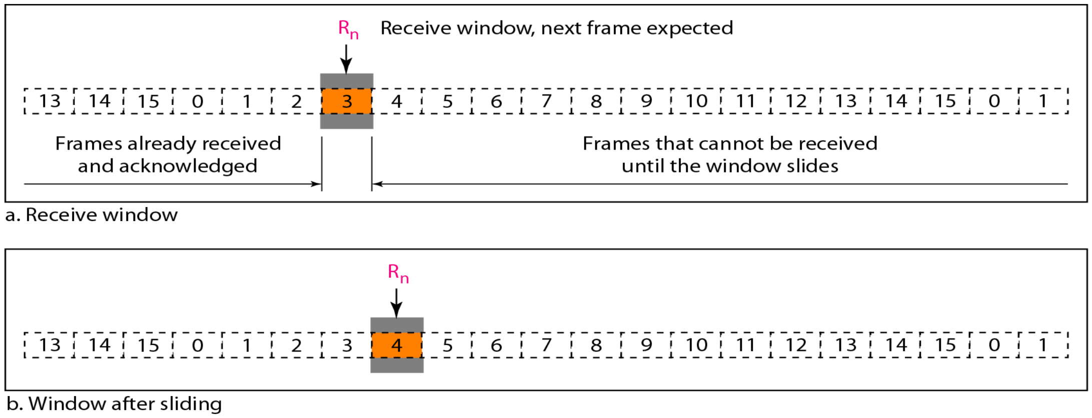
Finally, the fourth region defines sequence numbers that cannot be used until the window slides

The send window is an abstract concept defining an imaginary box ofsize $2^m - 1$ with three variables: S_f, S_n, and S_{size}. The variable *Sf* definesthe sequence number of the first (oldest) outstanding frame. The variable *Sn* holds the sequence number that will be assigned to the next frame to be sent. Finally, the variable Ssize 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, I, and 2 are acknowledged, so the window has slide to the right three slots. Note that thevalue of *Sf* is 3 because frame 3 is now the first outstanding frame.**The send window can slide one or more slots when a valid acknowledgment arrives.**

Receiver window: variable *Rn* (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 asequence number matching the value of *Rn* 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 1with one single variable Rn. The window slides when a correct frame has arrived; sliding occurs one slot at a time


R_n
Receive window, next frame expected
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already received and acknowledged
Frames that cannot be received until the window slides
R_n
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1

a. Receive window

b. Window after sliding

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 weuse 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.

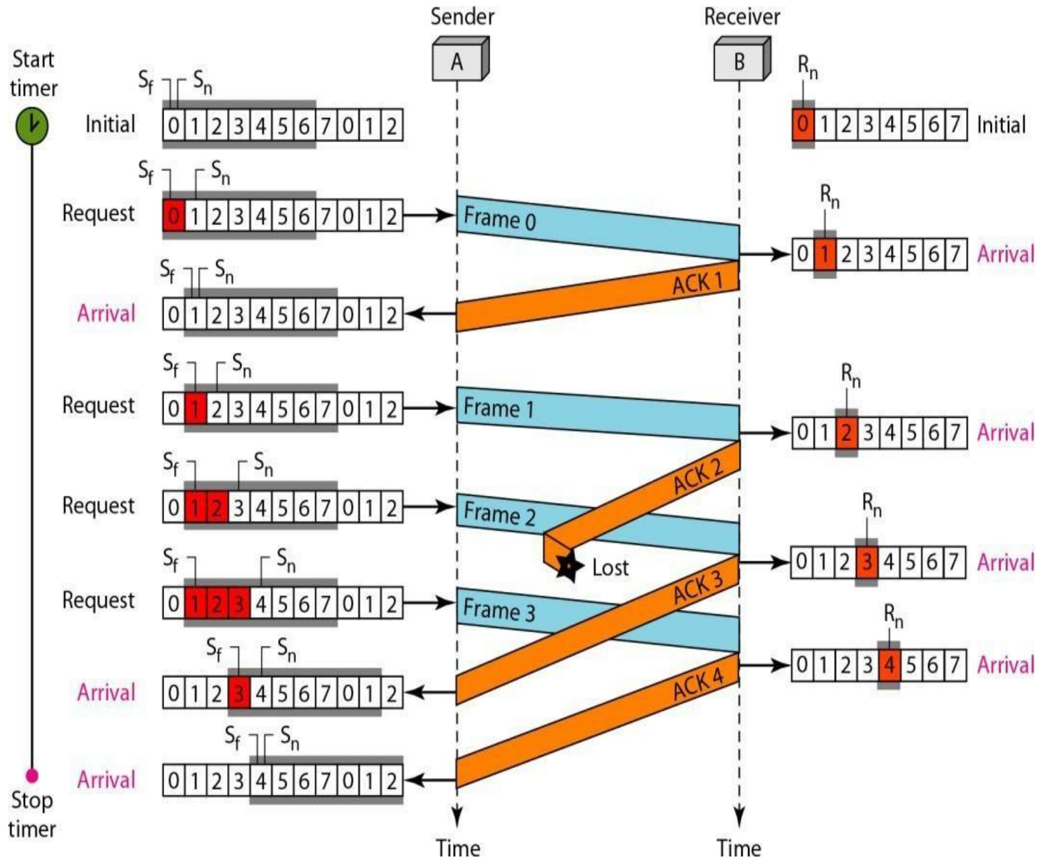
Acknowledgment

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.

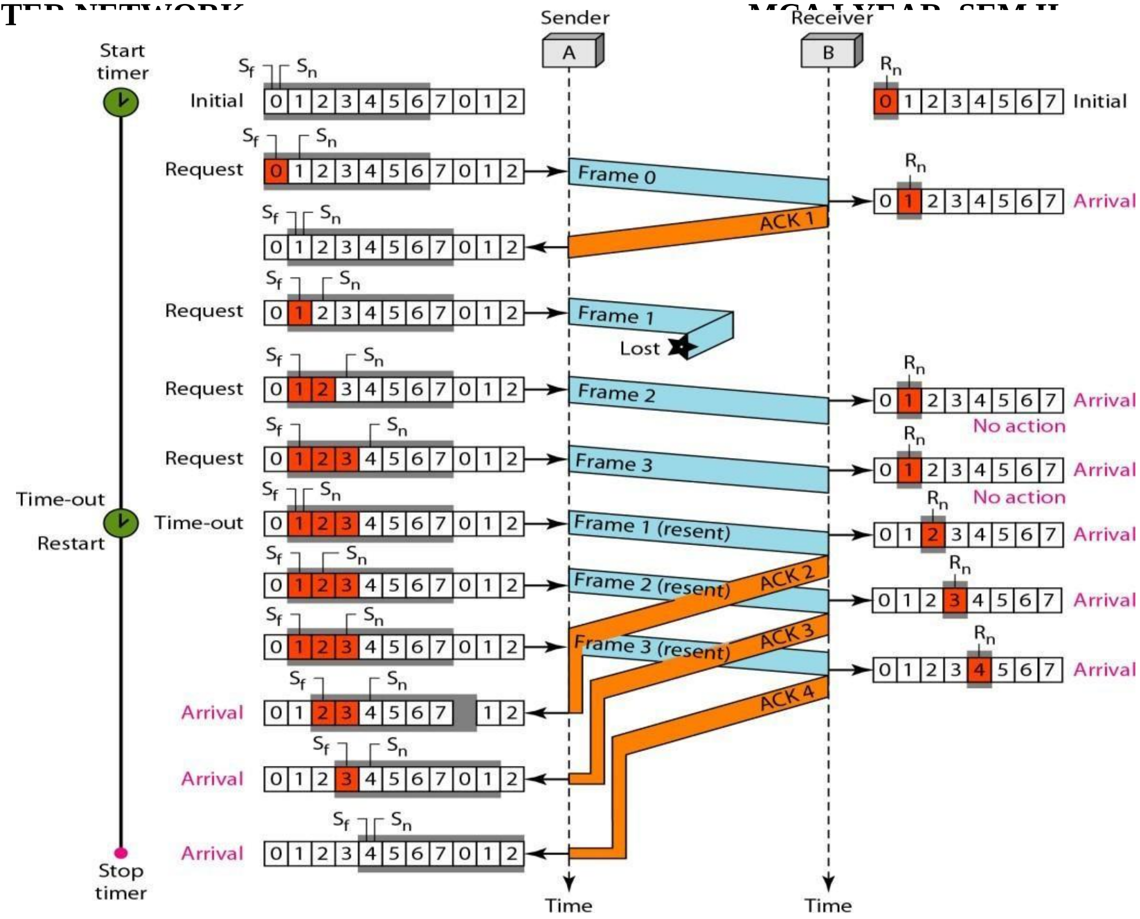
Resending a Frame

When the timer expires, the sender resends all outstanding frames. For example, suppose the sender has already sent frame 6, but the timer forframe 3 expires. This means that frame 3 has not been acknowledged; the sender goes back and sends frames 3,4,5, and 6 again. That is why the protocol is called *Go-Back-N* ARQ.

Below figure is an example(if ack lost) of a case where the forward channel isreliable, but the reverse is not. No data frames are lost, but some ACKs are delayed and one is lost. The example also shows how cumulative acknowledgments can help if acknowledgments are delayed or lost



Below figure is an example(if frame lost)



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

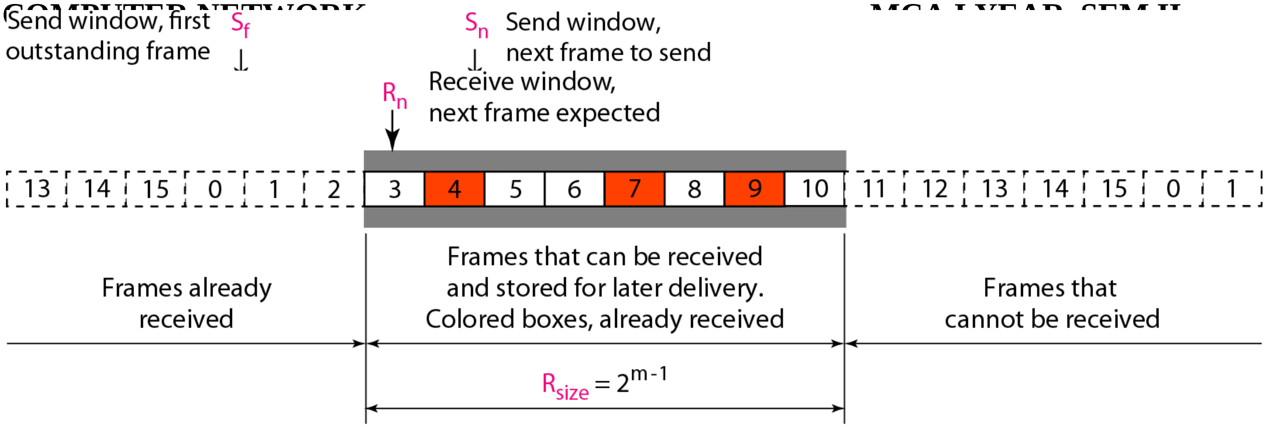
In Go-Back-N ARQ, The receiver keeps track of only one variable, and there is no need to buffer out-of- order frames; they are simply discarded. However, this protocol is very inefficient for a noisy link.

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)


Send window, first outstanding frame
S_f
S_n
Send window, next frame to send
R_n
Receive window, next frame expected
13
14
15
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0
1
Frames already received
Frames that can be received and stored for later delivery. Colored boxes, already received
Frames that cannot be received
$R_{size} = 2^{m-1}$

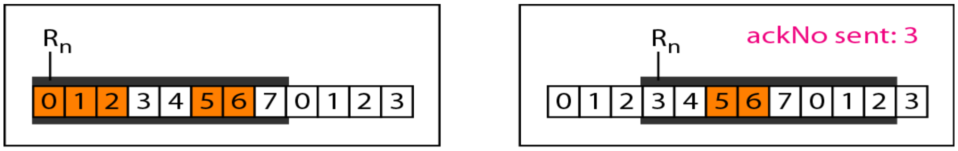
Receiver window

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 thesend 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 thesend window and receive window are the same, all the frames in the sendframe 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

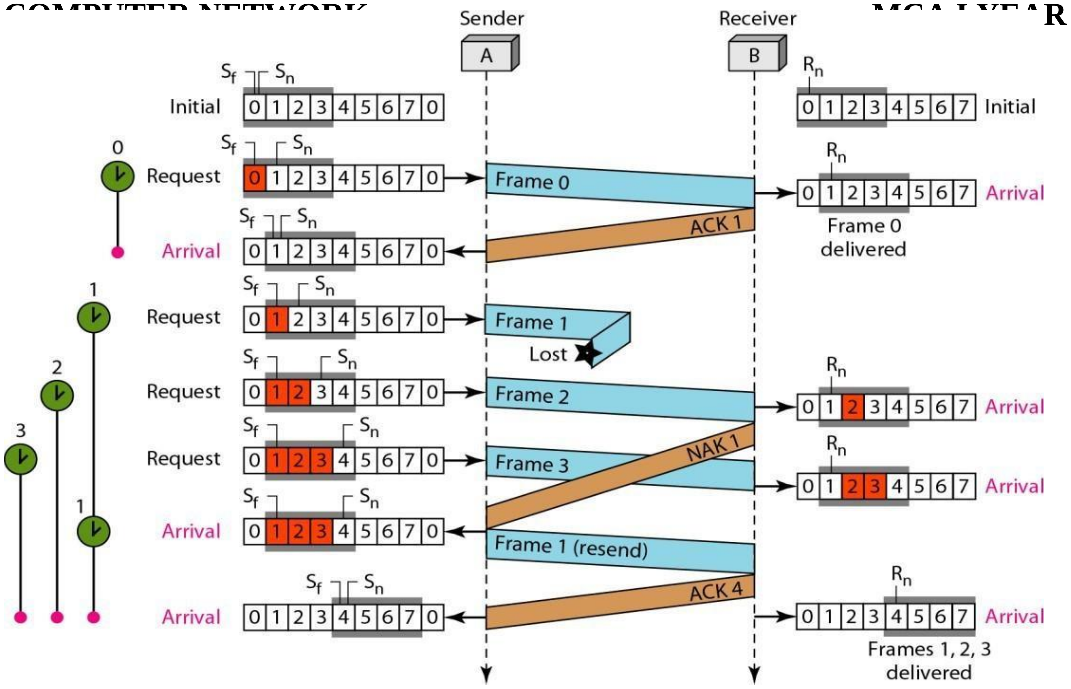
Delivery of Data in Selective Repeat ARQ:



a. Before delivery

b. After delivery

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.

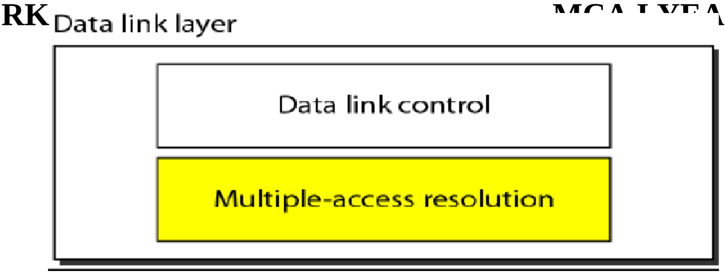
The next point is about the ACKs. Notice that only two ACKs are sent here. The first one acknowledges only the first frame; the second one acknowledges three frames. In Selective Repeat, ACKs are sent when data are delivered tothe network layer. If the data belonging to *n* frames are delivered in one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

Piggybacking

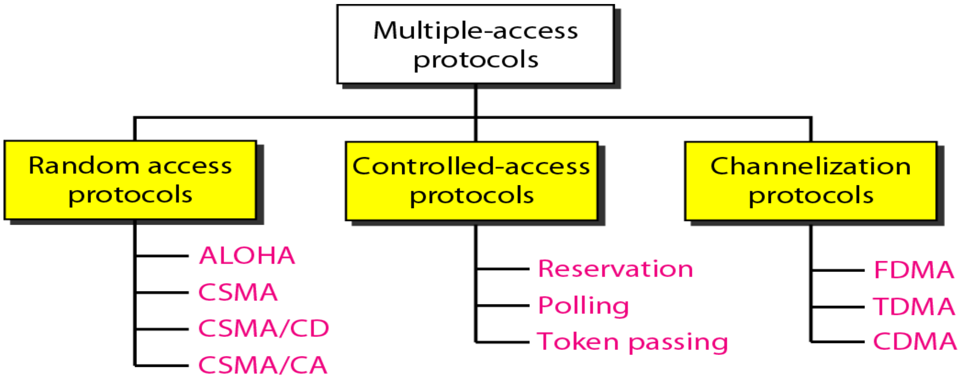
A technique called **piggybacking** is used to improve the efficiency of thebidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frameis carrying data from B to A, it can also carry control information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

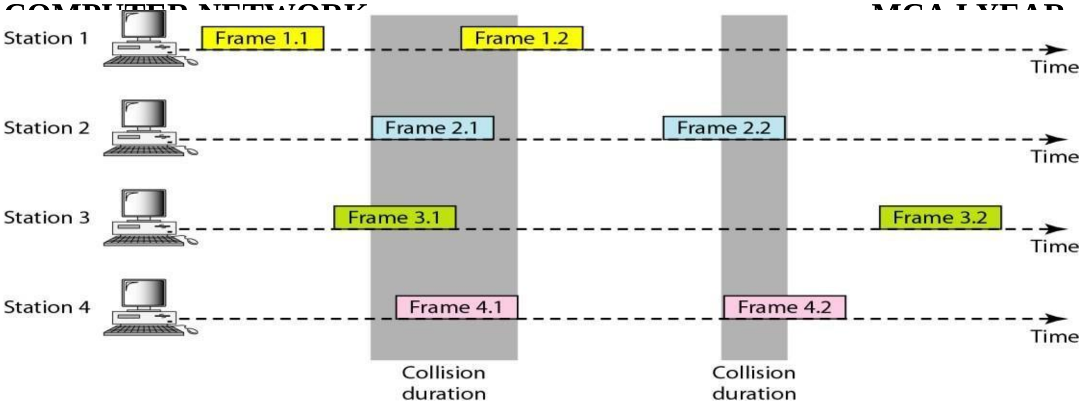
In random access or contention methods, no station is superior to another station and none is assigned the control over another.

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.



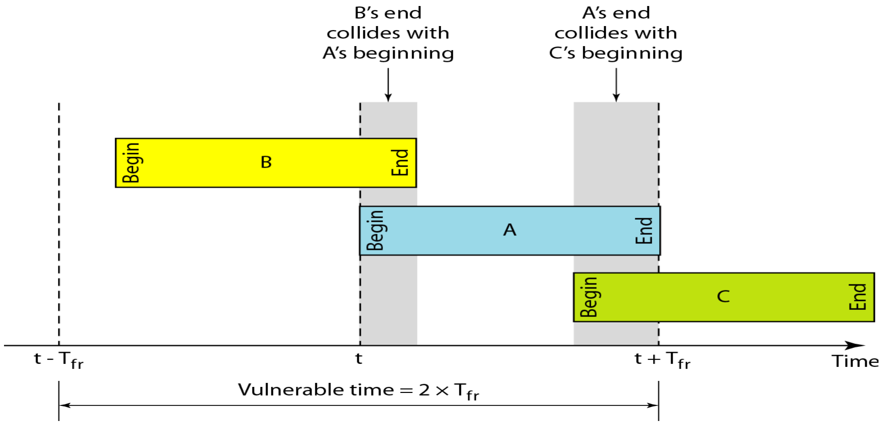
Frames in a pure ALOHA network

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.

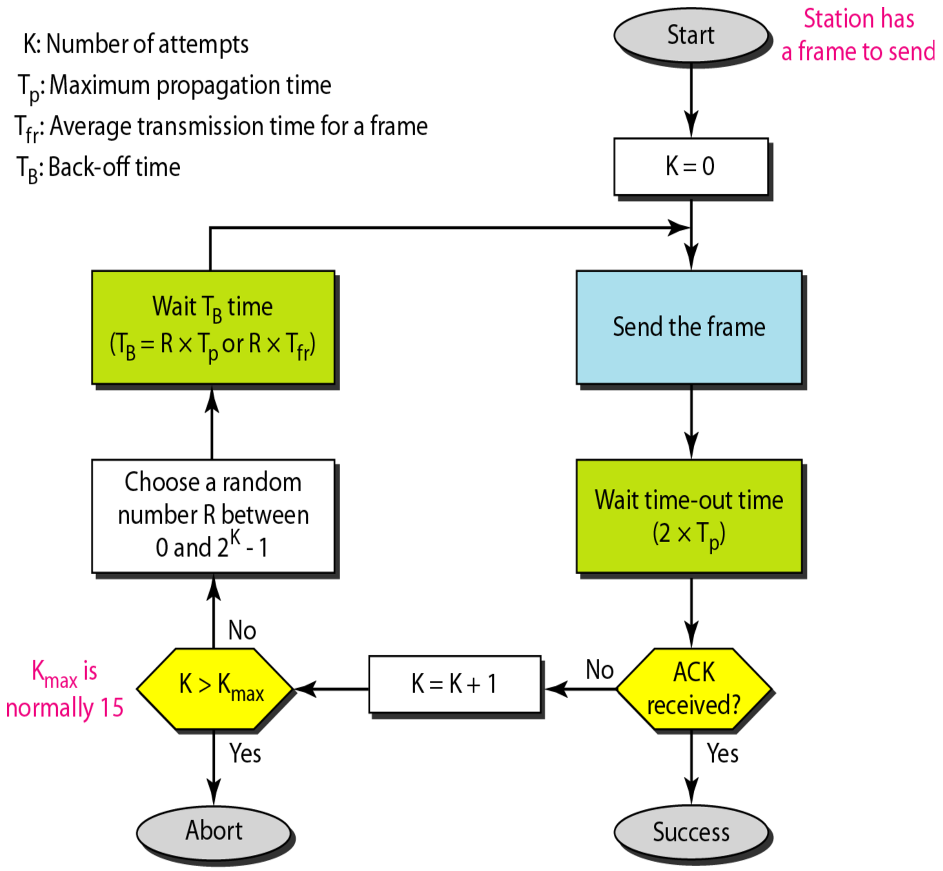
Vulnerable time Let us find the length of time, the **vulnerable time,** in which there is a possibility of collision. We assume that the stations send fixed- length frames with each frame taking *Tfr* S to send. Below Figure shows the vulnerable time for station A.

Station A sends a frame at time *t*. Now imagine station B has already sent a frame between *t* - *T*fr


B's end collides with A's beginning
A's end collides with C's beginning
Begin
B
End
Begin
A
End
Begin
C
End
t - T_fr
t
t + T_fr
Time
Vulnerable time = 2 × 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* + *Tfr* . 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

Looking at Figure, we see that the vulnerable time, during which a collision may occur in pure ALOHA, is 2 times the frame transmission time. Pure ALOHA vulnerable time = 2 x Tfr


K: Number of attempts
T_p: Maximum propagation time
T_{fr}: Average transmission time for a frame
T_B: Back-off time
Start
Station has a frame to send
K = 0
Wait T_B time ($T_B = R \times T_p$ or $R \times T_{fr}$)
Send the frame
Choose a random number R between 0 and $2^K - 1$
Wait time-out time ($2 \times T_p$)
No
K_{max} is normally 15
$K > K_{max}$
K = K + 1
No
ACK received?
Yes
Yes
Abort
Success

Procedure for pure ALOHA protocol

Example

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

Solution

Average frame transmission time *Tfr* is 200 bits/200 kbps or 1 ms. The vulnerable time is 2 x 1 ms =2 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 I-ms period that this station is sending.

The throughput for pure ALOHA is $S = G \times e^{-2G}$. The maximum throughput Smax = 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 millisecond. The load is 1. In this case S = G× $e^{-2\ G}$ or S = 0.135 (13.5 percent). This means that the throughput is 1000 × 0.135 = 135 frames. Only 135 frames out of 1000 will probably survive.

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 × e^{-2G} or S = 0.184 (18.4 percent). This means that the throughput is 500 × 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 × e^{-2G} or S = 0.152 (15.2 percent). This means that the throughput is 250 × 0.152 = 38. Only 38 frames out of 250 will probably survive.

2 Slotted ALOHA

Pure ALOHA has a vulnerable time of 2 x *Tfr* . 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 Tfr s and force the station to send only at the beginning of the time slot. Figure 3 shows an example offrame collisions in slotted ALOHA

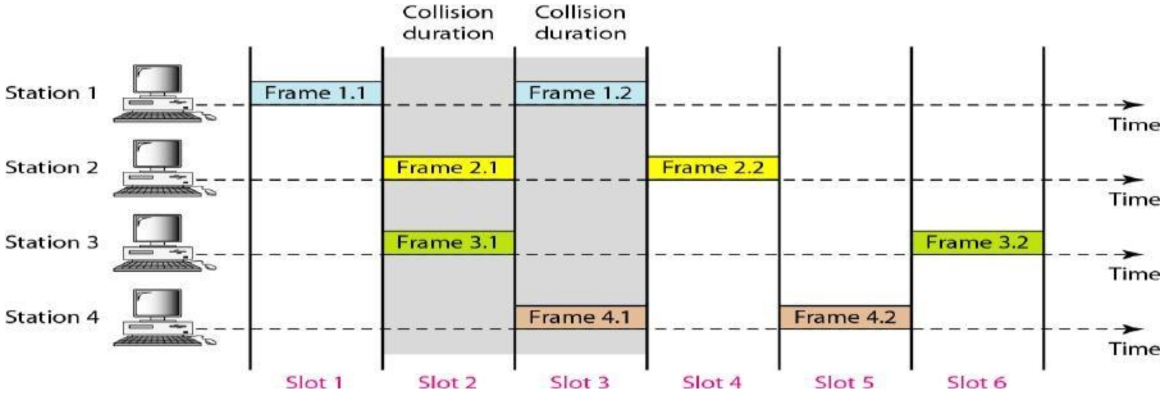
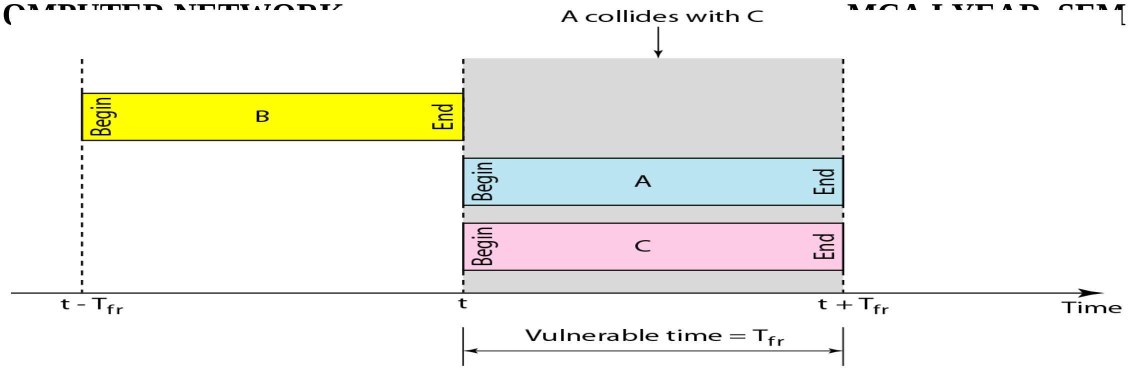


FIG:3

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 beginningof this slot has already finished sending its frame. Of course, there is still thepossibility 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 *Tfr* Figure 4 shows the situation

Below fig shows that the vulnerable time for slotted ALOHA is one-half that ofpure ALOHA. Slotted ALOHA vulnerable time = Tfr



The throughput for slotted ALOHA is S = G × e−G . The maximumthroughput Smax = 0.368 when G = 1.

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

a. 1000 frames per second b. 500 frames per second c. 250 frames per second

Solution

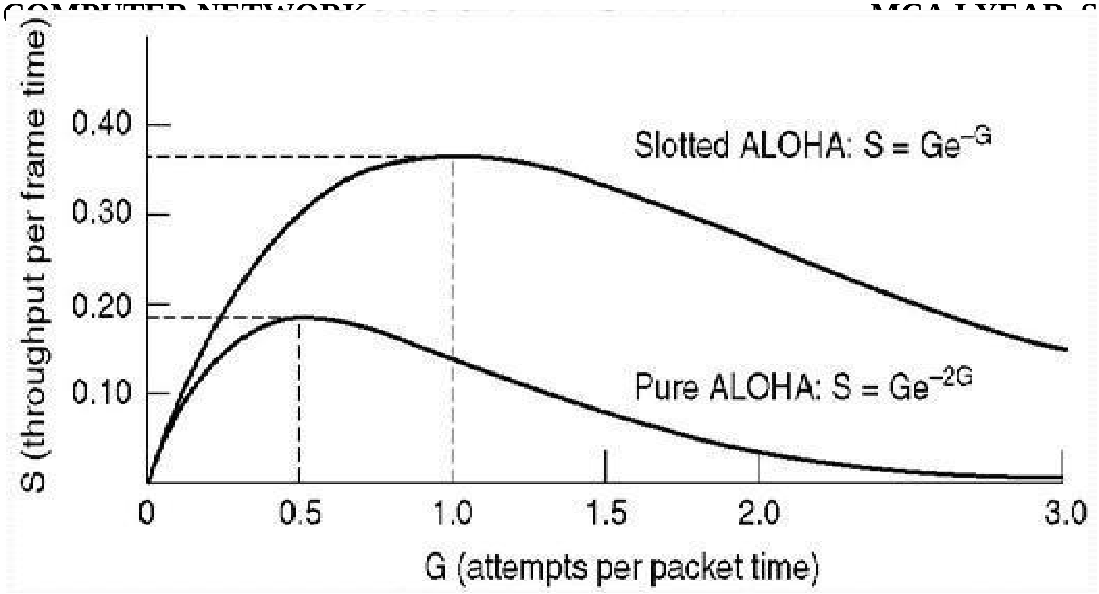
This situation is similar to the previous exercise except that the network is using slotted ALOHA instead of pure ALOHA. The frame transmission time is*200/200* kbps or 1 ms.

a. In this case G is 1. So S =G x *e-G* or *S* =0.368 (36.8 percent). This means that the throughput is 1000 x 0.0368 =368 frames. Only 368 out of 1000 frames will probably survive. Note that this is the maximum throughput case,percentagewise.

b. Here G is 1/2 In this case S =G x *e-G* or S =0.303 (30.3 percent). This means that the throughput is 500 x 0.0303 =151. Only 151 frames out of 500 will probably survive.

c. Now G is 1/4. In this case S =G x *e-G* or S =0.195 (19.5 percent). This means that the throughput is 250 x 0.195 = 49. Only 49 frames out of 250 will probably survive

Comparison between Pure Aloha & Slotted Aloha



Carrier Sense Multiple Access (CSMA)

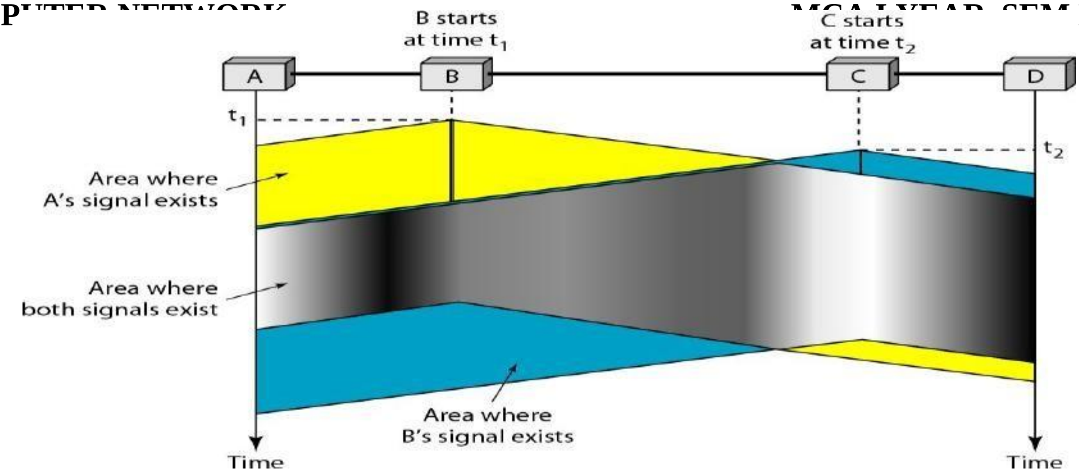
To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it. Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check 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 *tI'* station B senses the medium and finds it idle, so it sends a frame. At time *t2 (t2> tI)'* station C senses the medium and finds it idle because, at this time, the first bits from station B

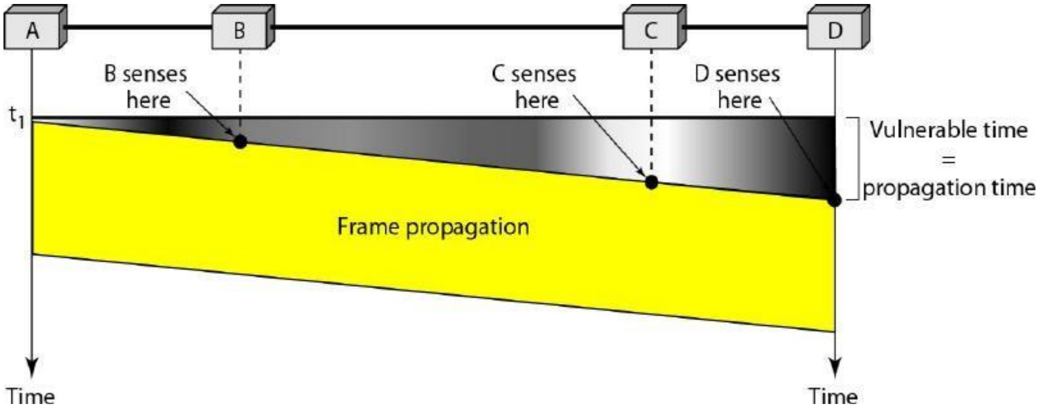
have not reached station C. Station C also sends a frame. The two signals collide and both frames are destroyed.



Space/time model of the collision in CSMA

Vulnerable Time

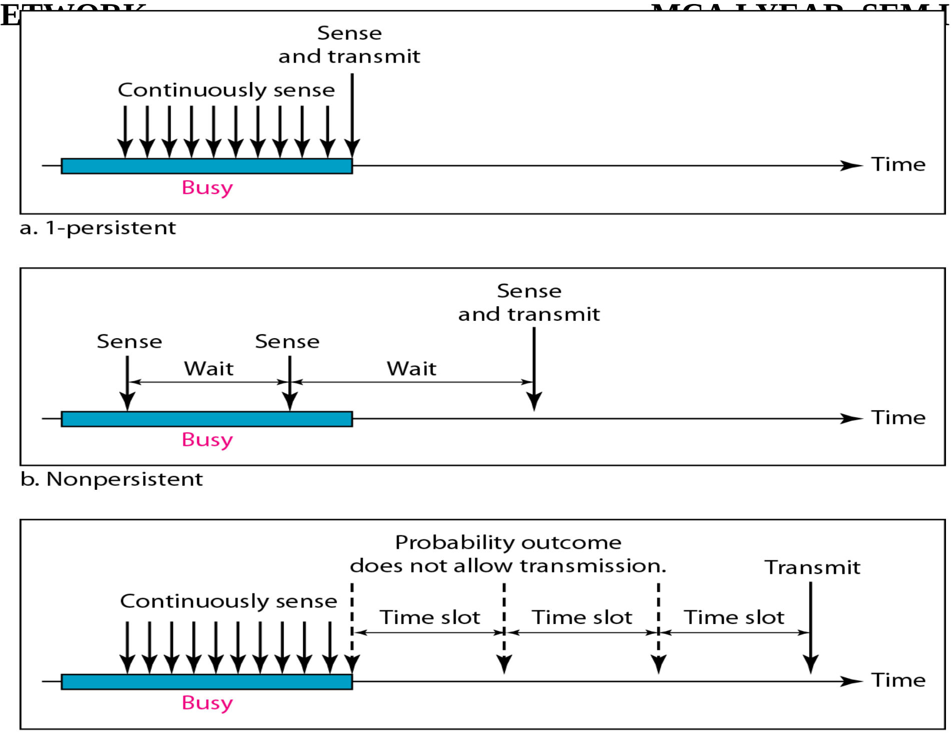
The vulnerable time for CSMA is the propagation time Tp . This is the time needed for a signal to propagate from one end of the medium to the other. When a station sends a frame, and any other station tries to send a frameduring this time, a collision will result. But if the first bit of the frame reaches 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



a. 1-persistent

b. Nonpersistent

c. p-persistent

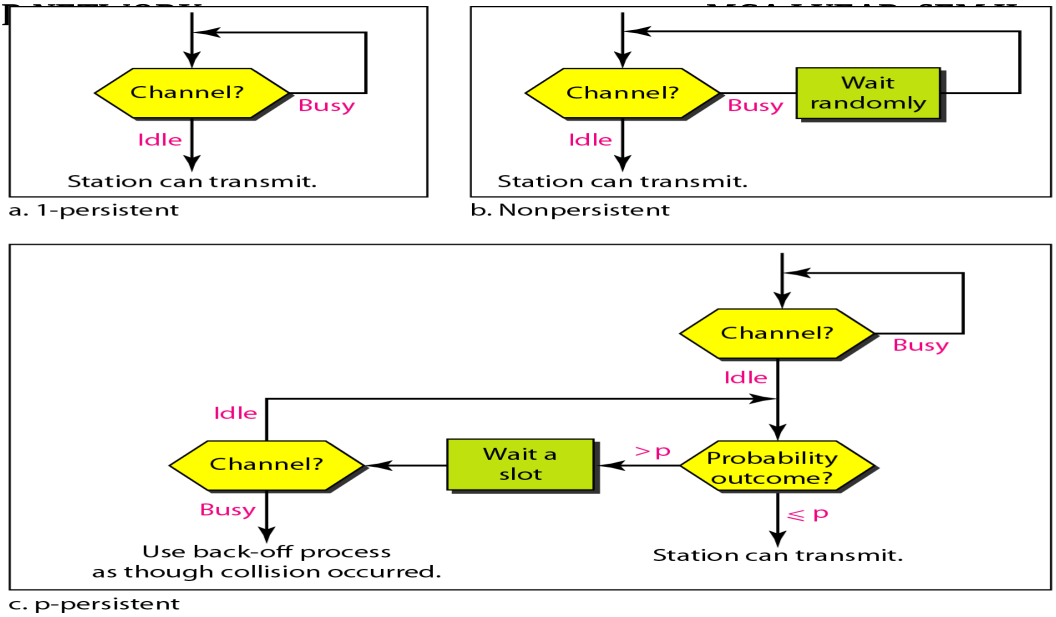
1-Persistent: In this method, after the station finds the line idle, it sends its frame immediately (with probability 1). This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.

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.



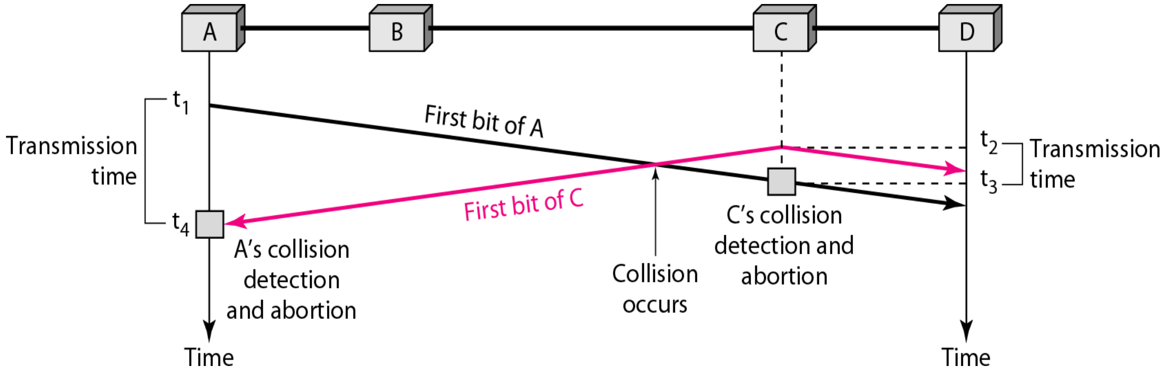
a.

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.



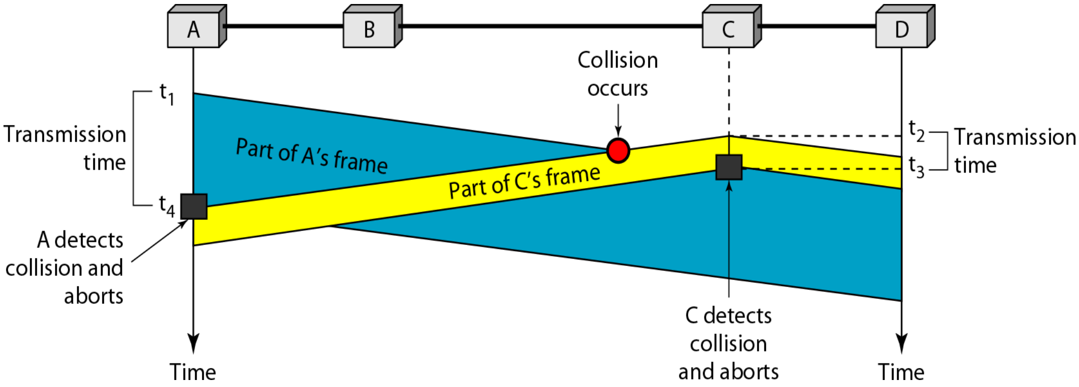
Collision of the first bit in CSMA/CD

At time *t* 1, station A has executed its persistence procedure and starts sending the bits of its frame. At time *t2,* station C has not yet sensed the first bit sent by A. Station C executes its persistence procedure and starts sendingthe bits in its frame, which propagate both to the left and to the right. The collision occurs sometime after time *t2.*Station C detects a collision at time *t3*when it receives the first bit of A's frame. Station C immediately (or after a short time, but we assume immediately) aborts transmission.

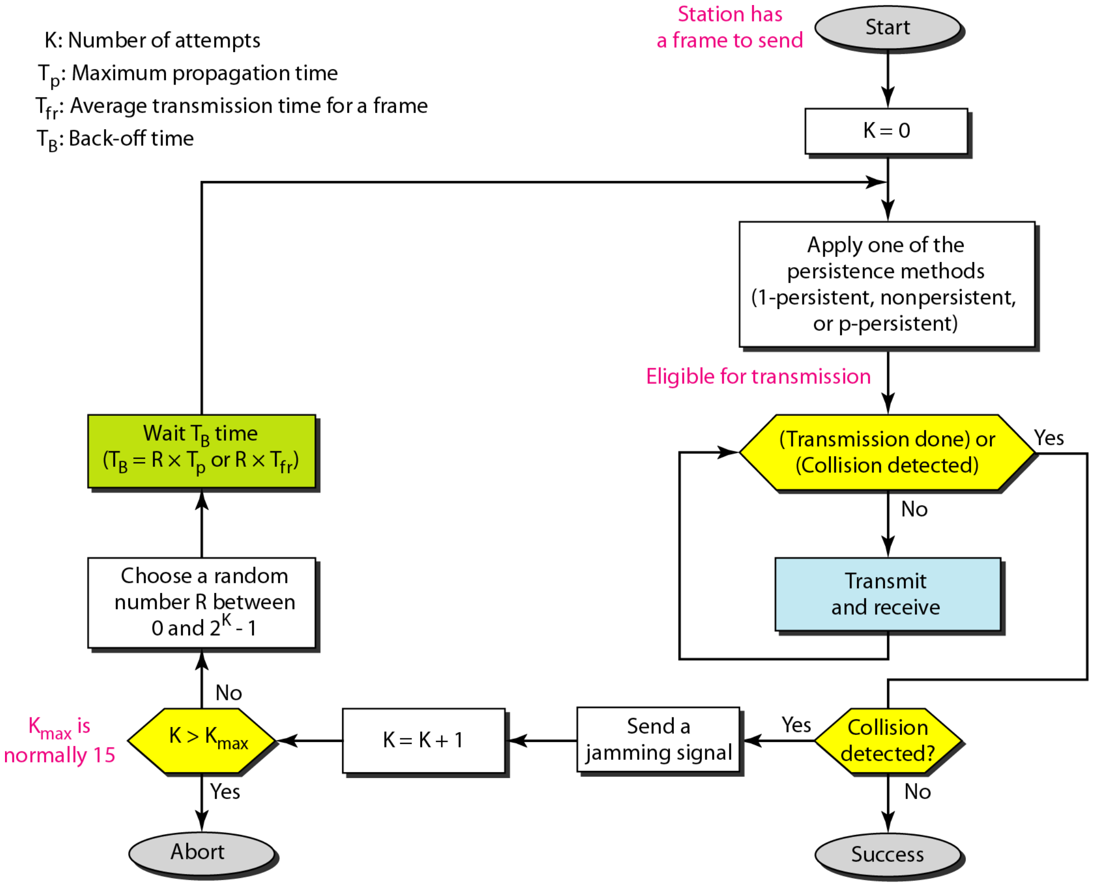
Station A detects collision at time *t4* when it receives the first bit of C's frame; it also immediately aborts transmission. Looking at the figure, we see that A transmits for the duration *t4 - tl;* C transmits for the duration *t3 - t2.*

Minimum Frame Size

For *CSMAlCD* to work, we need a restriction on the frame size. Before sending 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 *Tp*. 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 *Tp* toreach the second, and the effect of the collision takes another time *Tp* to reach the first. So the requirement is that the first station must still be transmitting after *2Tp* .



Collision and abortion in CSMA/CD


K: Number of attempts
T_p: Maximum propagation time
T_{fr}: Average transmission time for a frame
T_B: Back-off time
Station has a frame to send
Start
K = 0
Apply one of the persistence methods (1-persistent, nonpersistent, or p-persistent)
Eligible for transmission
(Transmission done) or (Collision detected)
Yes
No
Transmit and receive
Collision detected?
Yes
No
Success
Send a jamming signal
K = K + 1
$K > K_{max}$
K_{max} is normally 15
Yes
Abort
No
Choose a random number R between 0 and $2^K - 1$
Wait T_B time ($T_B = R \times T_p$ or $R \times T_{fr}$)

Flow diagram for the CSMA/CD

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 μs, what is the minimum size of the frame?

SOL

The frame transmission time is Tfr = 2 × Tp = 51.2 μs. This means, in the worst case, a station needs to transmit for a period of 51.2 μs to detect the collision. The minimum size of the frame is 10 Mbps × 51.2 μs = 512 bits or 64bytes. This is actually the minimum size of the frame for Standard Ethernet.

DIFFERENCES BETWEEN ALOHA & CSMA/CD

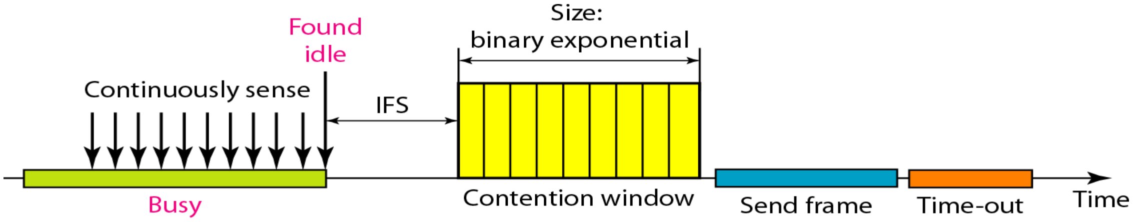
The first difference is the addition of the persistence process. We need to sense the channel before we start sending the frame by using one of the persistence processes

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 sendthe entire frame and then look for a collision. The station transmits andreceives continuously and simultaneously

The third difference is the sending of a short jamming signal that enforcesthe collision in case other stations have not yet sensed the collision.

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

We need to avoid collisions on wireless networks because they cannot bedetected. Carrier sense multiple access with collision avoidance *(CSMAlCA)* was invented for wirelesss 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 channelis 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 stationsor 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

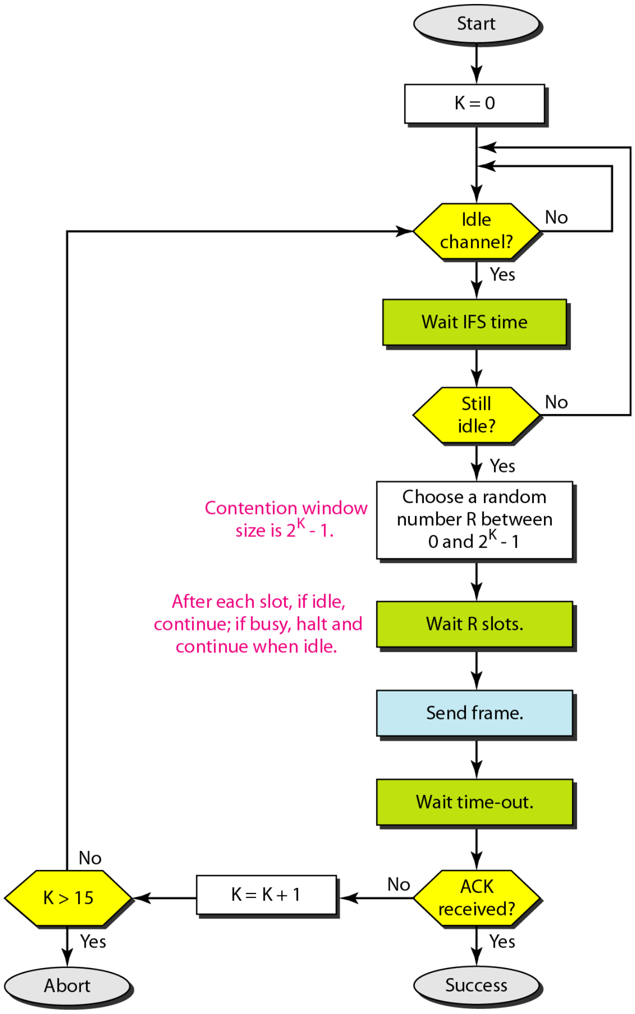
The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes according to the binary exponential back-off strategy. This means that it is set to one slot the first time and then 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.

In CSMA/CA, if the station finds the channel busy, it does not restart the timer of the contention window; it stops the timer and restarts it when the channel becomes idle.

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.


Start
K = 0
Idle channel?
No
Yes
Wait IFS time
Still idle?
No
Yes
Contention window size is 2^K - 1.
Choose a random number R between 0 and 2^K - 1
After each slot, if idle, continue; if busy, halt and continue when idle.
Wait R slots.
Send frame.
Wait time-out.
ACK received?
No
Yes
Success
K = K + 1
K > 15
No
Yes
Abort

This is the CSMA protocol with collision avoidance.

- The station ready to transmit, senses the line by using one of the persistentstrategies.
- 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 acknowledgementfrom 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

parameter, waits for the back off time and re senses the line

Controlled Access Protocols

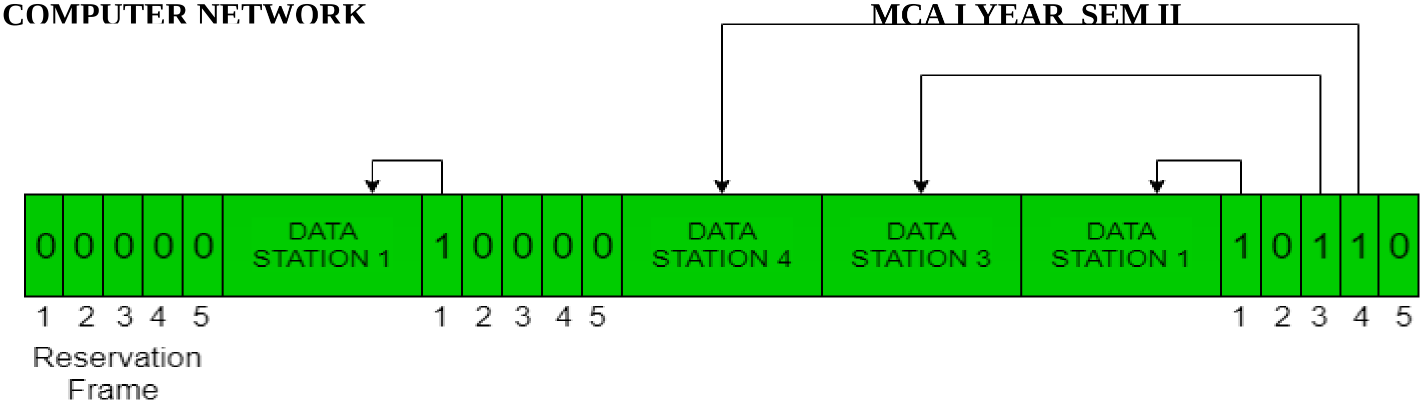
In controlled access, the stations seek information from one another to find which station has the right to send. It allows only one node to send at a time, to avoid collision of messages on shared medium. The three controlled-access methods are:

1 Reservation 2 Polling 3 Token Passing

Reservation

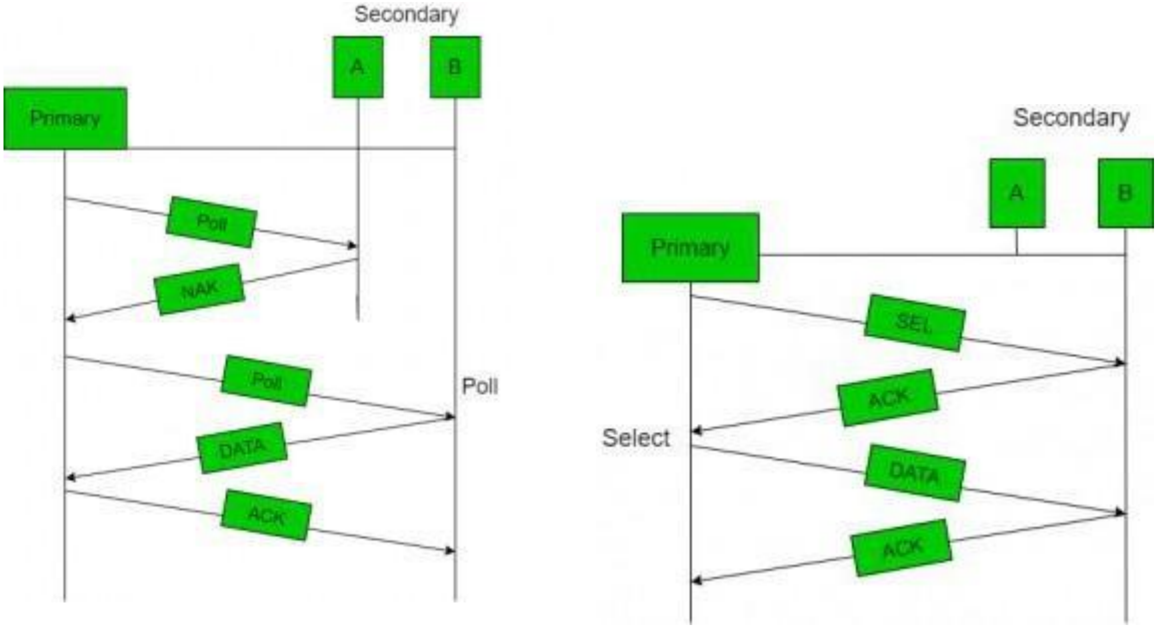
- In the reservation method, a station needs to make a reservation before sending data.
- The time line has two kinds of periods:
  1. Reservation interval of fixed time length
  2. Data transmission period of variable frames.
- If there are M stations, the reservation interval is divided into M slots, 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 stationknows 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 slotreservation 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

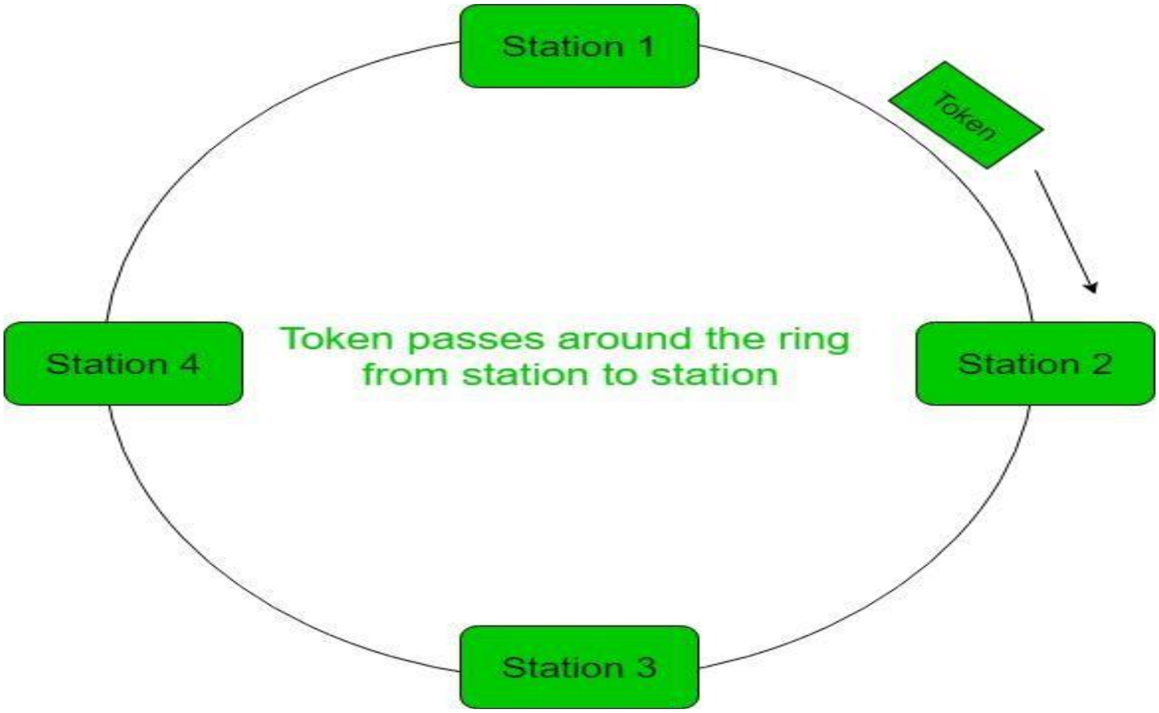
- Polling process is similar to the roll-call performed in class. Just like the teacher, a controller sends a message to each node in turn.
- In this, one acts as a primary station(controller) and the others are secondary stations. All data exchanges must be made through the controller.
- The message sent by the controller contains the address of the nodebeing selected for granting access.
- Although all nodes receive the message but the addressed one respondsto 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 eachother 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 fromone station to the next in the some predefined order.

- In Token ring, token is passed from one station to another adjacent station in the ring whereas incase of Token bus, each station uses the bus to send the token to the next station in 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 senda 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 matches 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 errorhas occurred during transmission of the message. Basic approach used for error detection is the use of redundancy bits, where additional bits are added to facilitate detection of errors. Some popular techniques for error detection are:

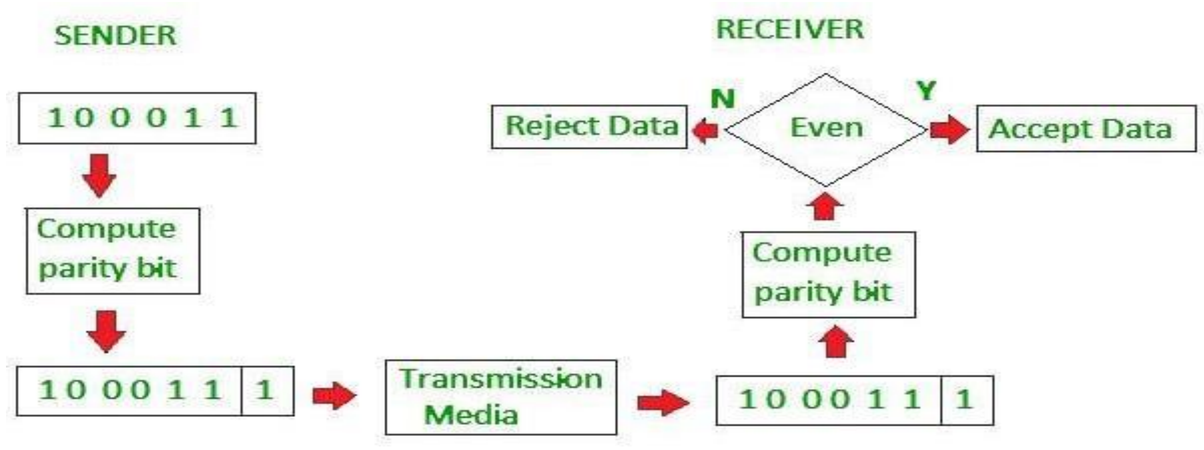
1. Simple Parity check
2. Two-dimensional Parity check
3. Checksum
4. Cyclic redundancy check

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

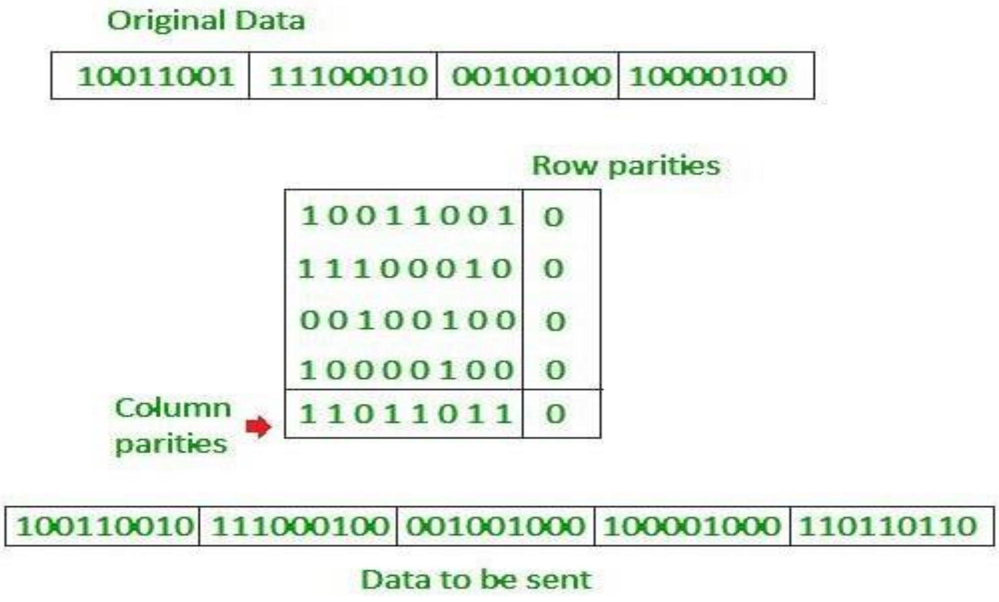
0 is added if it contains even number of 1's

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



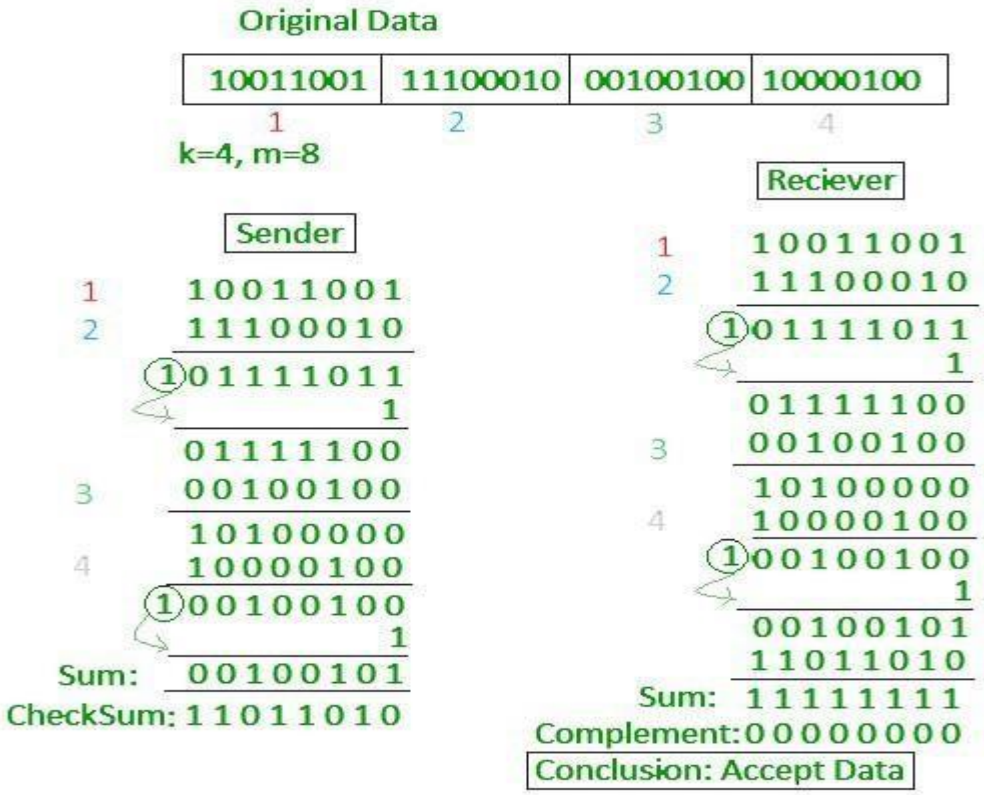
Two-dimensional Parity check

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.

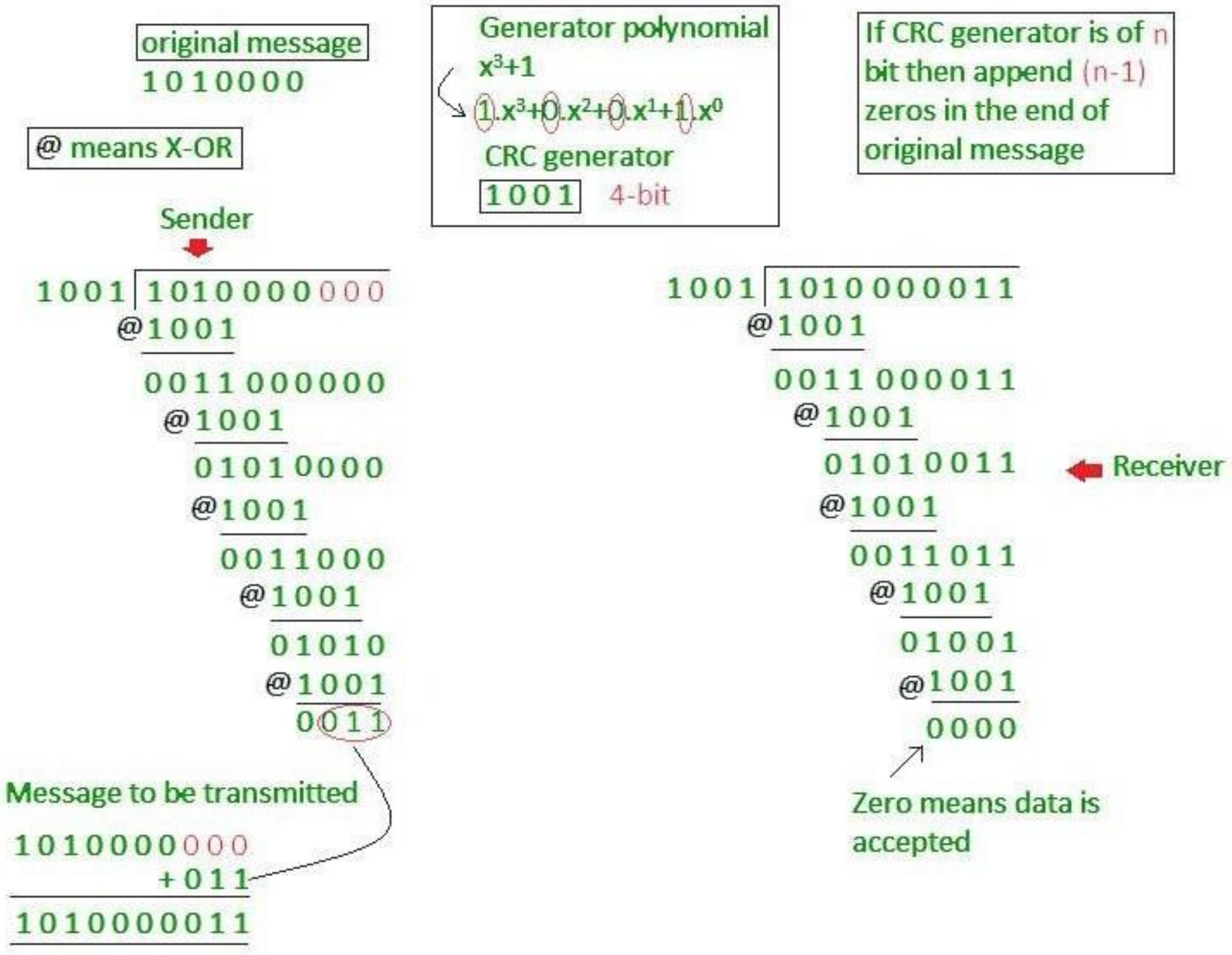


Checksum

- In checksum error detection scheme, the data is divided into k segmentseach of m bits.
- In the sender’s end the segments are added using 1’s complement arithmetic to get the sum. The sum is complemented to get the checksum.
- The checksum segment is sent along with the data segments.
- At the receiver’s end, all received segments are added using 1’s complementarithmetic to get the sum. The sum is complemented.
- If the result is zero, the received data is accepted; otherwise discarded.



Cyclic redundancy check (CRC)



- 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 becomesexactly 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 andis 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 istransmitted from the sender to the receiver.

Error Correction can be handled in two ways:

Backward error correction: Once the error is discovered, the receiver requeststhe 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.

For correcting the errors, one has to know the exact position of the error. For example, If we want to calculate a single-bit error, the error correction code willdetermine which one of seven bits is in error. To achieve this, we have to add 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:

$2^r >= d+r+1$

The value of r is calculated by using the above formula. For example, if thevalue 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 dataunit 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 thatthe 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 thevalue 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 paritybit is 0.

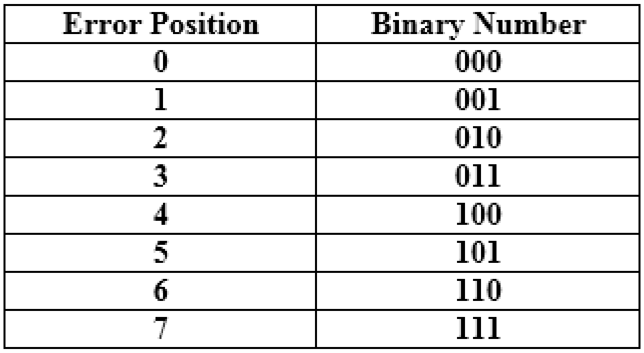
Algorithm of Hamming code:

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, ...2^{k-1}

At the receiving end, the parity bits are recalculated. The decimal value of theparity 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 : $2^r >= d+r+1$

$$2^r >= 4+r+1$$

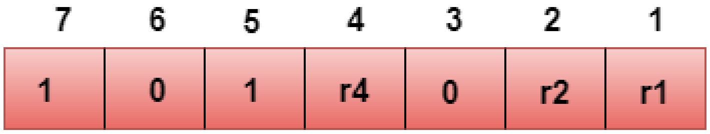
Therefore, the value of r is 3 that satisfies the above relation.Total number of bits = d+r = 4+3 = 7;

Determining the position of the redundant bits

The number of redundant bits is 3. The three bits are represented by r1, r2, r4. 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 r1 = 1, The position of r2 = 2 , The position of r4 = 4

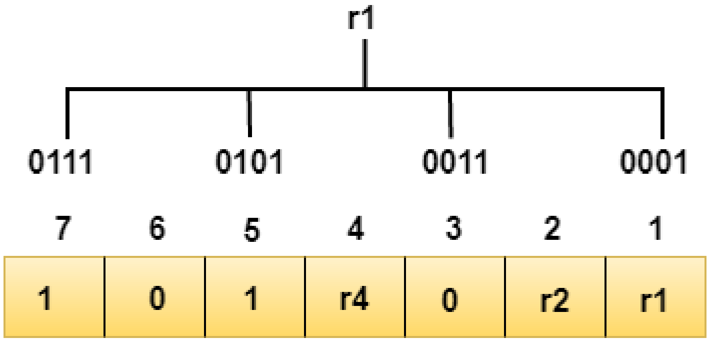
Representation of Data on the addition of parity bits:



7	6	5	4	3	2	1
1	0	1	r4	0	r2	r1

Determining the Parity bits

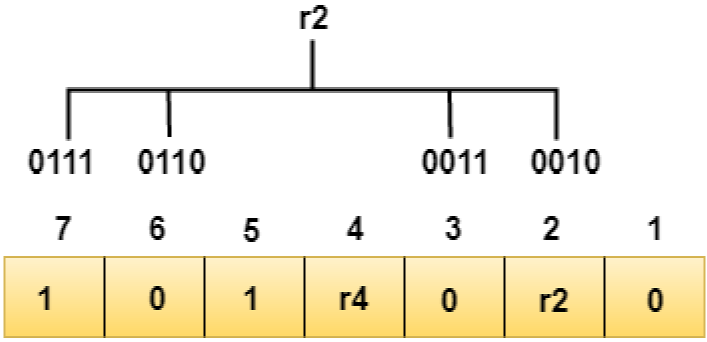
Determining the r1 bit: The r1 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the first position.



7	6	5	4	3	2	1
1	0	1	r4	0	r2	r1

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 bitpositions. The total number of 1 at these bit positions corresponding to r1 is even, therefore, the value of the r1 bit is 0.

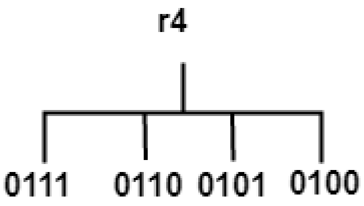
Determining r2 bit: The r2 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the second position



7	6	5	4	3	2	1
1	0	1	r4	0	r2	0

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 bit positions. The total number of 1 at these bit positions corresponding to r2 isodd, therefore, the value of the r2 bit is 1.

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.



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

We observe from the above figure that the bit positions that includes 1 in thethird 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.

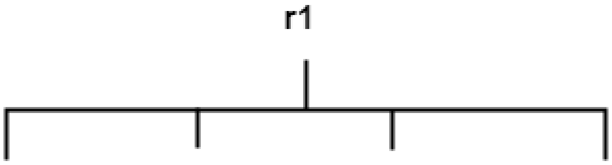
Data transferred is given below:

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

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

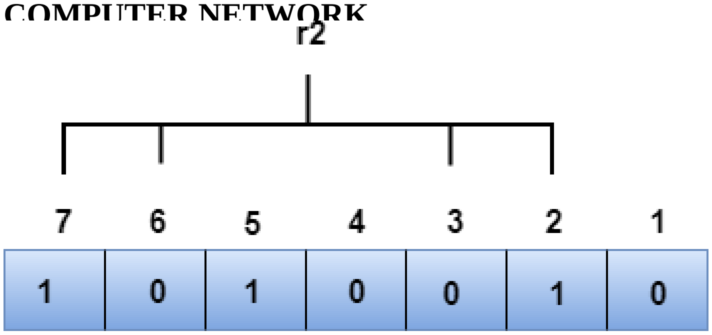


7	6	5	4	3	2	1
1	0	1	1	0	1	0

We observe from the above figure that the binary representation of 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

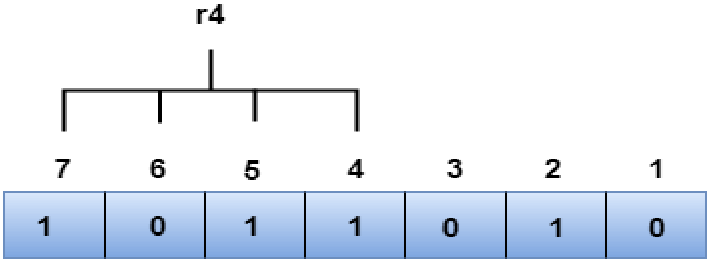
The bit positions of r2 bit are 2,3,6,7.



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.



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.

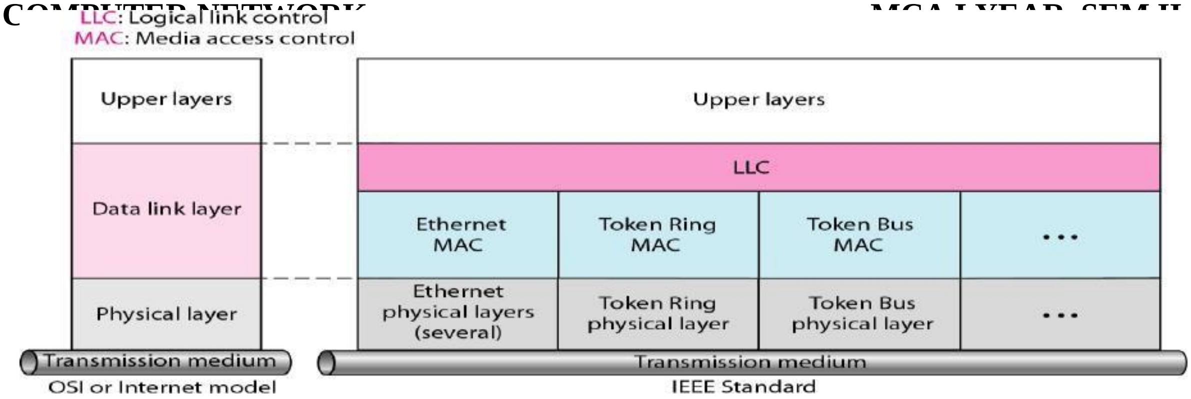
The binary representation of redundant bits, i.e., r4r2r1 is 100, and its corresponding decimal value is 4. Therefore, the error occurs in a 4th bitposition. The bit value must be changed from 1 to 0 to correct the error.

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



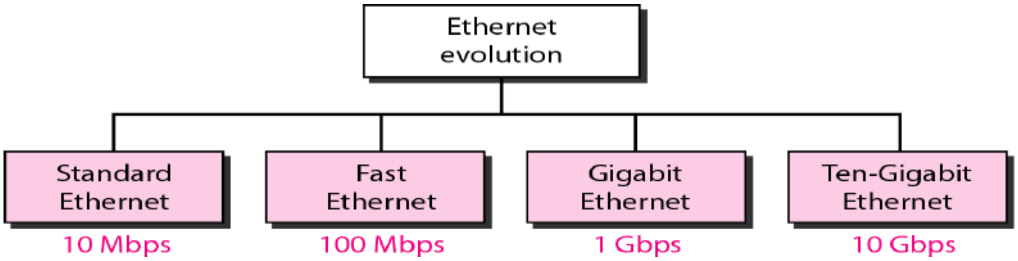
IEEE standard for LANs

STANDARD ETHERNET

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

Standard Ethernet (l0 Mbps), Fast Ethernet (100 Mbps), Gigabit Ethernet (lGbps), and Ten-Gigabit Ethernet (l0 Gbps),

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



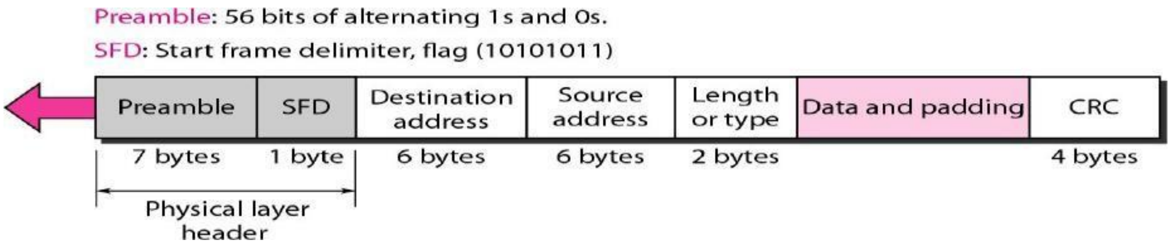
Ethernet evolution through four generations

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



802.3 MAC frame

Preamble. The first field of the 802.3 frame contains 7 bytes (56 bits) of 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 alertand 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 (l byte: 10101011) signals the beginning of the frame. The SFD warns the station or stations that this is thelast chance for synchronization. The last 2 bits is 11 and alerts the receiverthat 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.

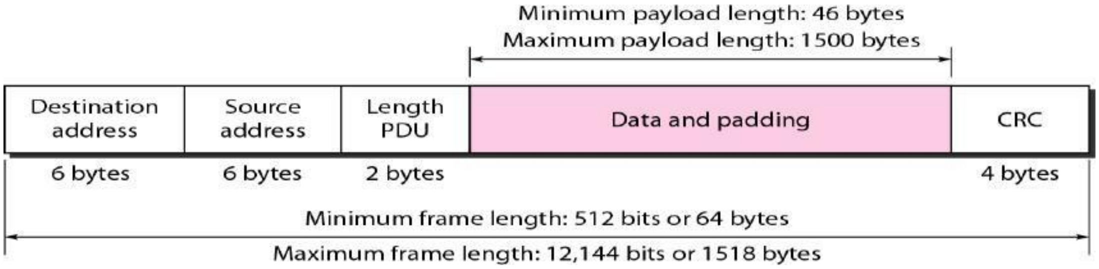
Length or type. This field is defined as a type field or length field. The original Ethernet used this field as the type field to define the upper-layer protocol using the MAC frame. The IEEE standard used it as the length field to definethe number of bytes in the data field. Both uses are common today.

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 payload length: 46 bytes
Maximum payload length: 1500 bytes
Destination address
Source address
Length PDU
Data and padding
CRC
6 bytes
6 bytes
2 bytes
4 bytes
Minimum frame length: 512 bits or 64 bytes
Maximum frame length: 12,144 bits or 1518 bytes

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 lessthan 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, the maximum length of the payload is 1500 bytes.

The maximum length restriction has two historical reasons.

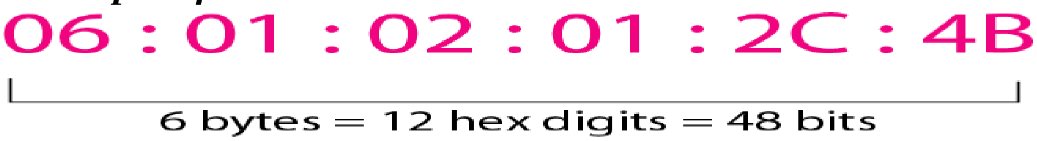
First, memory was very expensive when Ethernet was designed: a maximum length restriction helped to reduce the size of the buffer.

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

Addressing

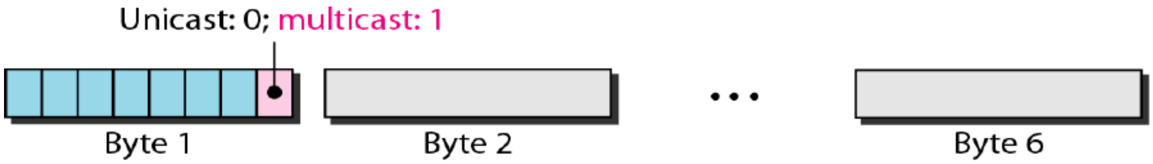
The Ethernet address is 6 bytes (48 bits), normally written in hexadecimalnotation, with a colon between the bytes.

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, theaddress 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-eight1s.

Access Method: CSMA/CD

Standard Ethernet uses I-persistent CSMA/CDSlot 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 to send 512 bits. This means that the actual slot time depends on the data rate; for traditional 10-Mbps Ethernet it is 51.2 micro sec.

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 x 10^8 m/s (two-thirds of the rate for propagation in air).

For traditional Ethernet, we calculate

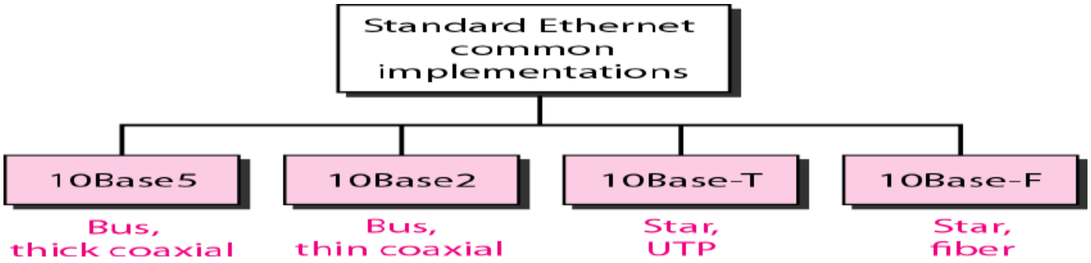
MaxLength =PropagationSpeedx (SlotTime/2)

MaxLength= (2 x 10^8) X(51.2 X10^{-6})/2= 5120m

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. MaxLength=2500 m

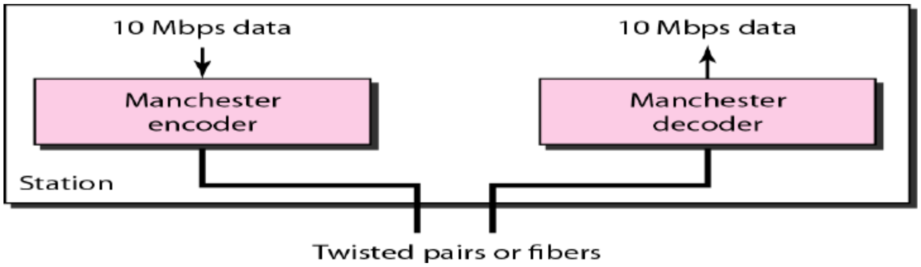
Physical Layer

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

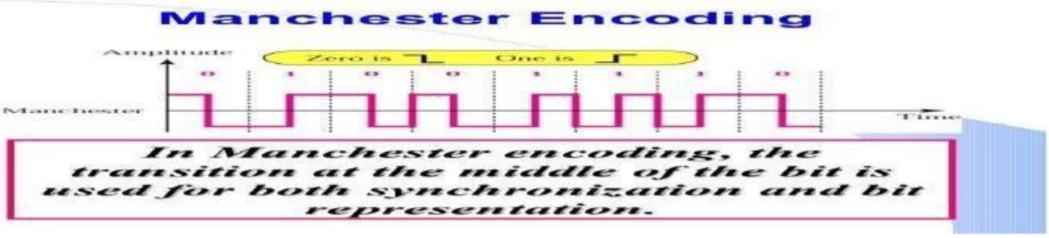


Encoding and Decoding

All standard implementations use digital signaling (baseband) at 10 Mbps. Atthe 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

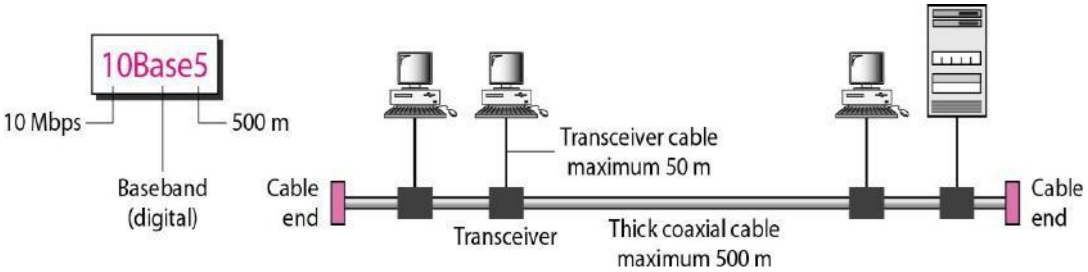


In Manchester encoding, the transition at the middle of the bit is used for synchronization


Manchester Encoding
Amplitude
Zero is
One is
0
1
0
0
1
1
1
0
Manchester
Time
In Manchester encoding, the transition at the middle of the bit is used for both synchronization and bit representation.

lOBase5: Thick Ethernet

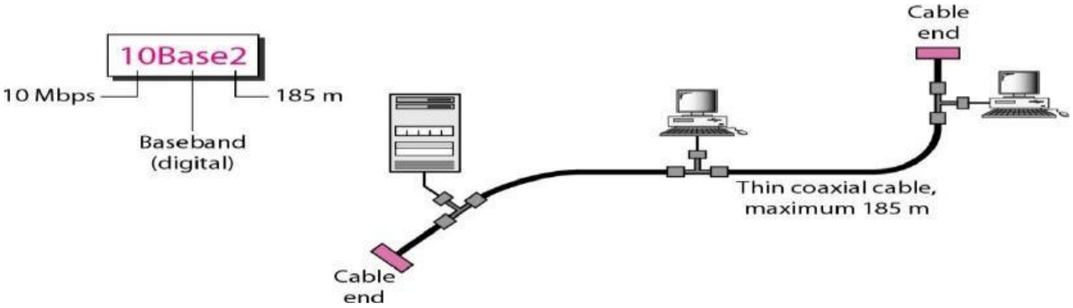
The first implementation is called **10Base5, thick Ethernet, or Thicknet.**lOBase5 was the first Ethernet specification to use a bus topology with an external **transceiver** (transmitter/receiver) connected via a tap to a thick coaxial cable. Figure shows a schematic diagram of a lOBase5 implementation



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.

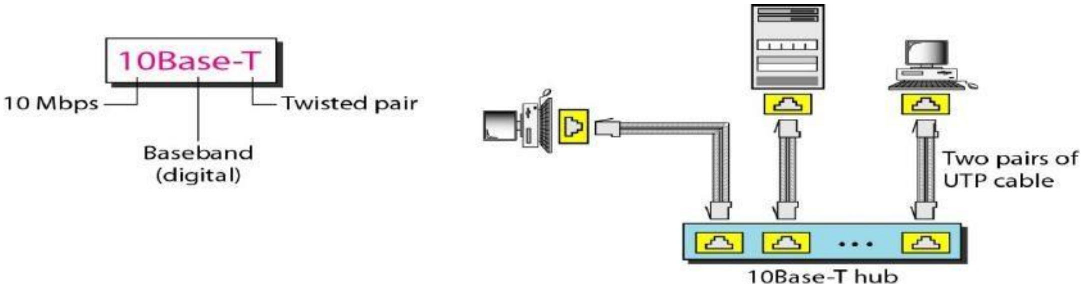


10Base2 implementation

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.

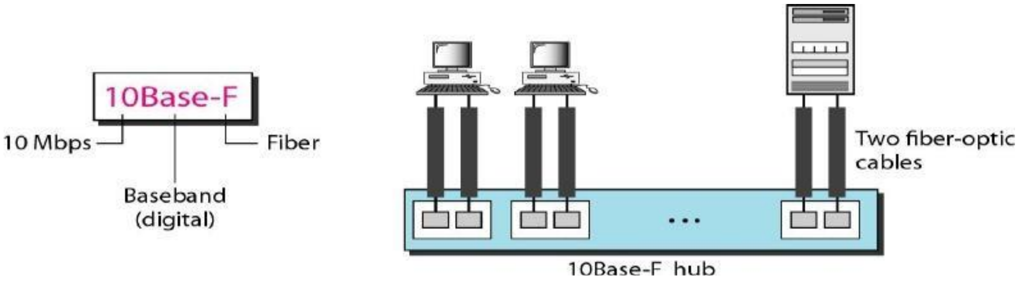
1OBase-T: Twisted-Pair Ethernet

The third implementation is called 10Base-T or twisted-pair Ethernet. It uses a physical star topology. The stations are connected to a hub via two pairs of twisted cable, as shown in Figure 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 mostcommon is called 10Base-F.10Base-F uses a star topology to connect stationsto a hub. The stations are connected to the hub using two fiber-optic cables, as shown in Figure



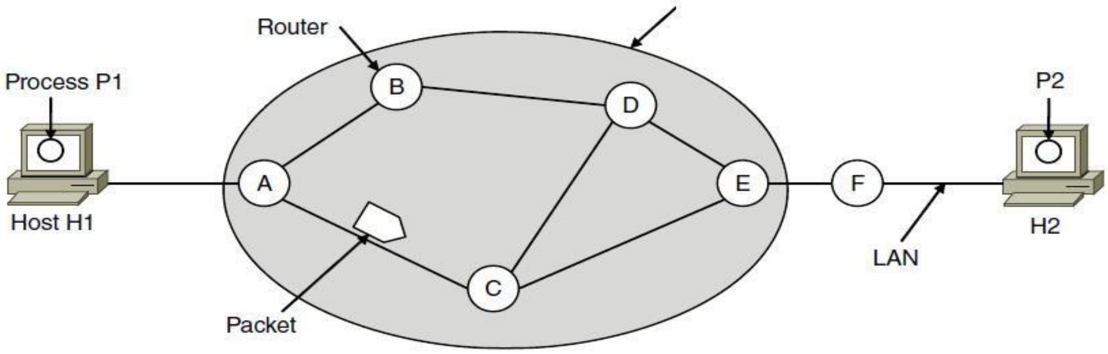
10Base-F implementation

UNIT-III

Network Layer Design Issues

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.

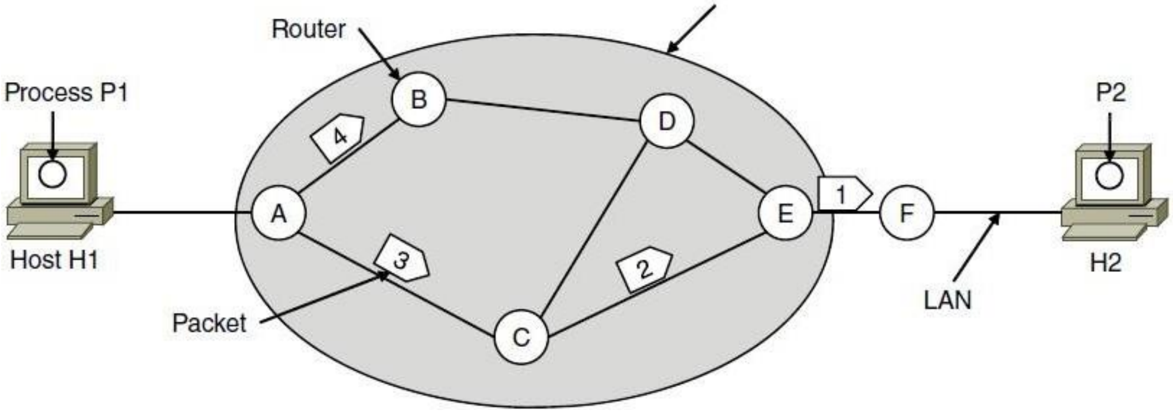
2 Services provided to transport layer

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:

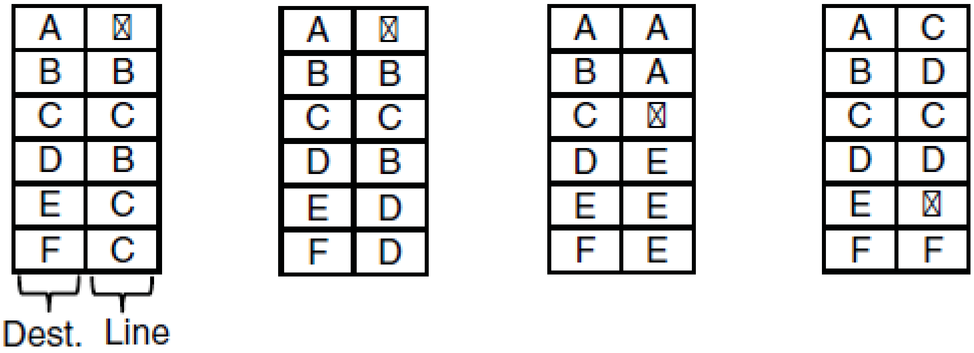
1. Services independent of router technology.
2. Transport layer shielded from number, type, topology of routers.
3. Network addresses available to transport layer use uniform numbering plan
   - even across LANs and WANs

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 are frequently called **datagrams** (in analogy with telegrams) and the network is called a **datagram network**.



A’s table (initially) A’s table (later) C’s Table E’s Table



A’s table (initially)

Dest.	Line
A	☒
B	B
C	C
D	B
E	C
F	C

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.

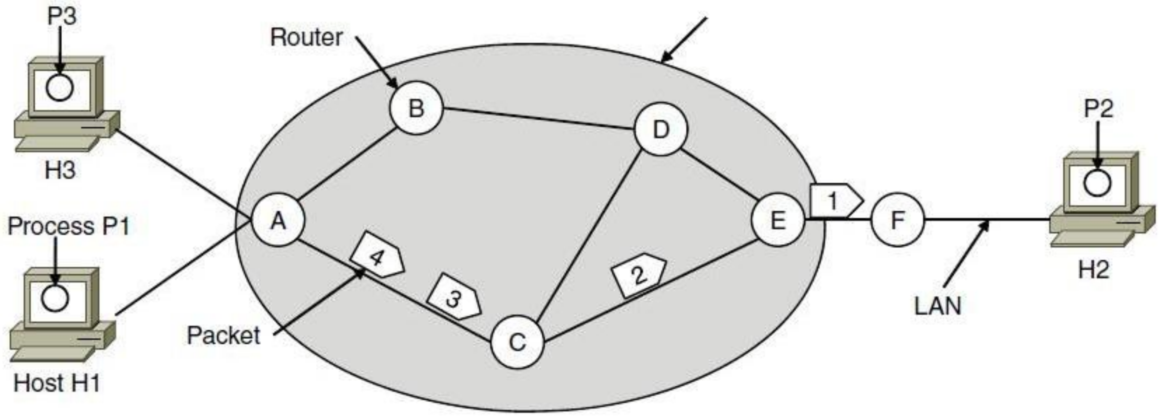
A’s initial routing table is shown in the figure under the label ‘‘initially.’’

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**.

4 Implementation of connection-oriented service



A's table

In		Out	
H1	1	C	1
H3	1	C	2

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.

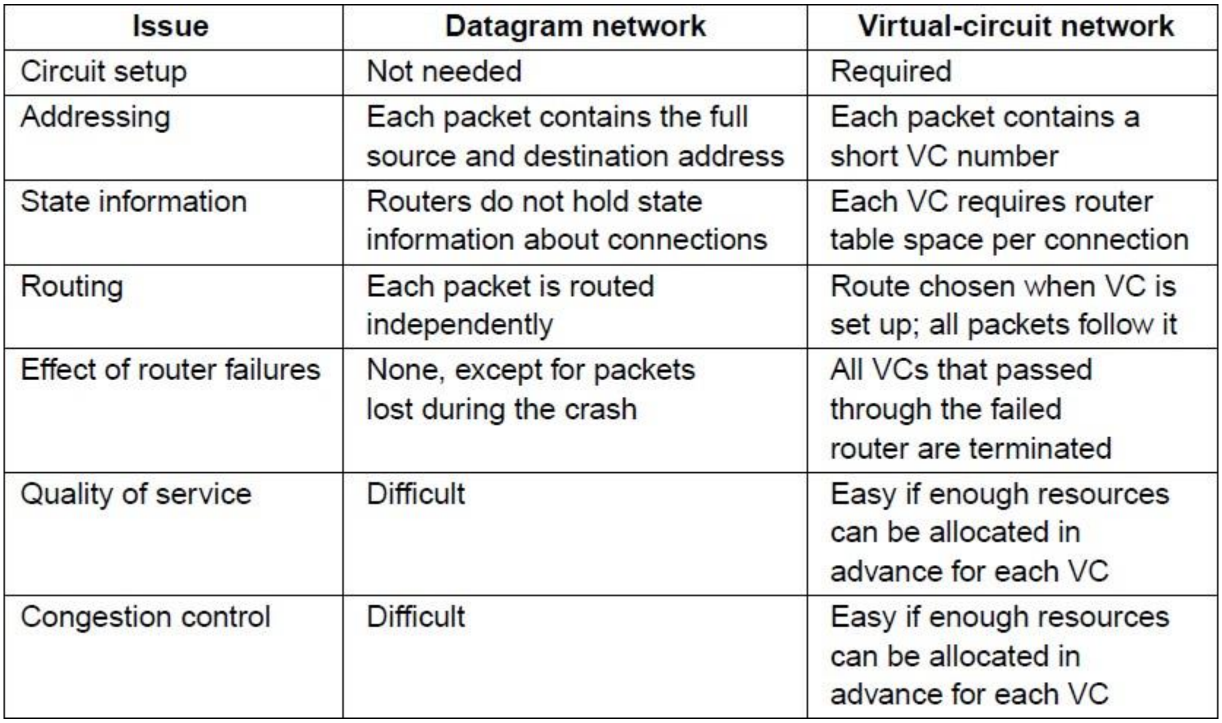
As an example, consider the situation shown in Figure. Here, host *H1* has established connection 1 with host *H2*. This connection is remembered as the first entry in each of the routing tables. The first line of *A*'s table says that if a packet bearing connection identifier 1 comes in from *H1*, it 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.

This leads to the second row in the tables. Note that we have a conflict here because although *A* can easily distinguish connection 1 packets from *H1* from connection 1 packets from *H3*, *C* 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:

a) 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.

b) 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**

Routing algorithms can be grouped into two major classes:

1) nonadaptive (Static Routing)
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

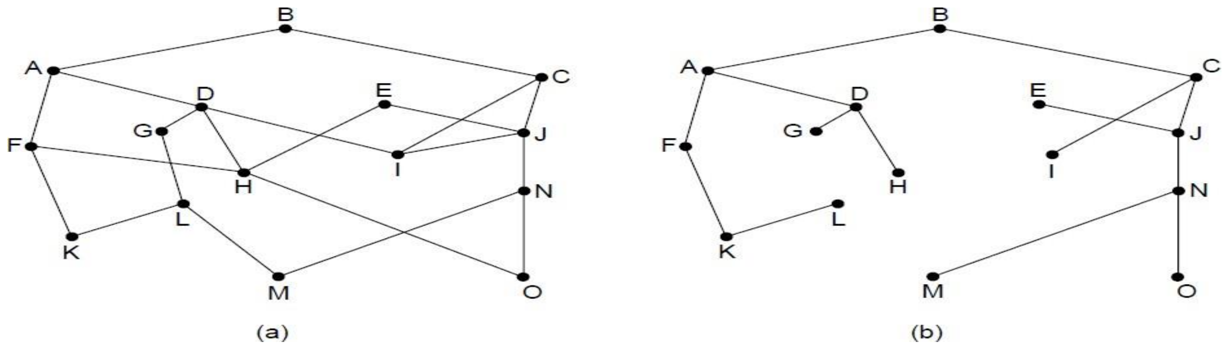
- Optimality principle
- Shortest path algorithm
- Flooding
- Distance vector routing
- Link state routing
- Hierarchical Routing

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



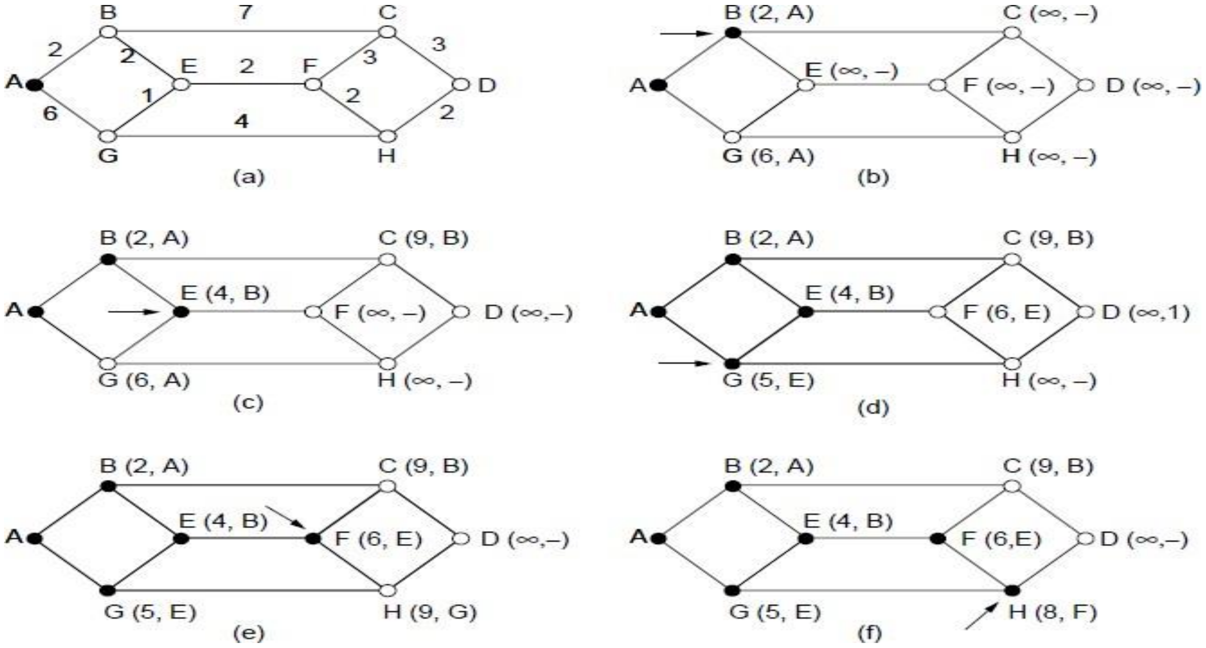
(a) A network. (b) A sink tree for router *B.*

Shortest Path Routing (Dijkstra's)

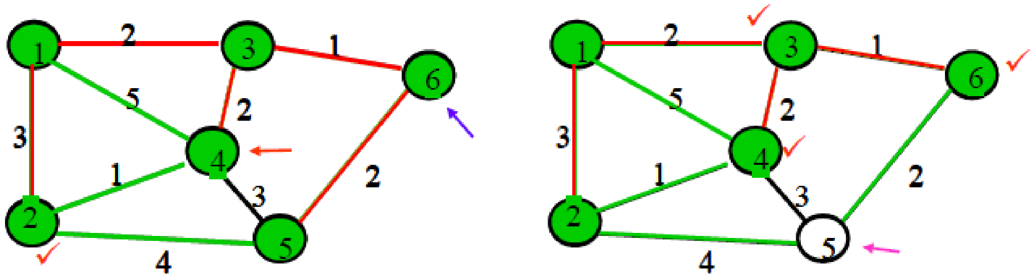
The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line or link.

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



Execution of Dijkstra's algorithm



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

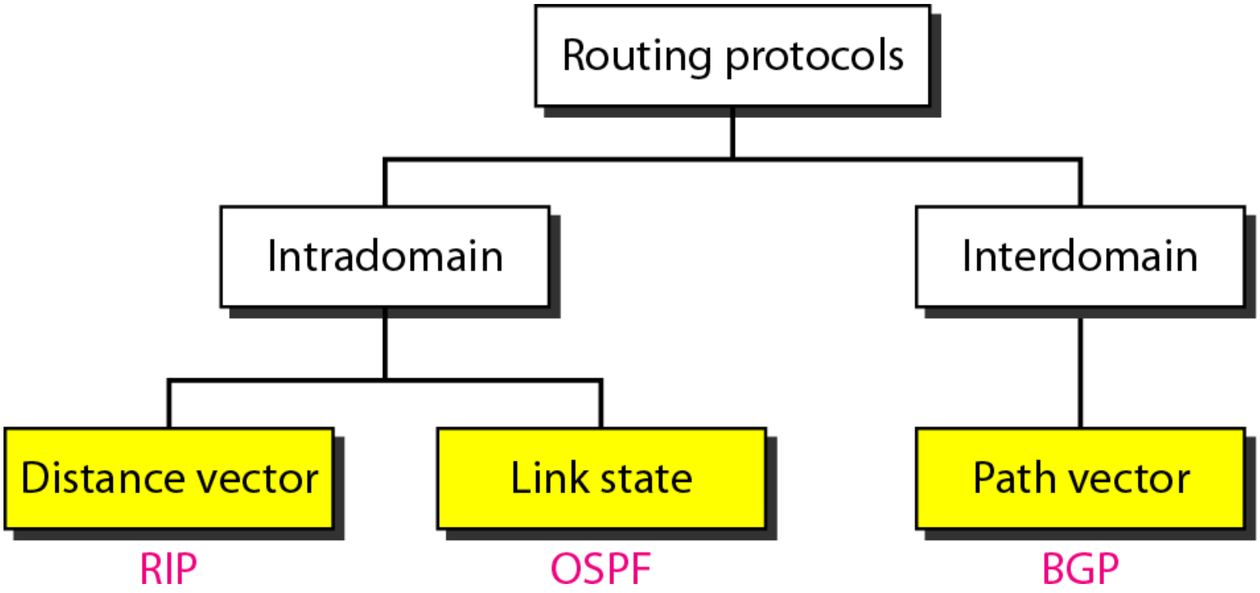
- Another static algorithm is flooding, in which every incoming packet is sent out on every outgoing line except the one it arrived on.
- Flooding obviously generates vast numbers of duplicate packets, in fact, an infinite number unless some measures are taken to damp the process.
- One such measure 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 between autonomous systems is referred to as inter domain routing. (PATH VECTOR) Each autonomous system can choose one or more intra domain routing protocols to handle 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.

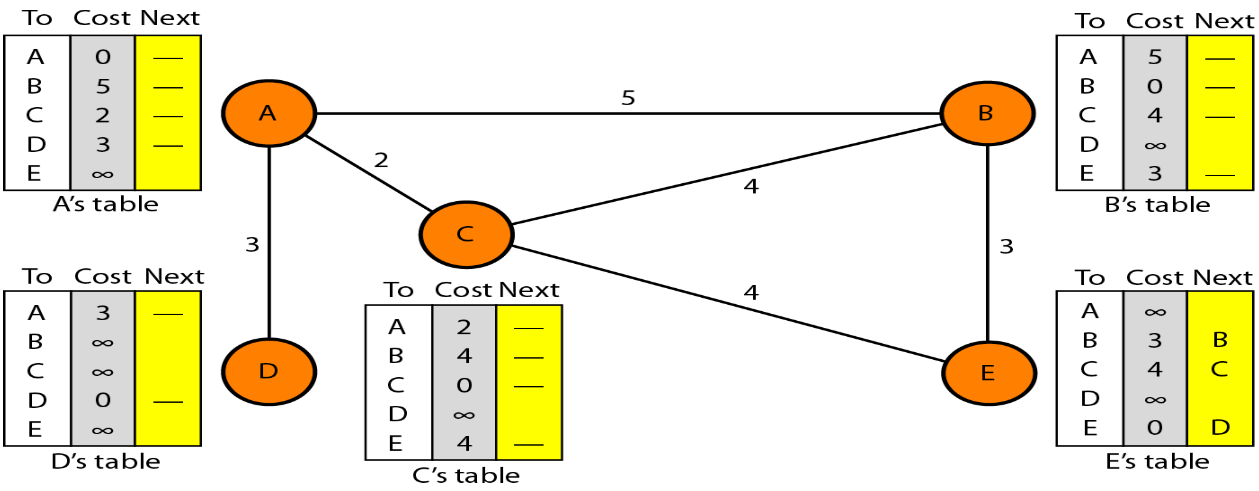
Mainly 3 things in this

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).

Initialization of tables in distance vector routing



To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	∞	

A's table

To	Cost	Next
A	5	—
B	0	—
C	4	—
D	∞	
E	3	—

B's table

To	Cost	Next
A	3	—
B	∞	
C	∞	
D	0	—
E	∞	

D's table

To	Cost	Next
A	2	—
B	4	—
C	0	—
D	∞	
E	4	—

C's table

To	Cost	Next
A	∞	
B	3	B
C	4	C
D	∞	
E	0	D

E's table

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.

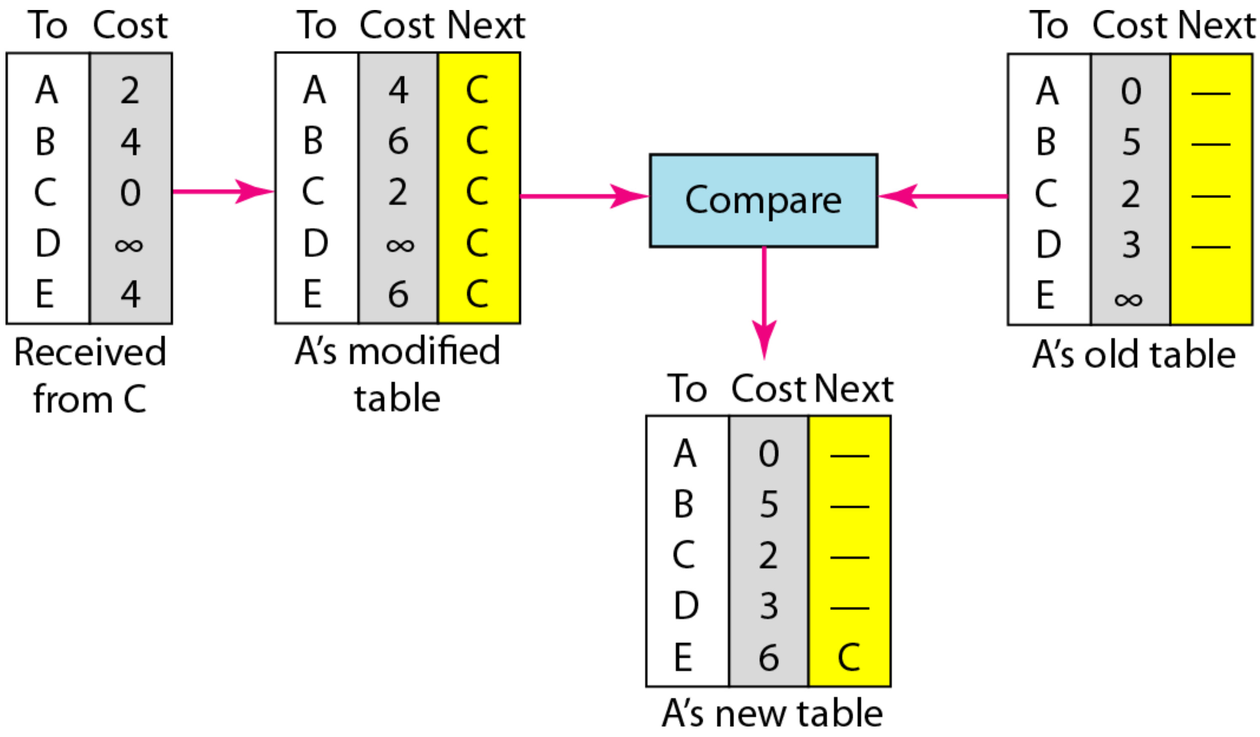
3. The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.

   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.

For example, suppose node C has previously advertised a route to node X with distance 3. Suppose that now there is no path between C and X; node C now advertises this route with a 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



Received from C

To	Cost
A	2
B	4
C	0
D	∞
E	4

A's modified table

To	Cost	Next
A	4	C
B	6	C
C	2	C
D	∞	C
E	6	C

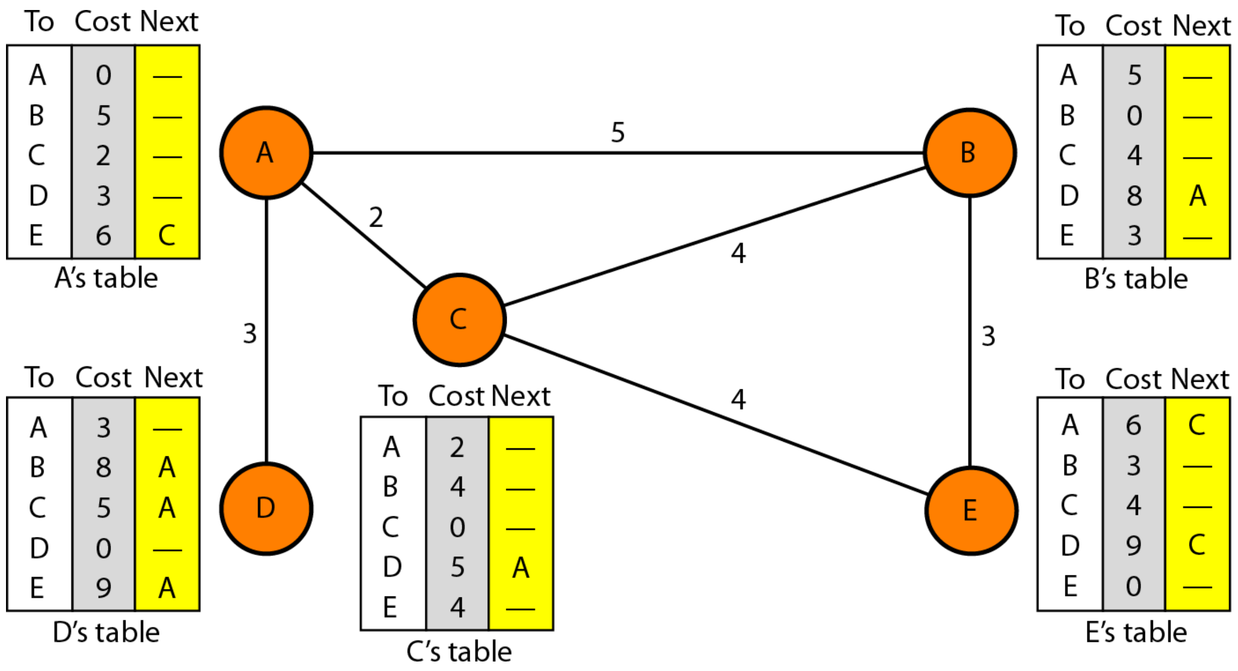
A's old table

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	∞	

A's new table

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

Final Diagram



A's table

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

B's table

To	Cost	Next
A	5	—
B	0	—
C	4	—
D	8	A
E	3	—

D's table

To	Cost	Next
A	3	—
B	8	A
C	5	A
D	0	—
E	9	A

C's table

To	Cost	Next
A	2	—
B	4	—
C	0	—
D	5	A
E	4	—

E's table

To	Cost	Next
A	6	C
B	3	—
C	4	—
D	9	C
E	0	—

When to Share

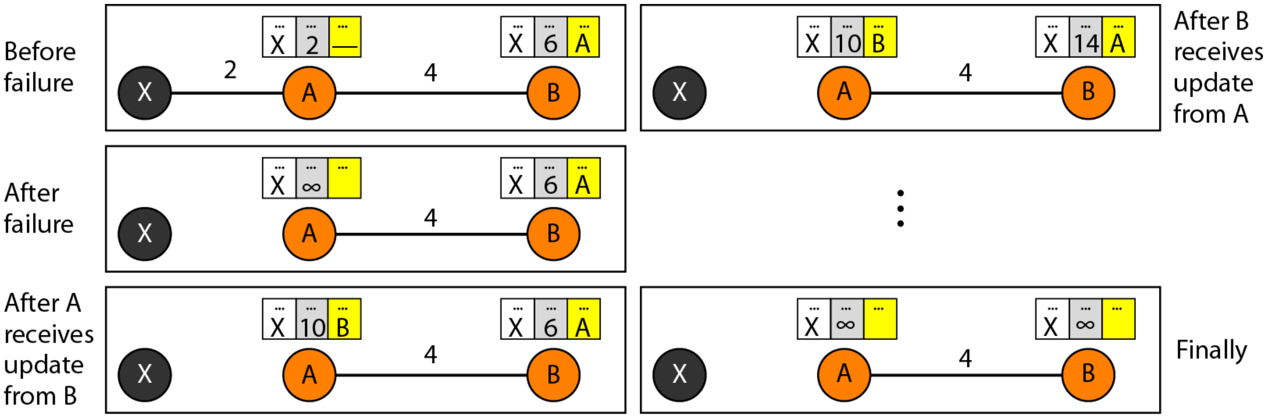
The question now is, When does a node send its partial routing table (only two columns) to all its immediate neighbors? The table is sent both periodically and when there is a change in the table.

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.

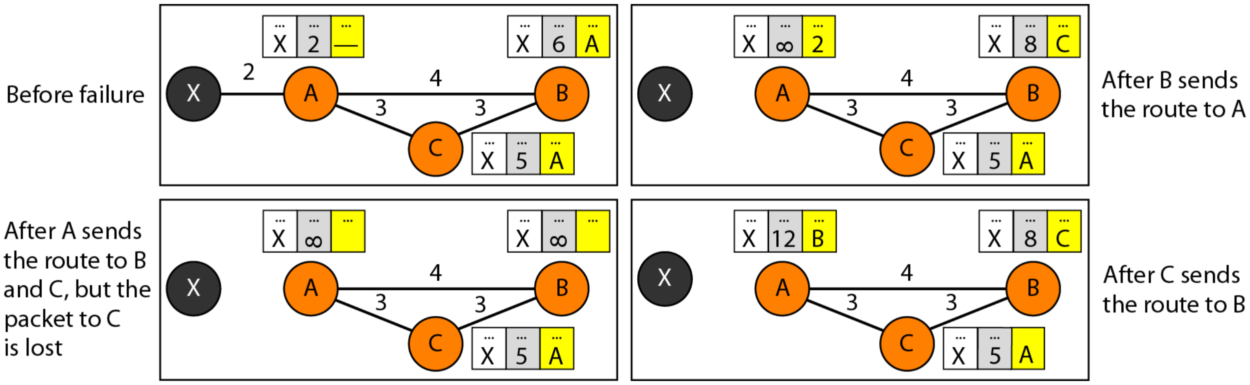
Triggered Update A node sends its two-column routing table to its neighbors anytime there is a change in its routing table. This is called a triggered update. The change can result from the following.

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


Before failure
X 2 —
X 6 A
X
2
A
4
3
3
B
C
X 5 A
X ∞ 2
X 8 C
X
A
4
3
3
B
C
X 5 A
After B sends the route to A
After A sends the route to B and C, but the packet to C is lost
X ∞
X ∞
X
A
4
3
3
B
C
X 5 A
X 12 B
X 8 C
X
A
4
3
3
B
C
X 5 A
After C sends the route to B

SOLUTIONS FOR INSTABILITY

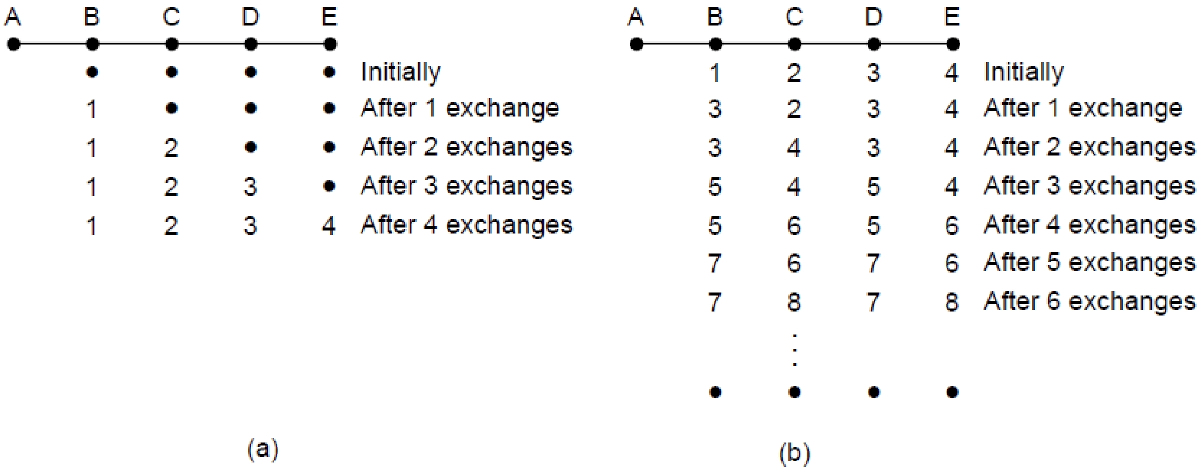
1. **Defining Infinity:** redefine infinity to a smaller number, such as 100. For our previous scenario, the system will be stable in less than 20 updates. As a matter of fact, most implementations of the distance vector protocol define the distance between each node to

be 1 and define 16 as infinity. However, this means that the distance vector routing cannot be used in large systems. The size of the network, in each direction, cannot exceed 15 hops.

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

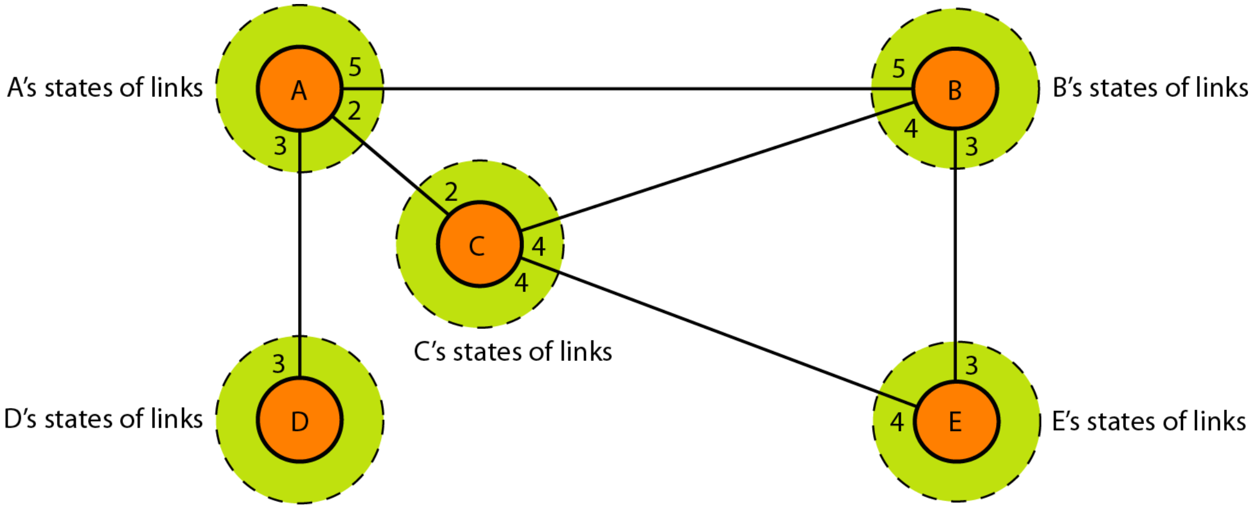
(a)

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)

Link State Routing

Link state routing is based on the assumption that, although the global knowledge about the 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

2. A node that receives an LSP compares it with the copy it may already have. If the newly arrived LSP is older than the one it has (found by checking the sequence number), it discards the LSP. If it is newer, the node does the following:

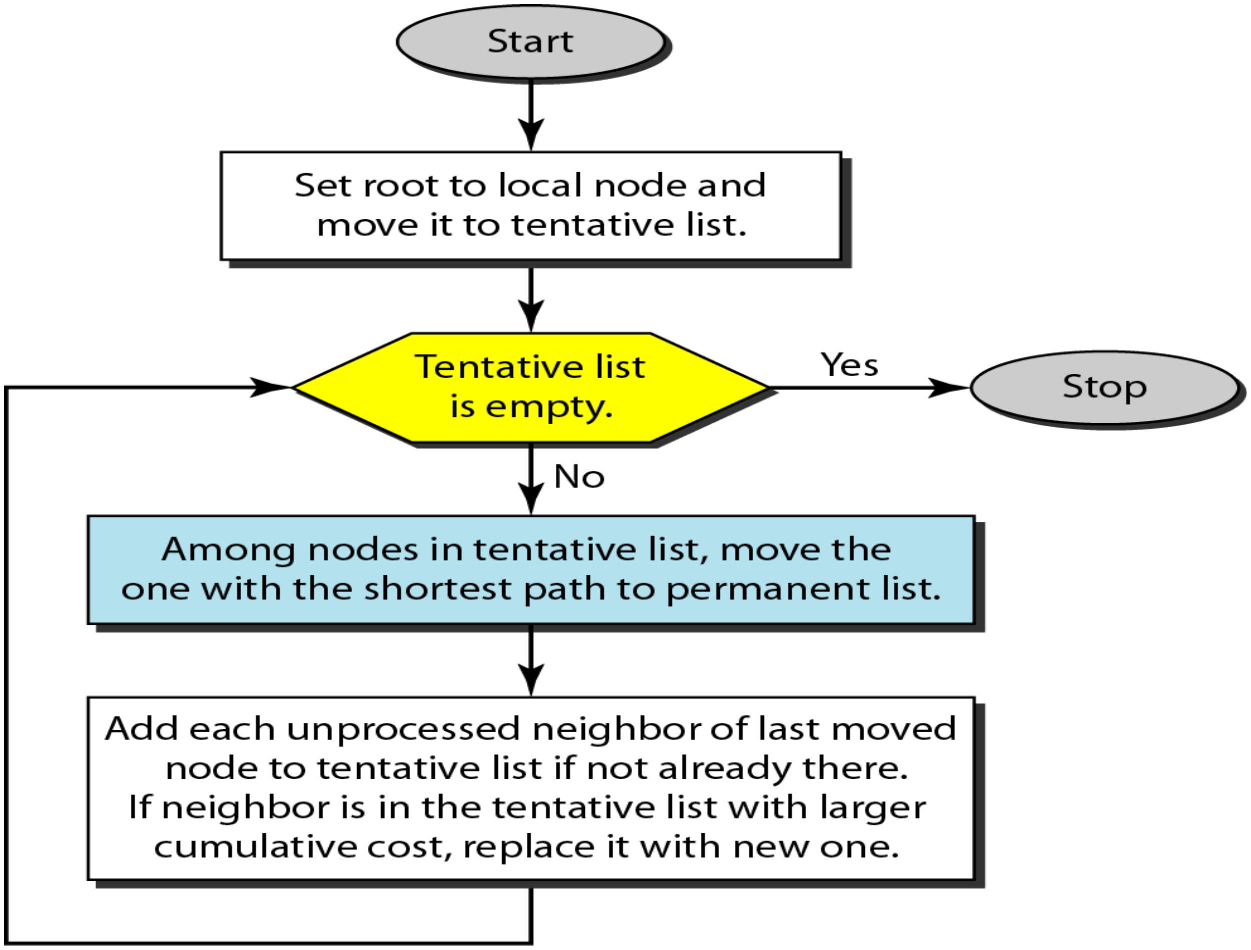
a. It discards the old LSP and keeps the new one.

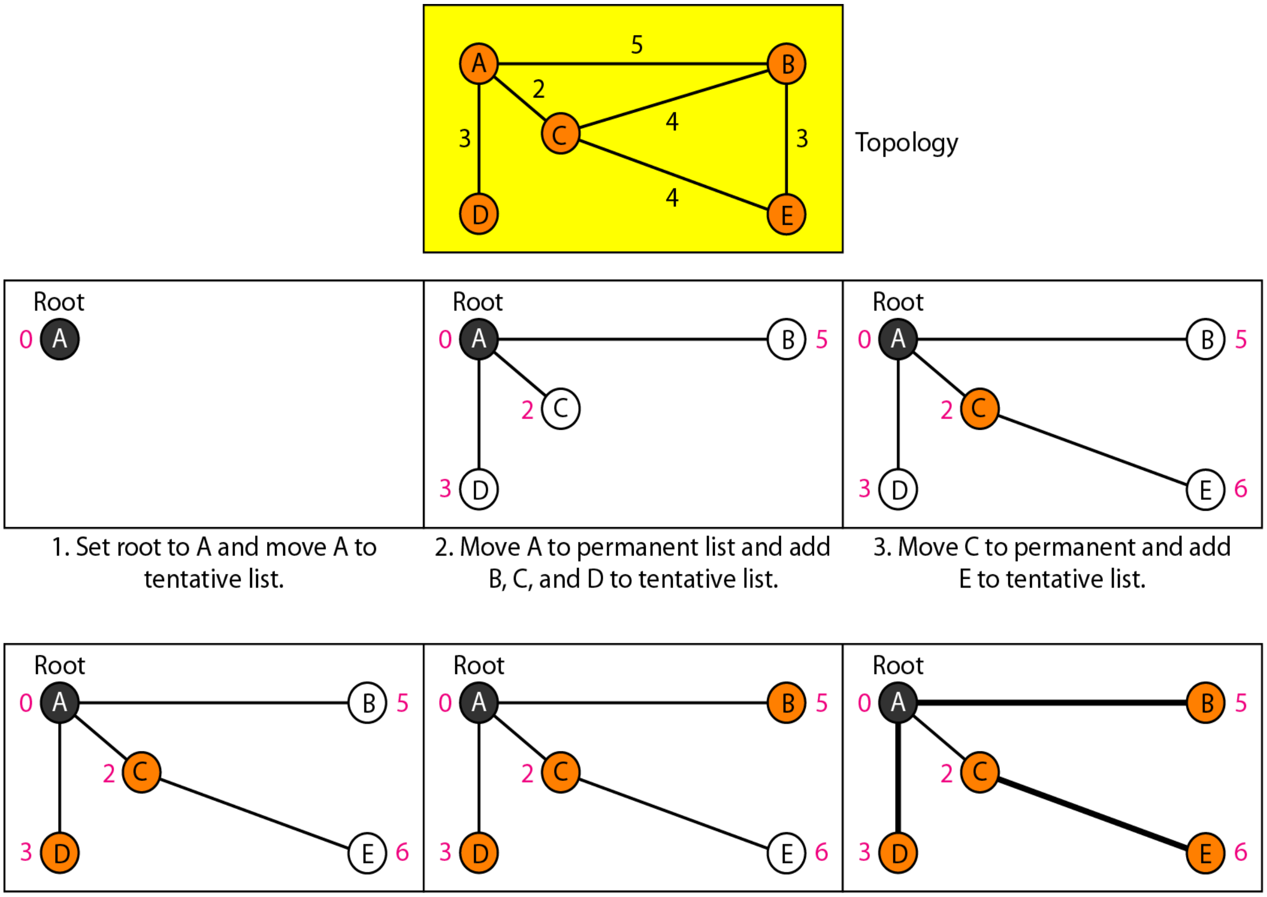
b. It sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the domain (where a node has only one interface).

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.


Start
Set root to local node and move it to tentative list.
Tentative list is empty.
Yes
Stop
No
Among nodes in tentative list, move the one with the shortest path to permanent list.
Add each unprocessed neighbor of last moved node to tentative list if not already there.
If neighbor is in the tentative list with larger cumulative cost, replace it with new one.



1. Set root to A and move A to tentative list.

2. Move A to permanent list and add B, C, and D to tentative list.

3. Move C to permanent and add E to tentative list.

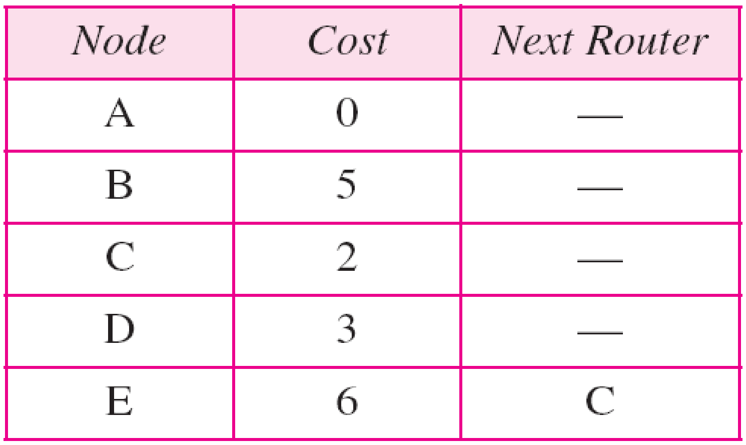
4. Move D to permanent list.

5. Move B to permanent list.

6. Move E to permanent list (tentative list is empty).

IV. Calculation of a routing table

routing table for node A



Node	*Cost*	*Next Router*
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

Path Vector Routing

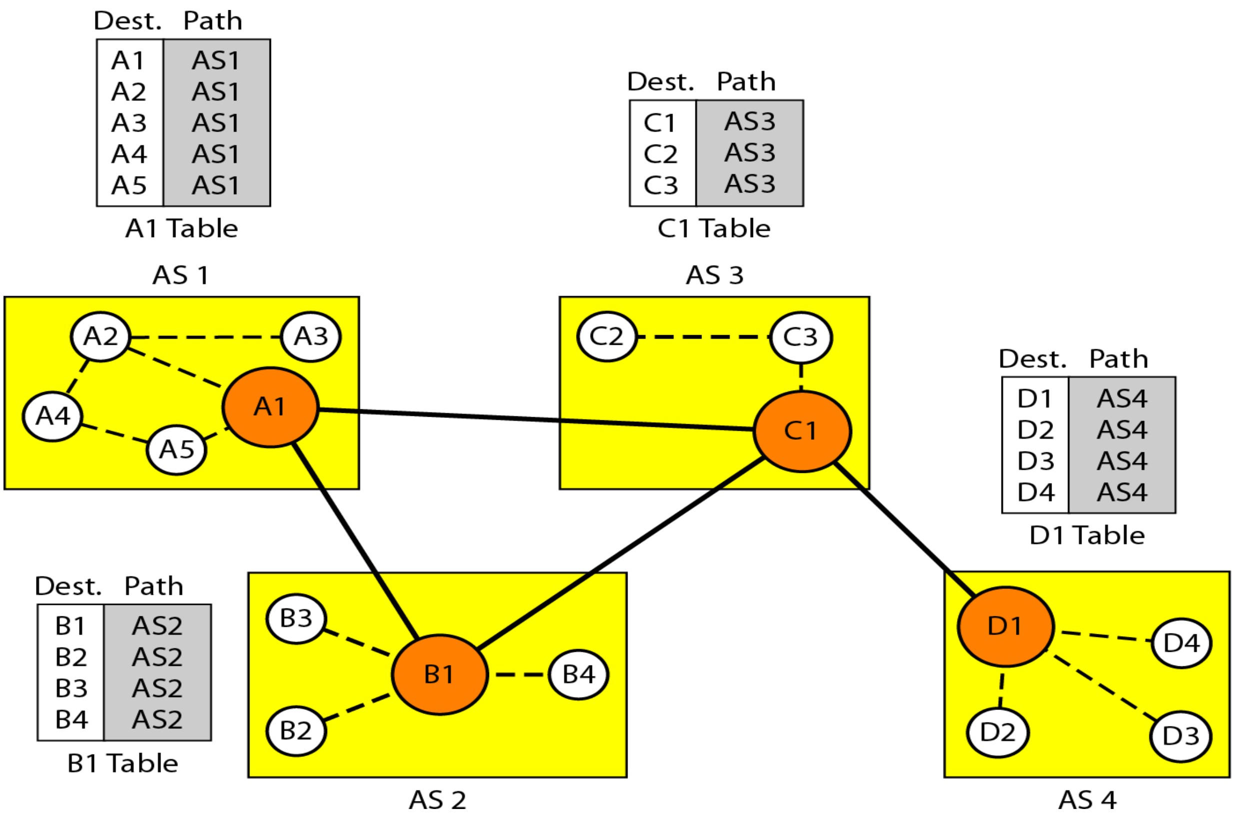
Distance vector and link state routing are both intra domain routing protocols. They can be used inside an autonomous system, but not between autonomous systems. These two protocols are not suitable for inter domain routing mostly because of scalability. Both of these routing protocols become intractable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

huge amount of resources to calculate routing tables. It also creates heavy traffic because of flooding. There is a need for a third routing protocol which we call path vector routing.

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



A1 Table

Dest.	Path
A1	AS1
A2	AS1
A3	AS1
A4	AS1
A5	AS1

C1 Table

Dest.	Path
C1	AS3
C2	AS3
C3	AS3

D1 Table

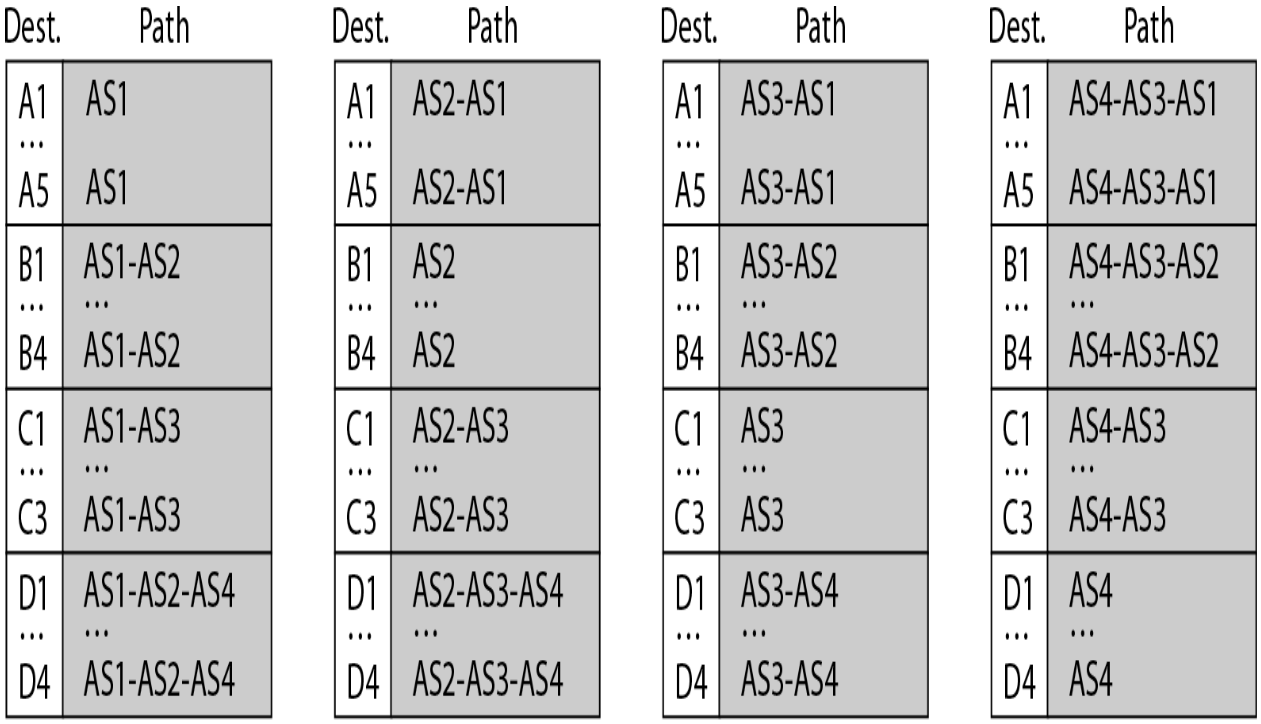
Dest.	Path
D1	AS4
D2	AS4
D3	AS4
D4	AS4

B1 Table

Dest.	Path
B1	AS2
B2	AS2
B3	AS2
B4	AS2

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 and C1. Node C1 shares its table with nodes D1, B1, and A1. Node B1 shares its table with C1 and A1. Node D1 shares its table with C1.



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

A1 Table

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

B1 Table

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

C1 Table

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

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

a) **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.

b) **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.

c) **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 ASI can be AS4-AS3-AS2-AS1, or it can be AS4-AS3-ASI. 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

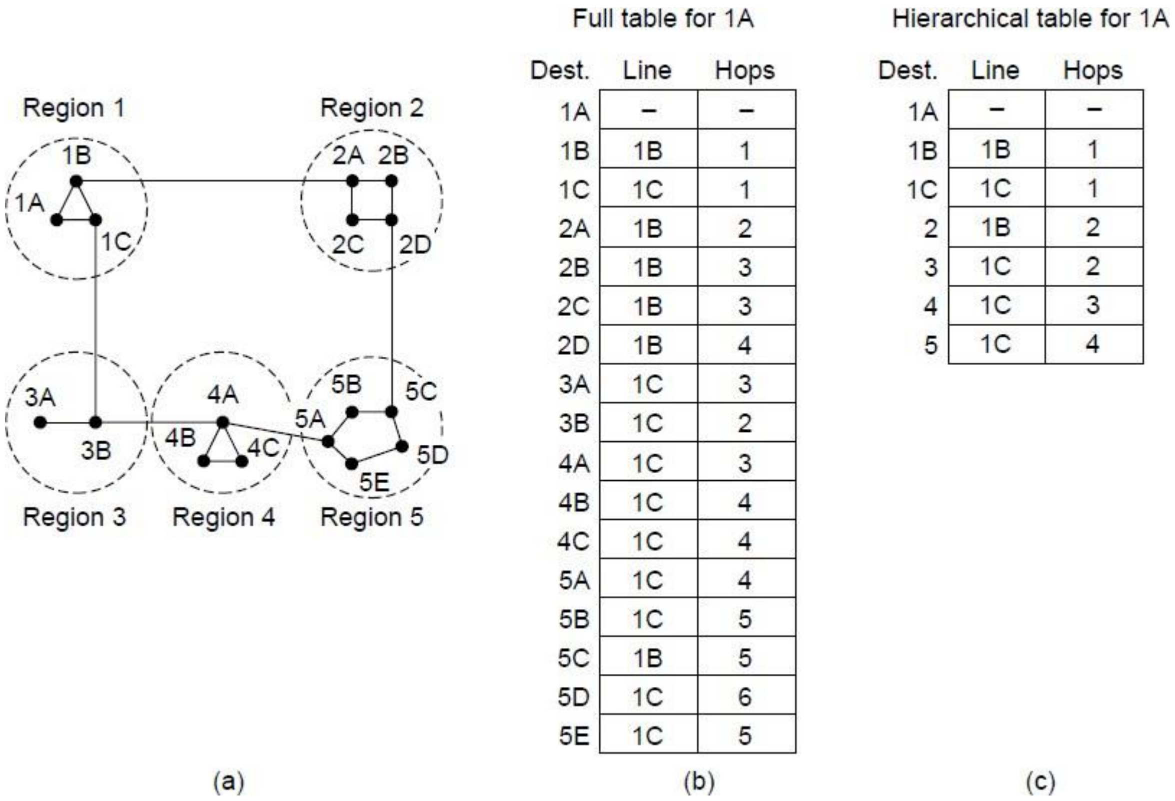
Hierarchical Routing

As networks grow in size, the router routing tables grow proportionally. Not only is router 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



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

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

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.

If a three-level hierarchy is chosen, with 8 clusters each containing 9 regions of 10 routers, each router needs 10 entries for local routers, 8 entries for routing to other regions within its own cluster, and 7 entries for distant clusters, for a total of 25 entries

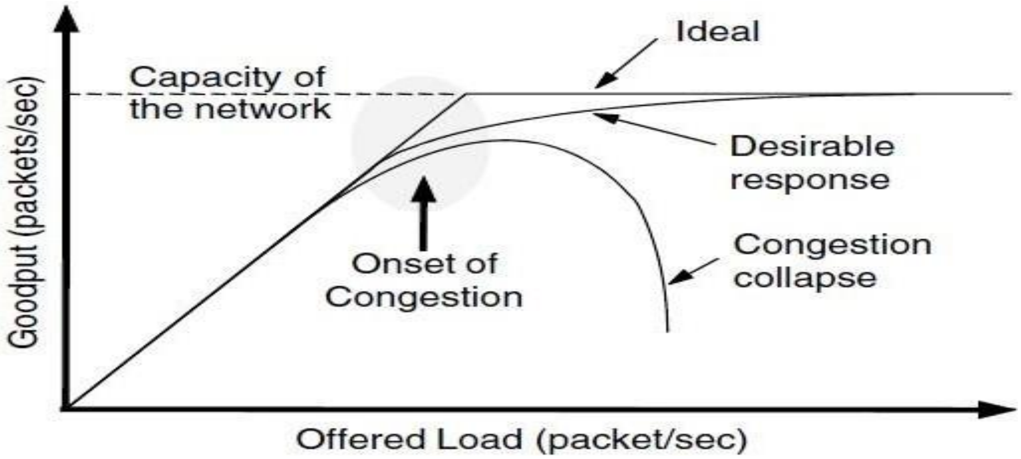
Kamoun and Kleinrock (1979) discovered that the optimal number of levels for an *N router network is ln N, requiring a total of e ln N entries per router*

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**

<u>difference between congestion control and flow control</u>.

Congestion control has to do with making sure the network is able to carry the offered traffic. 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.

Choke Packets

1. A more direct way of telling the source to slow down.
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.

   <u>Hop-by-Hop Choke Packets</u>

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)

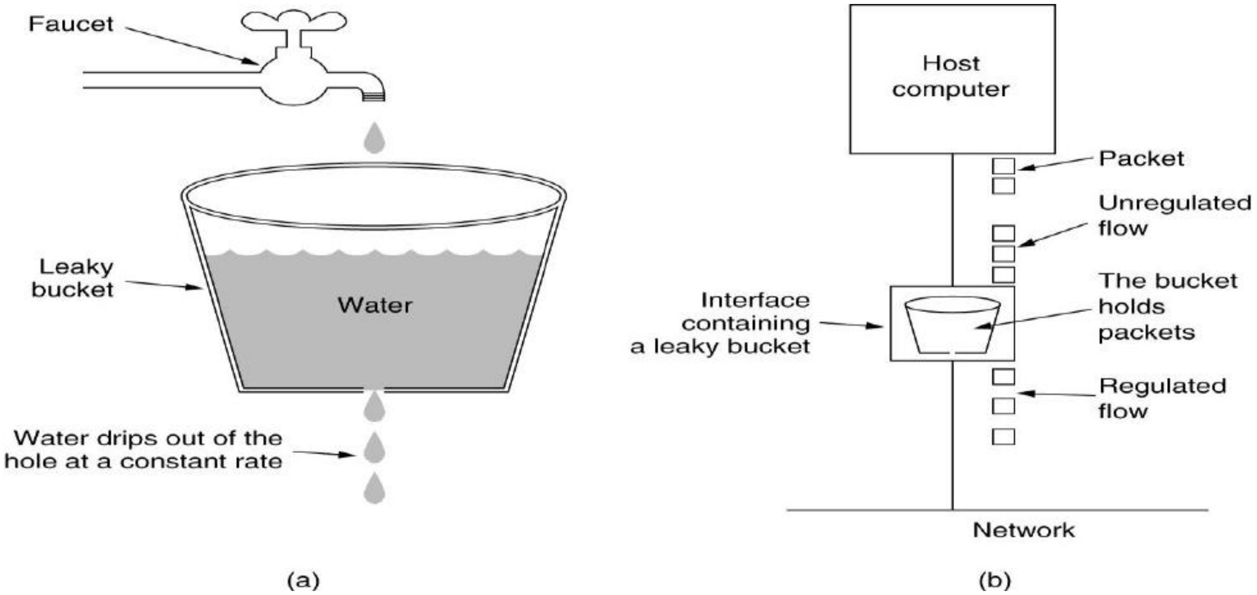
1. This is a proactive approach in which the router discards one or more packets *before* the buffer becomes completely full.
2. Each time a packet arrives, the RED algorithm computes the average queue length, ***avg***.
3. If *avg* is lower than some lower threshold, congestion is assumed to be minimal 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.

Traffic Shaping

1. Another method of congestion control is to "shape" the traffic before it enters the 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.

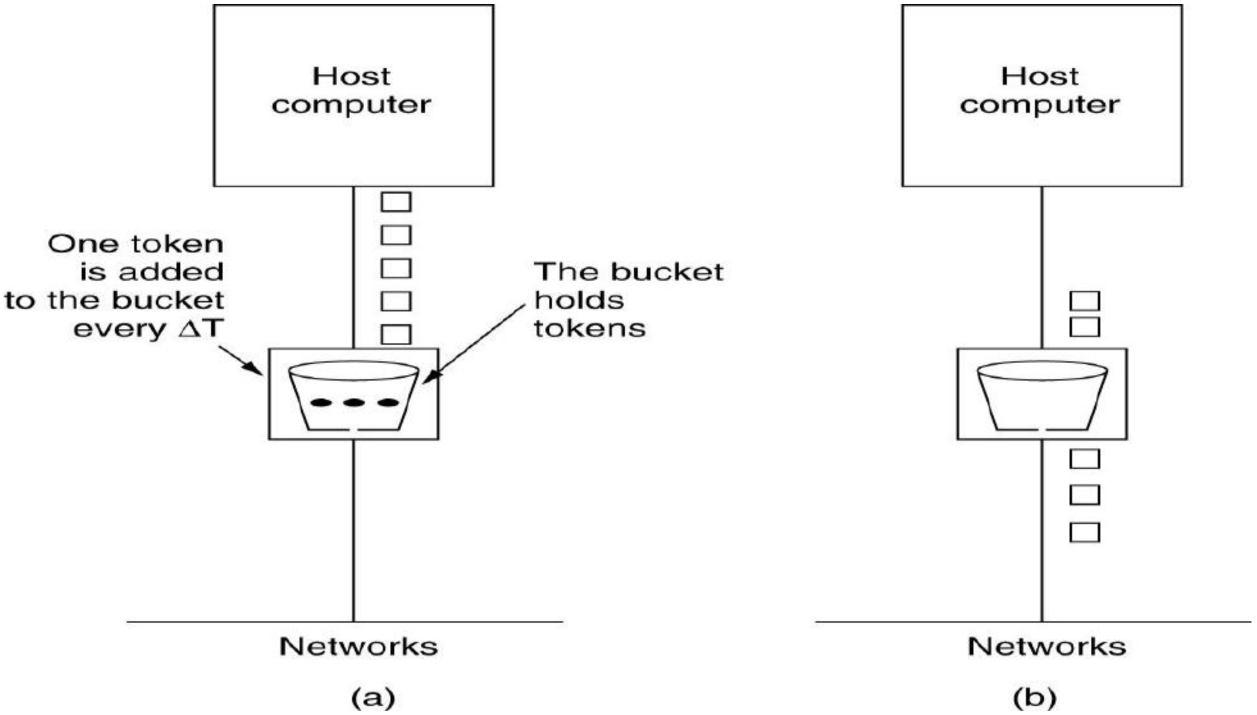


(a) A leaky bucket with water. (b) a leaky bucket with packets.

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

Token Bucket Algorithm

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 ∆t 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.

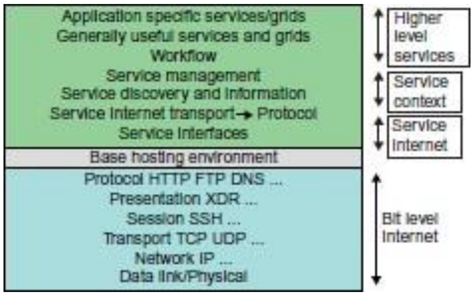


(a) Before. (b) After.

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.

- These interfaces are linked with customized, high-level communication systems: SOAP, RMI, and IIOP
- These communication systems support features including particular message patterns (such as Remote Procedure Call or RPC), fault recovery, and specialized routing.
- Communication systems are built on message-oriented middleware (enterprise bus) infrastructure such as Web-Sphere MQ or Java Message Service (JMS)

	Cases of fault tolerance- the features in the Web Services Reliable Messaging (WSRM) Security -reimplements the capabilities seen in concepts such as Internet Protocol Security (IPsec)
	Several models with, for example, JNDI (Jini and Java Naming and DirectoryInterface) illustrating different approaches within the Java distributed object model. The CORBA TradingService, UDDI (Universal Description, Discovery, and Integration), LDAP (Lightweight Directory Access Protocol), and ebXML (Electronic Business using eXtensibleMarkup Language
	earlier years, CORBA and Java approaches were used in distributed systems rather than today'sSOAP, XML, or REST (Representational State Transfer).

Web Services and Tools

REST approach:

- delegates most ofthe difficult problems to application (implementation-specific) software. In a web services language
- minimal information in the header, and the message body (that is opaque to genericmessage processing) carries all the needed information.
- architectures are clearly more appropriatefor rapid technology environments.
- REST can use XML schemas but not those that are part of SOAP; "XML overHTTP" is a popular design choice in this regard.
- Above the communication and managementlayers, we have the ability to compose new entities or distributed programs by integrating severalentities together.

CORBA and Java:

- the distributed entities are linked with RPCs, and the simplest way to buildcomposite applications is to view the entities as objects and use the traditional ways of linking themtogether.
- For Java, this could be as simple as writing a Java program with method calls replaced byRemote Method Invocation (RMI),
- CORBA supports a similar model with a syntax reflecting theC++ style of its entity (object) interfaces.

Parallel and Distributed Programming Models

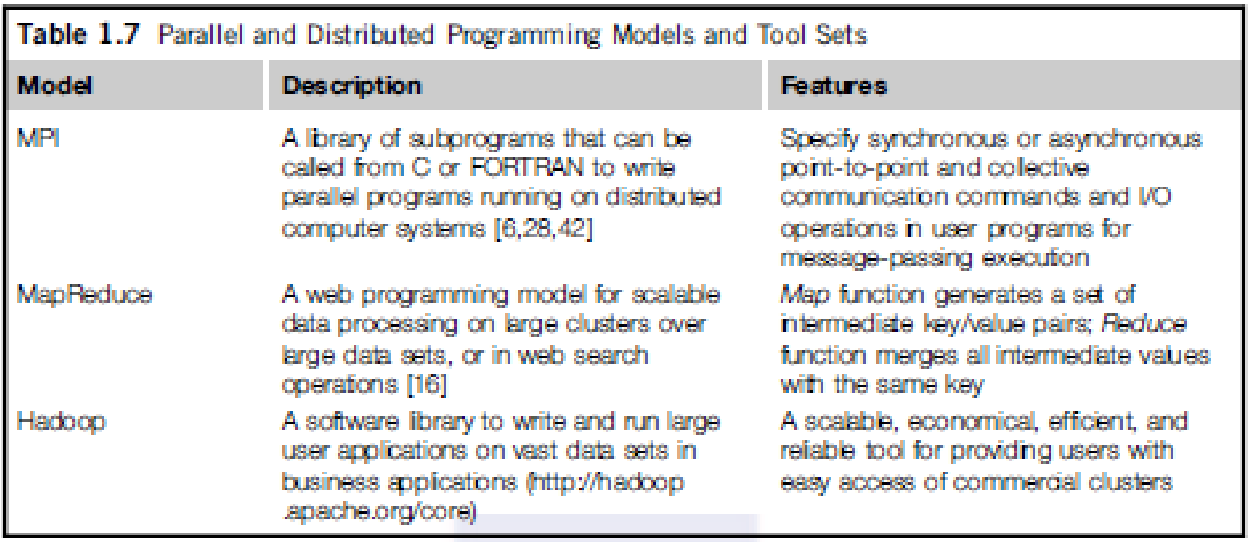


Table 1.7 Parallel and Distributed Programming Models and Tool Sets

Model	Description	Features
MPI	A library of subprograms that can be called from C or FORTRAN to write parallel programs running on distributed computer systems [6,28,42]	Specify synchronous or asynchronous point-to-point and collective communication commands and I/O operations in user programs for message-passing execution
MapReduce	A web programming model for scalable data processing on large clusters over large data sets, or in web search operations [16]	*Map* function generates a set of intermediate key/value pairs; *Reduce* function merges all intermediate values with the same key
Hadoop	A software library to write and run large user applications on vast data sets in business applications (http://hadoop .apache.org/core)	A scalable, economical, efficient, and reliable tool for providing users with easy access of commercial clusters

PERFORMANCE, SECURITY, AND ENERGY EFFICIENCY

Performance Metrics:

- In a distributed system, performance is attributed to a large numberof factors.
- System throughput is often measured in MIPS, Tflops (tera floating-point operations persecond), or TPS (transactions per second).
- Systemoverhead is often attributed to OS boot time, compile time, I/O data rate, and the runtime support systemused.
- Other performance-related metrics include the QoS for Internet and web services; systemavailability and dependability; and security resilience for system defense against network attacks

Dimensions of Scalability

Any resource upgrade ina system should be backward compatible with existing hardware and software resources. System scaling can increase or decrease resources depending on many practicalfactors

Size scalability

- This refers to achieving higher performance or more functionality by increasingthe machine size.
- The word “size” refers to adding processors, cache, memory, storage, or I/Ochannels. The most obvious way to determine size scalability is to simply count the number ofprocessors installed.

- Not all parallel computer or distributed architectures are equally sizescalable.
- For example, the IBM S2 was scaled up to 512 processors in 1997. But in 2008, theIBMBlueGene/L system scaled up to 65,000 processors.

• **Software scalability**

  - This refers to upgrades in the OS or compilers, adding mathematical andengineering libraries, porting new application software, and installing more user-friendlyprogramming environments.
  - Some software upgrades may not work with large systemconfigurations.
  - Testing and fine-tuning of new software on larger systems is a nontrivial job.

• **Application scalability**

  - This refers to matching problem size scalability with machine sizescalability.
  - Problem size affects the size of the data set or the workload increase. Instead of increasingmachine size, users can enlarge the problem size to enhance system efficiency or cost-effectiveness.

• **Technology scalability**

  - This refers to a system that can adapt to changes in building technologies,such as the component and networking technologies
  - Whenscaling a system design with new technology one must consider three aspects: time, space, andheterogeneity.
  - (1) Time refers to generation scalability. When changing to new-generation processors,one must consider the impact to the motherboard, power supply, packaging and cooling,and so forth. Based on past experience, most systems upgrade their commodity processors everythree to five years.
  - (2) Space is related to packaging and energy concerns. Technology scalabilitydemands harmony and portability among suppliers.
  - (3) Heterogeneity refers to the use ofhardware components or software packages from different vendors. Heterogeneity may limit thescalability.

Amdahl's Law

- Let the program has been parallelized or partitioned for parallelexecution on a cluster of many processing nodes.
- Assume that a fraction α of the code must be executedsequentially, called the sequential bottleneck.
- Therefore, (1 − α) of the code can be compiledfor parallel execution by n processors.

  The total execution time of the program is calculated byα T + (1 − α)T/n, where the first term is the sequential execution time on a single processor and thesecond term is the parallel execution time on n processing nodes.
- I/O time or exception handling timeis also not included in the following speedup analysis.

$$\text{Speedup} = S = T/[\alpha T + (1-\alpha)T/n] = 1/[\alpha + (1-\alpha)/n]$$

- Amdahl's Law states that the speedup factorof using the n-processor system over the use of a single processor is expressed by:

- the code is fully parallelizable with α = 0. As the cluster becomes sufficiently large, that is, n →∞, S approaches 1/α, an upper bound on the speedup S.

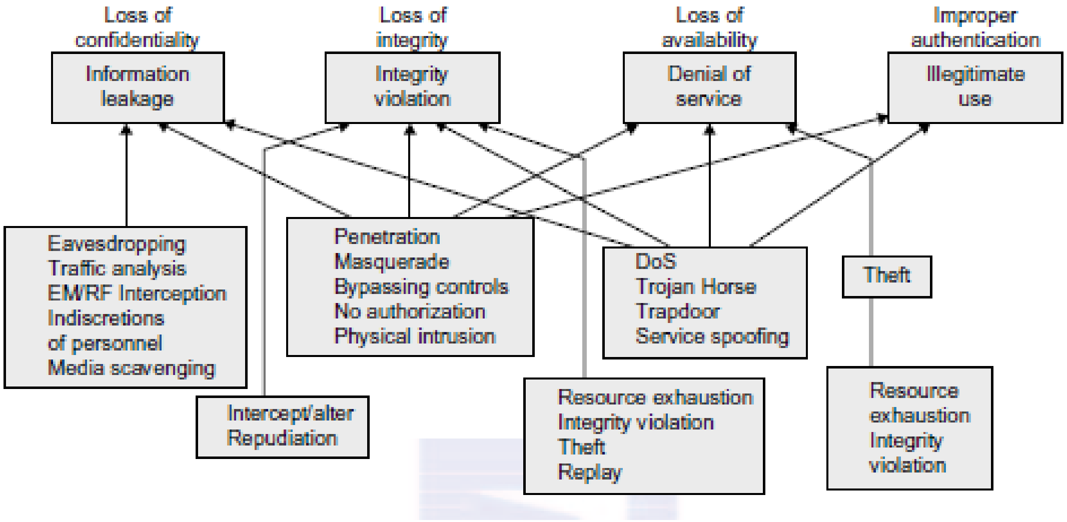
- this upper bound is independentof the cluster size n. The sequential bottleneck is the portion of the code that cannot be parallelized.

Gustafson's Law

- To achieve higher efficiency when using a large cluster, we must consider scaling the problem sizeto match the cluster capability. This leads to the following speedup law proposed by John Gustafson(1988), referred as scaled-workload speedup.
- Let W be the workload in a given program.
- When using an n-processor system, the user scales the workload to W′ = αW + (1 − α)nW.Scaled workload W′ is essentially the sequential execution time on a single processor. The parallelexecution time of a scaled workload W′ on n processors is defined by a scaled-workload speedupas follows:

$$S = W'/W = [\alpha W + (1-\alpha)nW]/W = \alpha + (1-\alpha)n$$

Network Threats and Data Integrity



ENERGY EFFICIENCY IN DISTRIBUTED COMPUTING

Primary performance goals in conventional parallel and distributed computing systems are highperformance and high throughput, considering some form of performance reliability (e.g., fault toleranceand security). However, these systems recently encountered new challenging issues includingenergy efficiency, and workload and resource outsourcing

Energy Consumption of Unused Servers: To run a server farm (data center) a company has to spend a huge amount of money for hardware,software, operational support, and energy every year. Therefore, companies should thoroughlyidentify whether their installed server farm (more specifically, the volume of provisioned resources)is at an appropriate level, particularly in terms of utilization.

Reducing Energy in Active Servers: In addition to identifying unused/underutilized servers for energy savings, it is also necessary toapply appropriate techniques to decrease energy consumption in active distributed systems with negligibleinfluence on their performance.

Application Layer: Until now, most user applications in science, business, engineering, and financial areas tend toincrease a system's speed or quality. By introducing energy-aware applications, the challenge is todesign sophisticated multilevel and multi-domain energy management applications without hurtingperformance.

Middleware Layer: The middleware layer acts as a bridge between the application layer and the resource layer. Thislayer provides resource broker, communication service, task analyzer, task scheduler, securityaccess, reliability control, and information service capabilities. It is also responsible for applyingenergy-efficient techniques, particularly in task scheduling.

Resource Layer: The resource layer consists of a wide range of resources including computing nodes and storageunits. This layer generally interacts with hardware devices and the operating system; therefore, itis responsible for controlling all distributed resources in distributed computing systems. Dynamic power management (DPM) and dynamic voltage-frequency scaling (DVFS) are two popular methods incorporated into recent computer hardware systems. In DPM, hardware devices, such as the CPU, have the capability to switch from idle mode to one or more lower power modes. In DVFS, energy savings are achieved based on the fact that the power consumptionin CMOS circuits has a direct relationship with frequency and the square of the voltage supply.

Network Layer: Routing and transferring packets and enabling network services to the resource layer are the mainresponsibility of the network layer in distributed computing systems. The major challenge to buildenergy-efficient networks is, again, determining how to measure, predict, and create a balancebetween energy consumption and performance.

Clustering for Massive Parallelism.

- A computer cluster is a collection of interconnected stand-alone computers which can work together collectively and cooperatively as a single integrated computing resource pool.
- Clustering explores massive parallelism at the job level and achieves high availability (HA) through stand-alone operations.
- Benefits of computer clusters and massively parallel processors (MPPs) include

- Scalable performance, HA, fault tolerance, modular growth, and use of commodity components. These features can sustain the generation changes experienced in hardware, software, and network components.

Design Objectives of Computer Clusters

Scalability:

- Clustering of computers is based on the concept of modular growth. To scale a cluster from hundreds of uniprocessor nodes to a supercluster with 10,000 multicore nodes is a nontrivial task.
- The scalability could be limited by a number of factors, such as the multicore chip technology, cluster topology, packaging method, power consumption, and cooling scheme applied.

Packaging

- Cluster nodes can be packaged in a compact or a slack fashion. In a compact cluster, the nodes are closely packaged in one or more racks sitting in a room, and the nodes are not attached to peripherals (monitors, keyboards, mice, etc.).
- In a slack cluster, the nodes are attached to their usual peripherals (i.e., they are complete SMPs, workstations, and PCs), and they may be located in different rooms, different buildings, or even remote regions.
- Packaging directly affects communication wire length, and thus the selection of interconnection technology used.
- While a compact cluster can utilize a high-bandwidth, low-latency communication network that is often proprietary, nodes of a slack cluster are normally connected through standard LANs or WANs.

Control

- A cluster can be either controlled or managed in a centralized or decentralized fashion. A compact cluster normally has centralized control, while a slack cluster can be controlled either way.
- In a centralized cluster, all the nodes are owned, controlled, managed, and administered by a central operator.
- In a decentralized cluster, the nodes have individual owners. This lack of a single point of control makes system administration of such a cluster very difficult. It also calls for special techniques for process scheduling, workload migration, checkpointing, accounting, and other similar tasks.

Homogeneity

- A homogeneous cluster uses nodes from the same platform, that is, the same processor architecture and the same operating system; often, the nodes are from the same vendors.
- A heterogeneous cluster uses nodes of different platforms. Interoperability is an important issue in heterogeneous clusters.
- In a homogeneous cluster, a binary process image can migrate to another node and continue execution.

- This is not feasible in a heterogeneous cluster, as the binary code will not be executable when the process migrates to a node of a different platform.

Security

- Intracluster communication can be either exposed or enclosed.
- In an exposed cluster, the communication paths among the nodes are exposed to the outside world. An outside machine can access the communication paths, and thus individual nodes, using standard protocols (e.g., TCP/IP).
- Such exposed clusters are easy to implement, but have several disadvantages:

  • Being exposed, intracluster communication is not secure, unless the communication subsystemperforms additional work to ensure privacy and security.

- Outside communications may disrupt intracluster communications in an unpredictable fashion.
- Standard communication protocols tend to have high overhead.
- In an enclosed cluster, intracluster communication is shielded from the outside world, which
- alleviates the aforementioned problems.
- A disadvantage is that there is currently no standard for efficient, enclosed intracluster communication. Consequently, most commercial or academic clusters realize fast communications through one-of-a-kind protocols

Fundamental Cluster Design Issues

1. **Scalable Performance: S**caling of resources (cluster nodes, memory capacity, I/O bandwidth, etc.) leads to a proportional increase in performance. Both scale-up and scale-down capabilities are needed, depending on application demand or cost-effectiveness considerations. Clustering is driven by scalability

2. **Single-System Image (SSI):** A set of workstations connected by an Ethernet network is not necessarily a cluster. A cluster is a single system.

3. **Availability Support:** Clusters can provide cost-effective HA capability with lots of redundancy in processors, memory, disks, I/O devices, networks, and operating system images

4. **Cluster Job Management:** Clusters try to achieve high system utilization from traditional workstations or PC nodes that are normally not highly utilized. Job management software is required to provide batching, load balancing, parallel processing, and other functionality

5. **Inter node Communication:** The inter node physical wire lengths are longer in a cluster than in an MPP. A long wire implies greater interconnect network latency. But, longer

wires have more problems in terms of reliability, clock skew, and cross talking. These problems call for reliable and secure communication protocols, which increase overhead. Clusters often use commodity networks (e.g., Ethernet) with standard protocols such as TCP/IP.

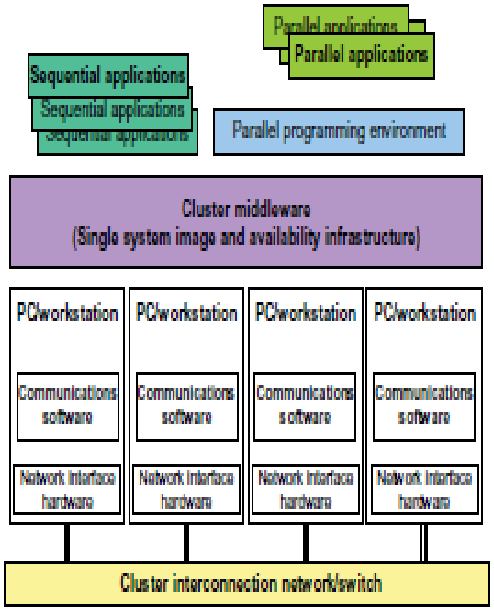
6. **Fault Tolerance and Recovery:** Clusters of machines can be designed to eliminate all single points of failure. Through redundancy, a cluster can tolerate faulty conditions up to a certain extent. Heartbeat mechanisms can be installed to monitor the running condition of all nodes. In case of a node failure, critical jobs running on the failing nodes can be saved by failing over to the surviving node machines. Rollback recovery schemes restore the computing results through periodic checkpointing.

7. **Cluster Family Classification:**computer clusters are divided into three classes

- **Compute clusters:**
  - These are clusters designed mainly for collective computationover a single large job. The compute clusters do not handle many I/O operations, such as database services. When a single compute job requires frequent communication among the cluster nodes, the cluster must share a dedicated network, and thus the nodes are mostly homogeneous and tightly coupled. This type of clusters is also known as a **Beowulf cluster**
- **High-Availability clusters** HA (high-availability)
  - clusters are designed to be fault-tolerant and achieve HA of services. HA clusters operate with many redundant nodes to sustain faults or failures.
- **Load-balancing clusters**
  - These clusters shoot for higher resource utilization through load balancing among all participating nodes in the cluster. All nodes share the workload or function as a single virtual machine (VM).
  - Requests initiated from the user are distributed to all node computers to form a cluster. This results in a balanced workload among different machines, and thus higher resource utilization or higher performance. Middleware is needed to achieve dynamic load balancing by job or process migration among all the cluster nodes

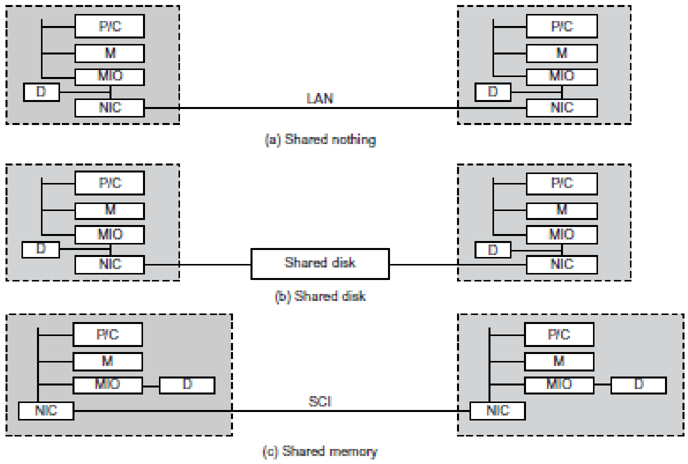
Basic Cluster Architecture

- simple cluster of computers built with commodity components supported with desired SSI features and HA capability
- commodity nodes are easy to replace or upgrade with new generations of hardware
- node operating systems should be designed for multiuser, multitasking, and multithreaded applications.
- nodes are interconnected by one or more fast commodity networks and use standard communication protocols
- network interface card is connected to the node’s standard I/O bus

	When the processor or the operating system is changed, only the driver software needs to change.
	a cluster OS is not commercially available. Instead, we can deploy some cluster middleware to glue together all node platforms at the user space
	• middleware offers HA services • An SSI layer provides a single entry point, a single file hierarchy, a single point of control • idealized cluster is supported by three subsystems • conventional databasesand OLTP monitors • A user interface subsystemis needed to combine the advantages of the web interface and the Windows GUI. • cluster supports parallel programming based on standard languages and communication libraries using PVM, MPI, or OpenMP. The programming environment also includes tools for debugging, profiling, monitoring

Resource Sharing in Clusters

Clustering improves both availability and performance. Some HA clusters use hardware redundancy for scalable performance. The nodes of a cluster can be connected in one of three ways

 (a) Shared nothing (b) Shared disk (c) Shared memory	Part (a) simply connects two or more autonomous computers via a LAN such as Ethernet. Part (b): shared-disk cluster Part (c) shared-memory cluster

DESIGN PRINCIPLES OF COMPUTER CLUSTERS

General-purpose computers and clusters of cooperative computers should be designed for scalability, availability, Single System Image, High Availability, Fault tolerance, and Rollback recovery

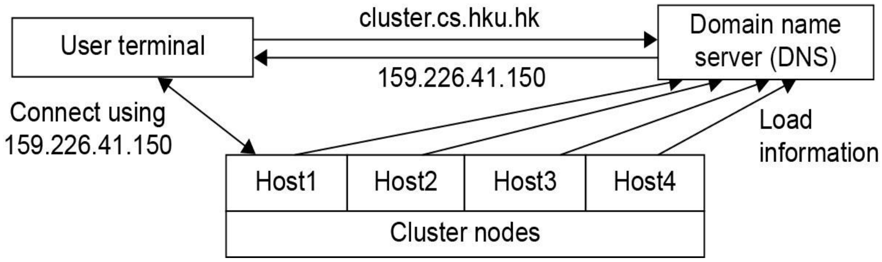
- **Single System Image**: A single system image is the illusion, created by software or hardware, that presents a collection of resources as an integrated powerful resource. SSI makes the cluster appear like a single machine to the user, applications, and network. A cluster with multiple system images is nothing but a collection of independent computers **Single-System-Image Features**

  - **Single System**: The entire cluster is viewed by the users as one system, which has multiple processors.
  - **Single Control**: Logically, an end user or system user utilizes services from one place with a single interface.
  - **Symmetry**: A user can use a cluster service from any node. All cluster services and functionalities are symmetric to all nodes and all users, except those protected by access rights.
  - **Location Transparent**: The user is not aware of the whereabouts of the physical device that eventually provides a service.

Basic SSI Services

A. **Single Entry Point**

```
telnet cluster.usc.edu
telnet node1.cluster.usc.edu
```



1. Four nodes of a cluster are used as host nodes to receive users' login requests.
2. To log into the cluster a standard Unix command such as "telnet cluster.cs.hku.hk", using the symbolic name of the cluster system is issued.
3. The symbolic name is translated by the DNS, which returns with the IP address 159.226.41.150 of the least-loaded node, which happens to be node Host1.
4. The user then logs in using this IP address.
5. The DNS periodically receives load information from the host nodes to make load-balancing translation decisions.

B. Single File Hierarchy: xFS, AFS, Solaris MC Proxy

The illusion of a single, huge file system image that transparently integrates local and global disks and other file devices (e.g., tapes). Files can reside on 3 types of locations in a cluster:

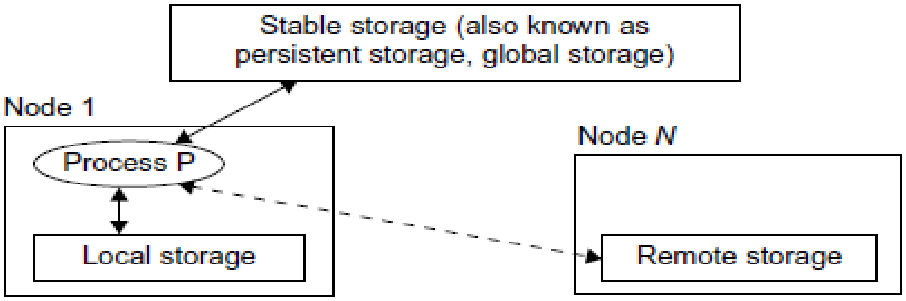
Local storage - disk on the local node.

Remote storage - disks on remote nodes.

Stable storage -

Persistent - data, once written to the stable storage, will stay there at least for a period of time (e.g., a week), even after the cluster shuts down.

Fault tolerant - to some degree, by using redundancy and periodical backup to tapes.


Stable storage (also known as persistent storage, global storage)
Node 1
Process P
Local storage
Node N
Remote storage

Three types of storage in a single file hierarchy. Solid lines show what process P can access and thedashed line shows what P may be able to access

C. Single I/O, Networking, and Memory Space: To achieve SSI, we need a:

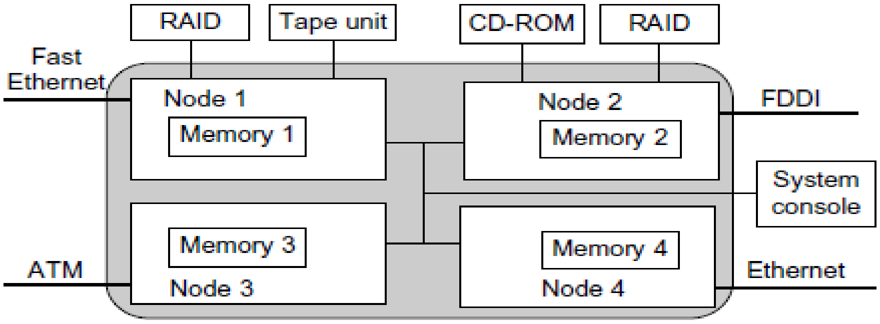
- single control point
- single address space
- single job management system
- single user interface
- single process control

Single Networking: A properly designed cluster should behave as one system. Any process on any node can use any network and I/O device as though it were attached to the local node. Single networking means any node can access any network connection.

Single Point of Control: The system administrator should be able to configure, monitor, test, and control the entire cluster and each individual node from a single point. Many clusters help with this through a system console that is connected to all nodes of the cluster

Single Memory Space: Single memory space gives users the illusion of a big, centralized main memory, which in reality may be a set of distributed local memory spaces.

Single I/O Address Space: A single I/O space implies that any node can access the RAIDs



A cluster with single networking, single I/O space, single memory, and single point of control

Other Services

Single Job Management: All cluster jobs can be submitted from any node to a single job management system. GlUnix, Codine, LSF, etc.

Single User Interface: The users use the cluster through a single graphical interface. Such an interface is available for workstations and PCs like CDE in Solaris/NT

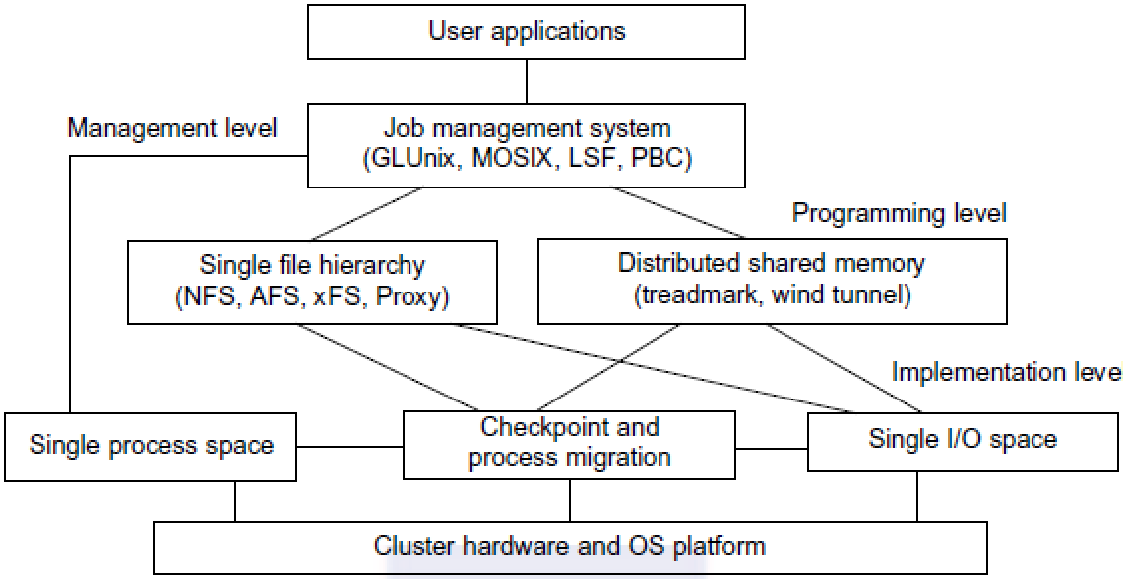
Single process space All user processes created on various nodes form a single process space and share a uniform process identification scheme. A process on any node can create(e.g., through a UNIX fork) or communicate with (e.g., through signals, pipes, etc.) processes on remote nodes.

Middleware support for SSI clusteringSSI features aresupported by middleware developed at three cluster application levels:

• **Management level** This level handles user applications and provides a job management system such as GLUnix, MOSIX, Load Sharing Facility (LSF), or Codine.

• **Programming level**This level provides single file hierarchy (NFS, xFS, AFS, Proxy) and distributed shared memory (TreadMark, Wind Tunnel).

• **Implementation level** This level supports a single process space, checkpointing, process migration, and a single I/O space. These features must interface with the cluster hardware and OS platform.



Relationship among clustering middleware at the job management, programming, and implementation levels.

- **High Availability through Redundancy:**

  - **Reliability** measures how long a system can operate without a breakdown.
  - **Availability** indicates the percentage of time that a system is available to the user, that is, thepercentage of system uptime.

• **Serviceability** refers to how easy it is to service the system, including hardware and softwaremaintenance, repair, upgrades, and so on.

A system's reliability is measured by the mean time to failure (MTTF), which is the average time of normal operation before the system (or a component of the system) fails. The metricfor serviceability is the mean time to repair (MTTR), which is the average time it takes to repair thesystem and restore it to working condition after it fails.

The availability of a system is defined by:

Availability = MTTF/ (MTTF +MTTR)

Failure is any event that prevents the system from normal operation

- **Unplanned failures** The system breaks, due to an operating system crash, a hardware failure, anetwork disconnection, human operation errors, a power outage, and so on. All these are simplycalled failures. The system must be repaired to correct the failure.
- **Planned shutdowns**The system is not broken, but is periodically taken off normal operationfor upgrades, reconfiguration, and maintenance.

Transient versus Permanent Failures

A lot of failures are **transient** in that they occur temporarily and then disappear. They can be dealtwith without replacing any components. A standard approach is to roll back the system to a known state and start over.

Permanent failures cannot be corrected by rebooting. Some hardwareor software component must be repaired or replaced. For instance, rebooting will not work ifthe system hard disk is broken.

Partial versus Total Failures

A failure that renders the entire system unusable is called a **total** failure. A failure that only affectspart of the system is called a **partial** failure if the system is still usable, even at a reduced capacity

Redundancy Techniques

Isolated Redundancy: A key technique to improve availability in any system is to use redundant components. When acomponent (the primary component) fails, the service it provided is taken over by another component(the backup component). Furthermore, the primary and the backup components should be isolatedfrom each other, meaning they should not be subject to the same cause of failure. Clustersprovide HA with redundancy in power supplies, fans, processors, memories, disks, I/O devices, networks,operating system images, and so on. In a carefully designed cluster, redundancy is also isolated.

N-Version Programming to Enhance Software Reliability

A common isolated-redundancy approach to constructing a mission-critical software system is calledN-version programming. The software is implemented by N isolated teams who may not even knowthe others exist. Different teams are asked to implement the software using different algorithms,programming languages, environment tools, and even platforms In a fault-tolerant system, the N versions all run simultaneously and their results are constantly compared. If the results differ, thesystem is notified that a fault has occurred.

- **Fault-Tolerant Cluster Configurations:** The cluster solution was targeted to provide availability support for two server nodes with threeascending levels of availability: hot standby, active takeover, and fault-tolerant. The level of availabilityincreases from standby to active and fault-tolerant cluster configurations. The shorter is the recoverytime, the higher is the cluster availability. Failback refers to the ability of a recovered node returningto normal operation after repair or maintenance. Activeness refers to whether the node is used inactive work during normal operation.

- **Hot standby server clusters**: In a hot standby cluster, only the primary node is actively doing all the useful work normally. The standby node is powered on (hot) and running some monitoring programs to communicate heartbeat signals to check the status of the primary node, but is not actively running other useful workloads. The primary node must mirror any data to shared disk storage, which is accessible by the standby node. The standby node requires a second copy of data.

- **Active-takeover clusters**: In this case, the architecture is symmetric among multiple server nodes. Both servers are primary, doing useful work normally. Both failover and failback are often supported on both server nodes. When a node fails, the user applications fail over to the available node in the cluster. Depending on the time required to implement the failover, users may experience some delays or may lose some data that was not saved in the last checkpoint.

- **Failover cluster:** When a component fails, this technique allows the remaining system to take over the services originally provided by the failed component. A failover mechanism mustprovide several functions, such as failure diagnosis, failure notification, and failure recovery.Failure diagnosis refers to the detection of a failure and the location of the failed componentthat caused the failure. A commonly used technique is heartbeat, whereby the cluster nodessend out a stream of heartbeat messages to one another. If the system does not receive thestream of heartbeat messages from a node, it can conclude that either the node or the networkconnection has failed.

Recovery Schemes

Failure recovery refers to the actions needed to take over the workload of a failed component. Thereare two types of recovery techniques. In **backward recovery**, the processes running on a cluster periodicallysave a consistent state (called a checkpoint) to a stable storage. After a failure, the systemis reconfigured to isolate the failed component, restores the previous checkpoint, and resumes normaloperation. This is called rollback. Backward recovery is relatively easy to implement in an application-independent, portable fashion

If execution time is crucial,such as in real-time systems where the rollback time cannot be tolerated, a **forward recovery** schemeshould be used. With such a scheme, the system is not rolled back to the previous checkpoint upon afailure. Instead, the system utilizes the failure diagnosis information to reconstruct a valid system stateand continues execution. Forward recovery is application-dependent and may need extra hardware

Checkpointing and Recovery Techniques

Checkpointingisthe process of periodically saving the state of an executing program to stable storage, from whichthe system can recover after a failure. Each program state saved is called a checkpoint. The disk filethat contains the saved state is called the checkpoint file. Checkpointing techniques are useful not only for availability, but also for program debugging, process migration, and load balancing

Checkpointing can be realized by the operating system at the **kernel level**, where the OS transparentlycheckpoints and restarts processes
A less transparent approach linksthe user code with a checkpointing**library in the user space**. Checkpointing and restarting are handled by this runtime support. This approach is used widely because it has the advantage thatuser applications do not have to be modified.

A third approach requires the **user (or the compiler)** to insert checkpointingfunctions in the application; thus, the application has to be modified, and the transparencyis lost. However, it has the advantage that the user can specify where to checkpoint. This is helpful to reduce checkpointing overhead. Checkpointing incurs both time and storage overheads.

Checkpoint Overheads
During a program's execution, its states may be saved many times. This is denoted by the time consumedto save one checkpoint. The storage overhead is the extra memory and disk space requiredfor checkpointing. Both time and storage overheads depend on the size of the checkpoint file.

Choosing an Optimal Checkpoint Interval

The time period between two checkpoints is called the checkpoint interval. Making the interval larger can reduce checkpoint time overhead.
Wong and Franklin derived an expression for optimal checkpoint interval

Optimal checkpoint interval = Square root (MTTF x t_c)/h

MTTF is the system's mean time to failure. This MTTF accounts the time consumed to save one checkpoint, and h is the average percentage of normal computation performed in a checkpointinterval before the system fails. The parameter h is always in the range. After a system isrestored, it needs to spend h × (checkpoint interval) time to recompute.

Incremental Checkpoint
Instead of saving the full state at each checkpoint, an incremental checkpoint scheme saves only theportion of the state that is changed from the previous checkpoint In full-state checkpointing, only one checkpoint file needs to be kepton disk. Subsequent checkpoints simply overwrite this file. With incremental checkpointing, old filesneeded to be kept, because a state may span many files. Thus, the total storage requirement is larger

Forked Checkpointing
Most checkpoint schemes are blocking in that the normal computation is stopped while checkpointingis in progress. With enough memory, checkpoint overhead can be reduced by

making a copy ofthe program state in memory and invoking another asynchronous thread to perform the checkpointingconcurrently. A simple way to overlap checkpointing with computation is to use the UNIXfork( ) system call. The forked child process duplicates the parent process's address space andcheckpoints it. Meanwhile, the parent process continues execution. Overlapping is achieved sincecheckpointing is disk-I/O intensive

User-Directed Checkpointing
The checkpoint overheads can sometimes be substantially reduced if the user inserts code (e.g., library or system calls) to tell the system when to save, what to save, and what not to save. What should be the exact contents of a checkpoint? It should contain just enough information to allow asystem to recover. The state of a process includes its data state and control state

Checkpointing Parallel Programs The state of a parallel program is usually much larger than that of a sequential program, as it consists of the set of the states of individual processes, plus thestate of the communication network. Parallelism also introduces various timing and consistency problems

Consistent Snapshot
A global snapshot is called consistent if there is no message that is received by the checkpoint of one process, but not yet sent by another process. Graphically, this corresponds to the case that no arrow crosses a snapshot line from right to left

Coordinated versus Independent Checkpointing
Checkpointing schemes for parallel programs can be classified into two types. In coordinated checkpointing(also called consistent checkpointing), the parallel program is frozen, and all processes arecheckpointed at the same time. In independent checkpointing, the processes are checkpointed independentof one another.

Cluster Job Scheduling and Management

A Job Management System (*JMS*) should have three parts:

- A **user server** lets the user submit jobs to one or more queues, specify resource requirements for each job, delete a job from a queue, inquire about the status of a job or a queue.
- A **job scheduler** that performs job scheduling and queuing according to job types, resource requirements, resource availability, and scheduling policies.
- A **resource manager** that allocates and monitors resources, enforces scheduling policies, and collects accounting information**.**

JMS Administration

- JMS should be able to dynamically reconfigure the cluster with minimal impact on the running jobs.
- The administrator's prologue and epilogue scripts should be able to run before and after each job for security checking, accounting, and cleanup.

- Users should be able to cleanly kill their own jobs.
- The administrator or the JMS should be able to cleanly suspend or kill any job.
    - Clean means that when a job is suspended or killed, all its processes must be included.
    - Otherwise some “orphan” processes are left in the system, wasting cluster resources and may eventually render the system unusable**.**

Several types of jobs execute on a cluster.

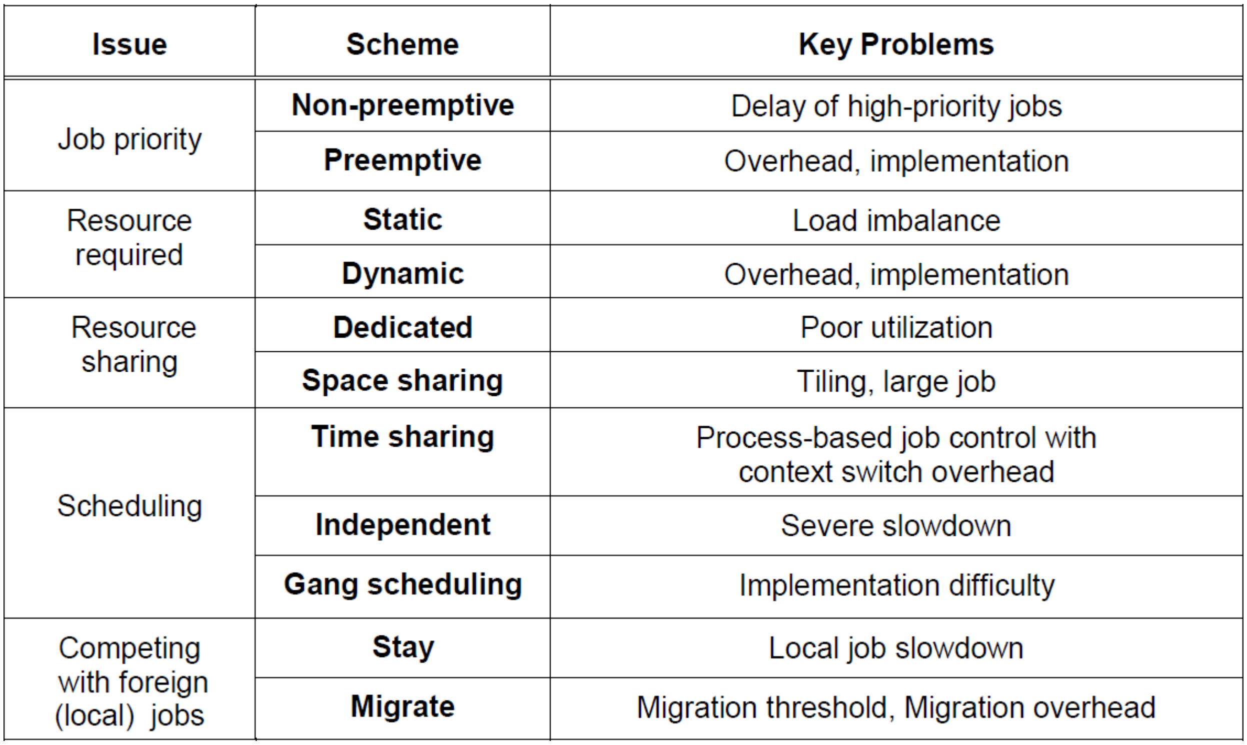
- Serial jobs run on a single node.
- Parallel jobs use multiple nodes.
- Interactive jobs are those that require fast turnaround time, and their input/output is directed to a terminal.
    - These jobs do not need large resources, and the users expect them to execute immediately, not made to wait in a queue.
- Batch jobs normally need more resources, such as large memory space and long CPU time.
    - But they do not need immediate response.
    - They are submitted to a job queue to be scheduled to run when the resource becomes available (e.g., during off hours).

Multi-Job Scheduling Schemes

- Cluster jobs may be scheduled to run at a specific time (**calendar scheduling**) or when a particular event happens (**event scheduling**).
- Jobs are scheduled according to priorities based on submission time, resource nodes, execution time, memory, disk, job type, and user identity.
- With **static priority**, jobs are assigned priorities according to a predetermined, fixed scheme.
    - A simple scheme is to schedule jobs in a first-come, first-serve fashion.
    - Another scheme is to assign different priorities to users.

With **dynamic priority**, the priority of a job may change over time.

Job Scheduling Issues and Schemes for Cluster Nodes



Issue	Scheme	Key Problems
Job priority	**Non-preemptive**	Delay of high-priority jobs
	Preemptive	Overhead, implementation
Resource required	**Static**	Load imbalance
	Dynamic	Overhead, implementation
Resource sharing	**Dedicated**	Poor utilization
	Space sharing	Tiling, large job
Scheduling	**Time sharing**	Process-based job control with context switch overhead
	Independent	Severe slowdown
	Gang scheduling	Implementation difficulty
Competing with foreign (local) jobs	**Stay**	Local job slowdown
	Migrate	Migration threshold, Migration overhead

Scheduling Modes

Dedicated Mode:

- Only one job runs in the cluster at a time, and at most one process of the job is assigned to a node at a time.
- The single job runs until completion before it releases the cluster to run other jobs.

Space Sharing:

Multiple jobs can run on disjoint partitions (groups) of nodes simultaneously.

- At most one process is assigned to a node at a time.
- Although a partition of nodes is dedicated to a job, the interconnect and the I/O subsystem may be shared by all jobs.

Time sharing :

- Multiple user processes are assigned to the same node.

Time-sharing introduces the following parallel scheduling policies:

- **Independent Scheduling (Independent):** Uses the operating system of each cluster node to schedule different processes as in a traditional workstation.
- **Gang Scheduling**: Schedules all processes of a parallel job together. When one process is active, all processes are active.

- **Competition with Foreign (Local) Jobs**: Scheduling becomes more complicated when both cluster jobs and local jobs are running. The local jobs should have priority over cluster jobs.

1. **Migration Scheme IssuesNode Availability**: Can the job find another available node to migrate to?
   - ➢ Berkeley study : Even during peak hours, 60% of workstations in a cluster are available.

2. **Migration Overhead**: The migration time can significantly slow down a parallel job.
   - ➢ Berkeley study : a slowdown as great as 2.4 times.
   - ➢ Slowdown is less if a parallel job is run on a cluster of twice the size.
   - ➢ e.g. a 32-node job on a 60-node cluster – migration slowdown no more than 20%, even when migration time of 3 minutes.

3. **Recruitment Threshold**: the amount of time a workstation stays unused before the cluster considers it an idle node.

UNIT -2

Levels of Virtualization Implementation

- Virtualization is a computer architecture technology by which multiple virtual machines (VMs) aremultiplexed in the same hardware machine.
- After virtualization, different user applications managed by their own operating systems (guest OS) can run onthe same hardware independent of the host OS
- done by adding additional software, called a virtualization layer
- This virtualization layer is known as hypervisor or virtual machine monitor (VMM)

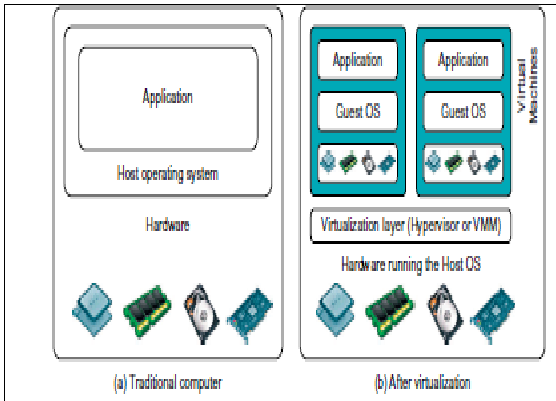
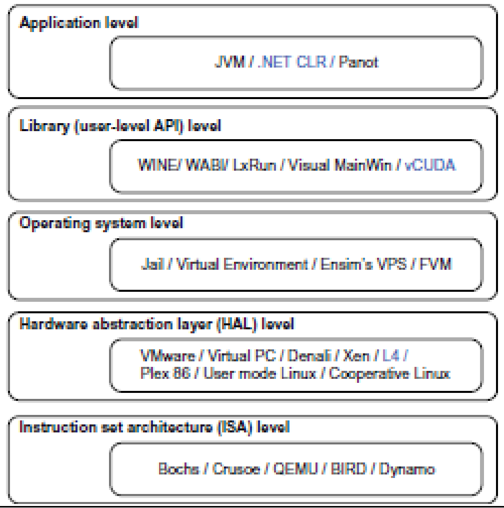


FIGURE 3.1

The architecture of a computer system before and after virtualization, where VMM stands for virtual machine monitor.


Application level
JVM / .NET CLR / Panot
Library (user-level API) level
WINE/ WABI/ LxRun / Visual MainWin / vCUDA
Operating system level
Jail / Virtual Environment / Ensim's VPS / FVM
Hardware abstraction layer (HAL) level
VMware / Virtual PC / Denali / Xen / L4 / Plex 86 / User mode Linux / Cooperative Linux
Instruction set architecture (ISA) level
Bochs / Crusoe / QEMU / BIRD / Dynamo

- function of the software layer for virtualization is to virtualize the physical hardware of a host machine into virtual resources to be used by the VMs
- Common virtualization layers include the instruction set architecture (ISA) level, hardware level, operating system level, library support level, and application level

Instruction Set Architecture Level

- At the ISA level, virtualization is performed by emulating a given ISA by the ISA of the host machine. For example, MIPS binary code can run on an x86-based host machine with the help of ISA emulation. With this approach, it is possible to run a large amount of legacy binary code written for various processors on any given new hardware host machine.
- Instruction set emulation leads to virtual ISAs created on any hardware machine. The basic emulation method is through code interpretation. An interpreter program interprets the source instructions to target instructions one by one. One source instruction may require tens or hundreds of native target instructions to perform its function. Obviously, this process is relatively slow. For better performance, dynamic binary translation is desired.
- This approach translates basic blocks of dynamic source instructions to target instructions. The basic blocks can also be extended to program traces or super blocks to increase translation efficiency.
- Instruction set emulation requires binary translation and optimization. A virtual instruction set architecture (V-ISA) thus requires adding a processor-specific software translation layer to the compiler.

Hardware Abstraction Level

- Hardware-level virtualization is performed right on top of the bare hardware.
- Thisapproach generates a virtual hardware environment for a VM.
- The process managesthe underlying hardware through virtualization. The idea is to virtualize a computer's resources, such asits processors, memory, and I/O devices.
- The intention is to upgrade the hardware utilization rate bymultiple users concurrently. The idea was implemented in the IBM VM/370 in the 1960s.
- Morerecently, the Xen hypervisor has been applied to virtualize x86-based machines to run Linux or otherguest OS applications.

Operating System Level

- This refers to an abstraction layer between traditional OS and user applications.
- OS-level virtualizationcreates isolated containers on a single physical server and the OS instances to utilize the hardwareand software in data centers.
- The containers behave like real servers.
- OS-level virtualization iscommonly used in creating virtual hosting environments to allocate hardware resources among alarge number of mutually distrusting users.
- It is also used, to a lesser extent, in consolidating serverhardware by moving services on separate hosts into containers or VMs on one server.

Library Support Level

- Most applications use APIs exported by user-level libraries rather than using lengthy system callsby the OS.
- Since most systems provide well-documented APIs, such an interface becomes anothercandidate for virtualization.
- Virtualization with library interfaces is possible by controlling the communicationlink between applications and the rest of a system through API hooks.

User-Application Level

- Virtualization at the application level virtualizes an application as a VM.
- On a traditional OS, anapplication often runs as a process. Therefore, application-level virtualization is also known as process-level virtualization.
- The most popular approach is to deploy high level language (HLL)VMs. In this scenario, the virtualization layer sits as an application program on top of the operatingsystem,
- The layer exports an abstraction of a VM that can run programs written and compiledto a particular abstract machine definition.
- Any program written in the HLL and compiled for thisVM will be able to run on it. The Microsoft .NET CLR and Java Virtual Machine (JVM) are twogood examples of this class of VM.

VMM Design Requirements and Providers

- layer between real hardware and traditional operating systems. This layer is commonly called the Virtual Machine Monitor (VMM)
- three requirements for a VMM
- a VMM should provide an environment for programs which is essentially identical to the original machine
- programs run in this environment should show, at worst, only minor decreases in speed
- VMM should be in complete control of the system resources.
- VMM includes the following aspects:
- (1) The VMM is responsible for allocating hardware resources for programs;
- (2) it is not possible for a program to access any resource not explicitly allocated to it;
- (3) it is possible under certain circumstances for a VMM to regain control of resources already allocated.

Virtualization Support at the OS Level

- Why OS-Level Virtualization? :
  - it is slow to initialize a hardware-level VM because each VM creates its own image from scratch.
- OS virtualization inserts a virtualization layer inside an operating system to partition a machine's physical resources.
- It enables multiple isolated VMs within a single operating system kernel.
- This kind of VM is often called a virtual execution environment (VE), Virtual Private System (VPS), or simply container
- The benefits of OS extensions are twofold:
  - (1) VMs at the operating system level have minimal startup/shutdown costs, low resource requirements, and high scalability;
  - (2) for an OS-level VM, it is possible for a VM and its host environment to synchronize state changes when necessary**.**

Middleware Support for Virtualization

- Library-level virtualization is also known as user-level Application Binary Interface (ABI) or API emulation.
- This type of virtualization can create execution environments for running alien programs on a platform

Hypervisor and Xen Architecture

- The hypervisor software sits directly between the physical hardware and its OS.
- This virtualization layer is referred to as either the VMM or the hypervisor

Xen Architecture

- Xen is an open source hypervisor program developed by Cambridge University.
- Xen is a microkernel hypervisor
- The core components of a Xen system are the hypervisor, kernel, and applications
- The **guest OS**, which has control ability, is called **Domain 0**, and the others are called **Domain U**
- Domain 0 is designed to access hardware directly and manage devices

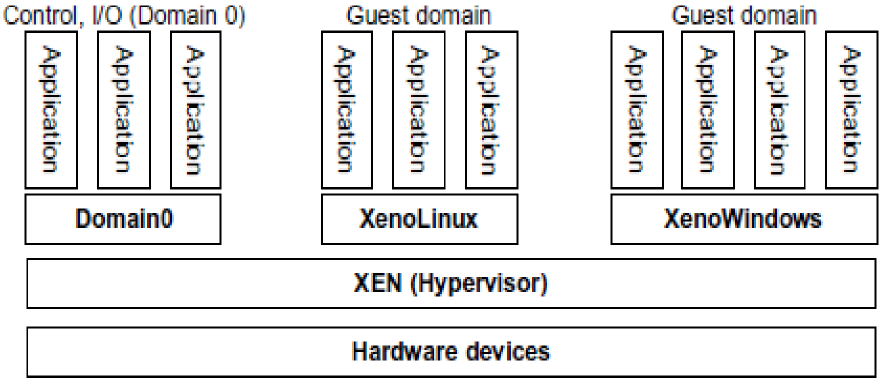


FIGURE 3.5

The Xen architecture's special domain 0 for control and I/O, and several guest domains for user applications.

(Courtesy of P. Barham, et al. [7])

- VM state is akin to a tree: the current state of the machine is a point that progresses monotonically as the software executes.
- VMs are allowed to roll back to previous states in their execution (e.g., to fix configuration errors) or rerun from the same point many times

Full virtualization

- Full virtualization, noncritical instructions run on the hardware directly while critical instructions are discovered and replaced with traps into the VMM to be emulated by software
- **VMware** puts the **VMM at Ring 0** and the **guest OS at Ring 1**.
- The VMM scans the instruction stream and identifies the privileged, control- and behavior-sensitive instructions.
- When these instructions are identified, they are trapped into the VMM, which emulates the behavior of these instructions.
- The method used in this emulation is called binary translation.
- Therefore, **full virtualization combines binary translation and direct execution.**

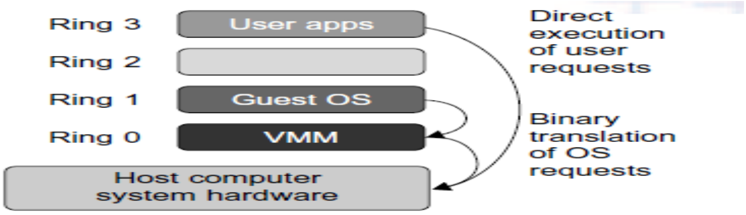

Ring 3
Ring 2
Ring 1
Ring 0
User apps
Guest OS
VMM
Host computer system hardware
Direct execution of user requests
Binary translation of OS requests

FIGURE 3.6

Indirect execution of complex instructions via binary translation of guest OS requests using the VMM plus direct execution of simple instructions on the same host.

(Courtesy of VM Ware [71])

Para-Virtualization

- Para-virtualization needs to modify the guest operating systems
- A para-virtualized VM provides special APIs requiring substantial OS modifications in user applications

CPU Virtualization

- A CPU architecture is virtualizable if it supports the ability to run the VM's privileged and unprivileged instructions in the CPU's user mode while the VMM runs in supervisor mode.
- Hardware-Assisted CPU Virtualization: This technique attempts to simplify virtualization because full or paravirtualization is complicated

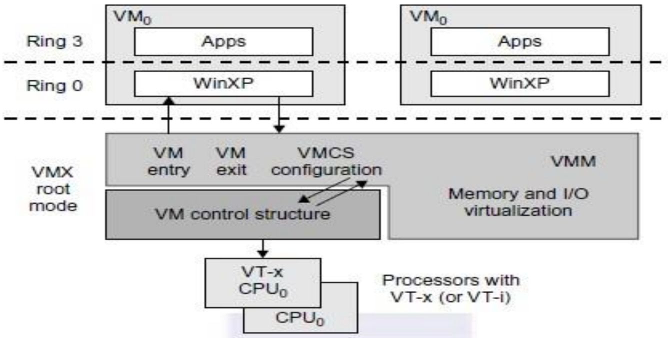


FIGURE 3.11

Intel hardware-assisted CPU virtualization.

(Modified from [68], Courtesy of Lizhong Chen, USC)

Memory Virtualization

- **Memory Virtualization** :the operating system maintains mappings of virtual memory to machine memory using page table
- All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance
- Two-stage mapping process should be maintained by the guest OS and the VMM, respectively: virtual memory to physical memory and physical memory to machine memory.
- The VMM is responsible for mapping the guest physical memory to the actual machine memory.

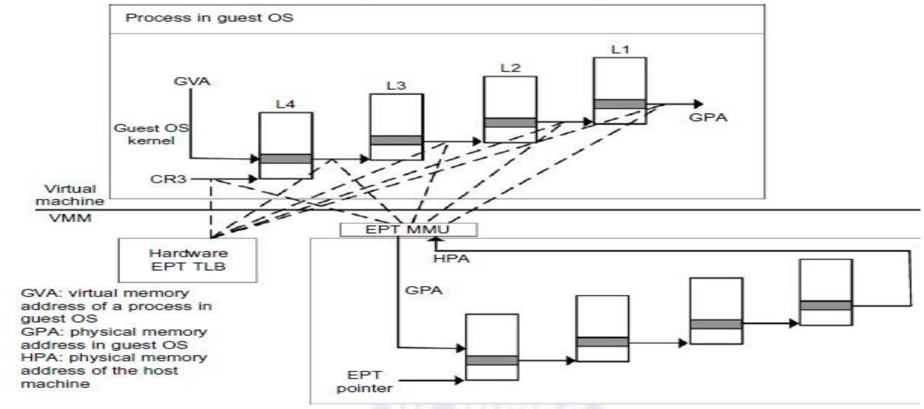

Process in guest OS
GVA
Guest OS kernel
CR3
L4
L3
L2
L1
GPA
Virtual machine
VMM
EPT MMU
Hardware EPT TLB
HPA
GPA
EPT pointer
GVA: virtual memory address of a process in guest OS
GPA: physical memory address in guest OS
HPA: physical memory address of the host machine

FIGURE 3.13
Memory virtualization using EPT by Intel (the EPT is also known as the shadow page table [68]).

I/O Virtualization

- I/O Virtualization managing the routing of I/O requests between virtual devices and the shared physical hardware
- managing the routing of I/O requests between virtual devices and the shared physical hardware
- Full device emulation emulates well-known, real-world devices All the functions of a device or bus infrastructure, such as device enumeration, identification, interrupts, and DMA, are replicated in software. This software is located in the VMM and acts as a virtual device
- Two-stage mapping process should be maintained by the guest OS and the VMM, respectively: virtual memory to physical memory and physical memory to machine memory.
- The VMM is responsible for mapping the guest physical memory to the actual machine memory.

Virtualization in Multi-Core Processors

- **Muti-core virtualization** has raised some new **challenges**
- **Two difficulties**: Application programs must be parallelized to use all cores fully, and software must explicitly
- Assign tasks to the cores, which is a very complex problem
- The **first challenge**, new programming models, languages, and libraries are needed to make parallel programming easier.
- The **second challenge** has spawned research involving scheduling algorithms and resource management policies
- **Dynamic heterogeneity** is emerging to mix the fat CPU core and thin GPU cores on the same chip

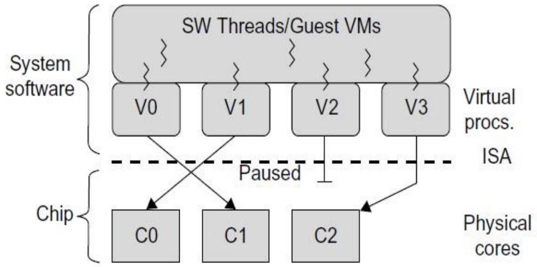


FIGURE 3.16

Multicore virtualization method that exposes four VCPUs to the software, when only three cores are actually present.

(*Courtesy of Wells, et al. [74]*)

- In **many-core chip multiprocessors (CMPs)**→
- Instead of supporting **time-sharing jobs** on one or a few cores, use the abundant cores →**space-sharing**, where single-threaded or multithreaded jobs are simultaneously assigned to separate groups of cores

Physical versus Virtual Clusters

- Virtual clusters are built with VMs installed at distributed servers from one or more physical clusters.
- Assign tasks to the cores, which is a very complex problem
- Fast deployment
- High-Performance Virtual Storage
- reduce duplicated blocks

Virtual Clusters

- **Four ways to manage a virtual cluster**.
- First, you can use a **guest-based manager**, by which the cluster manager resides on a guest system.
- The **host-based manager** supervises the guest systems and can restart the guest system on another physical machine
- Third way to manage a virtual cluster is to use an **independent cluster manager** on both the host and guest systems.
- Finally, use an **integrated cluster** on the guest and host systems.
- This means the manager must be designed to distinguish between virtualized resources and physical resources

Virtualization for data-center automation

- **Data-center automation** means that huge volumes of hardware,software, and database resources in these data centers can be allocated dynamically to millions of Internet users simultaneously, with guaranteed QoS and cost-effectiveness
- This **automation** process is triggered by the growth of virtualization products and cloud computing services.

- The latest virtualization development highlights high availability (HA), backup services, workload balancing, and further increases in client bases.

Server Consolidation in Data Centers

- heterogeneous workloads -chatty workloads and noninteractive workloads
- **Server consolidation** is an approach to improve the low utility ratio of hardware resources by reducing the number of physical servers

Virtual Storage Management

- **storage virtualization** has a different meaning in a system virtualization environment
- **system virtualization,** virtual storage includes the storage managed by VMMs
- and guest OSes data stored in this environment
- can be classified into two categories: VM images and application data.

Cloud OS for Virtualized Data Centers

- Data centers must be virtualized to serve as cloud providers
- **Eucalyptus for Virtual Networking of Private Cloud** :
- Eucalyptus is an **open source software system** intended mainly for supporting **Infrastructure as a Service (IaaS)** clouds
- The system primarily supports **virtual networking** and the management of **VMs**;
- **virtual storage is not supported.**
- Its purpose is to build **private clouds**
- three resource managers
  - Instance Manager
  - Group Manager
  - Cloud Manager

UNIT -3

Introduction to Cloud Computing

- **Cloud computing** allowing **access** to large amounts of **computing power** in a **fully virtualized** manner, by aggregating resources and offering a single system view
- **Cloud computing** has been coined as an umbrella term to describe a category of sophisticated on-demand computing services initially offered by commercial providers, such as Amazon, Google, and Microsoft.
- The main **principle** behind this model is offering computing, storage, and software **"as a service**
- **Cloud** is a parallel and distributed computing system consisting of a collection of **inter-connected and virtualised computers t**hat are dynamically **provisioned**
- The National Institute of Standards and Technology (NIST) characterizes cloud computing as ". . . a pay-per-use model for enabling available, convenient, on-demand network access to a shared pool of configurable computing resources

Common characteristics a cloud should have:

- pay-per-use (no ongoing commitment, utility prices);
- elastic capacity and the illusion of infinite resources;
- self-service interface; and
- resources that are abstracted or virtualised.
-

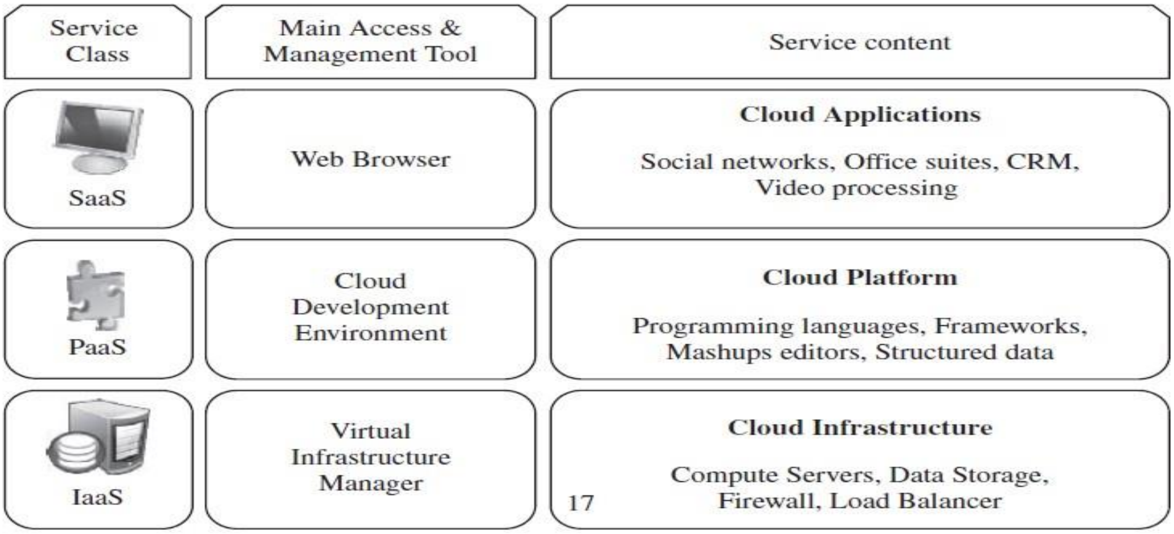
ROOTS OF CLOUD COMPUTING

- **Autonomic Computing**: seeks to improve systems by decreasing human involvement in their operation
- **Autonomic, or self-managing**, systems rely on monitoring probes and gauges (sensors), on an adaptation engine (autonomic manager) for computing optimizations based on monitoring data, and on effectors to carry out changes on the system.
- **IBM's Autonomic Computing** Initiative - four properties of autonomic systems: self-configuration, selfoptimization, self-healing, and self-protection.
- **Reference model** for autonomic control loops of **autonomic managers**, called MAPE-K (Monitor Analyze Plan Execute—Knowledge)

LAYERS AND TYPES OF CLOUDS

- **Cloud computing services** are divided into three classes:
- (1) Infrastructure as a Service,
- (2) Platform as a Service, and
- (3) Software as a Service.
- **Infrastructure as a Service** :A cloud infrastructure enables on-demand provisioning of servers running several choices of operating systems and a customized software stack.
    - **Amazon Web Services** mainly offers IaaS, which in the case of its EC2 service means offering VMs with a software stack that can be customized similar to how an ordinary physical server would be customized.
- **Platform as a Service**: A cloud platform offers an environment on which developers create and deploy applications
    - **Google AppEngine**, an example of Platform as a Service, offers a scalable environment for developing and hosting Web applications

Software as a Service: Traditional desktop applications such as word processing and spreadsheet can now be accessed as a service in the Web. This model of delivering applications, known as Software as a Service (SaaS)



Service Class	Main Access & Management Tool	Service content
SaaS	Web Browser	**Cloud Applications** Social networks, Office suites, CRM, Video processing
PaaS	Cloud Development Environment	**Cloud Platform** Programming languages, Frameworks, Mashups editors, Structured data
IaaS	Virtual Infrastructure Manager	**Cloud Infrastructure** Compute Servers, Data Storage, Firewall, Load Balancer

FIGURE 1.3. The cloud computing stack.

Deployment Models

- **Public cloud** as a “cloud made available in a pay-as-you-go manner to the general public” and
- **Private cloud** as “internal data center of a business or other organization, not made available to the general public.”
- A **community cloud** is “shared by several organizations and supports a specific community that has shared concerns
- A **hybrid cloud** takes shape when a private cloud is supplemented with computing capacity from public clouds.
- The approach of temporarily renting capacity to handle spikes in load is known as “cloud-bursting”

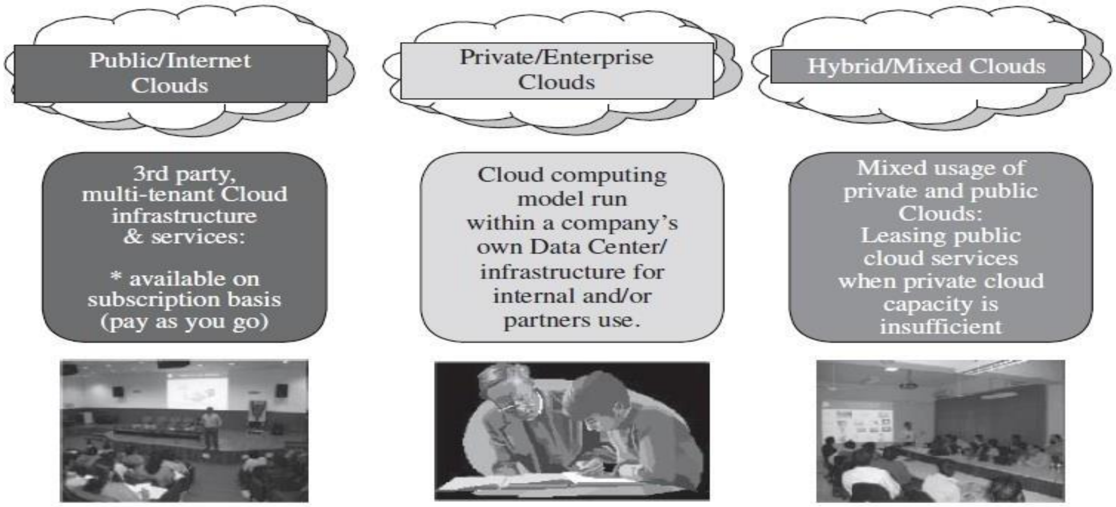

Public/Internet Clouds
Private/Enterprise Clouds
Hybrid/Mixed Clouds
3rd party, multi-tenant Cloud infrastructure & services:
* available on subscription basis (pay as you go)
Cloud computing model run within a company's own Data Center/ infrastructure for internal and/or partners use.
Mixed usage of private and public Clouds:
Leasing public cloud services when private cloud capacity is insufficient

FIGURE 1.4. Types of clouds based on deployment models.

- **Features of a cloud**
- are **essential** to enable services that truly represent the cloud computing model
- **Self-Service** : clouds must allow self-service access so that customers can request, customize, pay, and use services (expect on-demand, nearly instant access to resources) without intervention of human operators

- **Per-Usage Metering and Billing** : Services must be priced on a shortterm basis (e.g., by the hour), allowing users to release (and not pay for) resources as soon as they are not needed
- **Elasticity** : users expect clouds to rapidly provide resources in any quantity at any time. In particular, it is expected that the additional resources can be
- (a) provisioned, possibly automatically, when an application load increases and
- (b) released when load decreases (scale up and down)

Customization: resources rented from the cloud must be highly customizable.

- In the case of infrastructure services, customization means allowing users to deploy specialized virtual appliances and to be given privileged (root) access to the virtual servers

MIGRATING INTO THE CLOUD

- **Why Migrate?**
- There are economic and business reasons why an enterprise application can be migrated into the cloud, and there are also a number of technological reasons.
- Initiatives in adoption of cloud technologies in the enterprise,
- resulting in integration of enterprise applications running off the captive data centers with the new ones that have been developed on the cloud.

Migration can happen at one of the **five levels** of

- application,
- code,
- design,
- architecture,
- usage

The migration of an enterprise application is best captured by the following

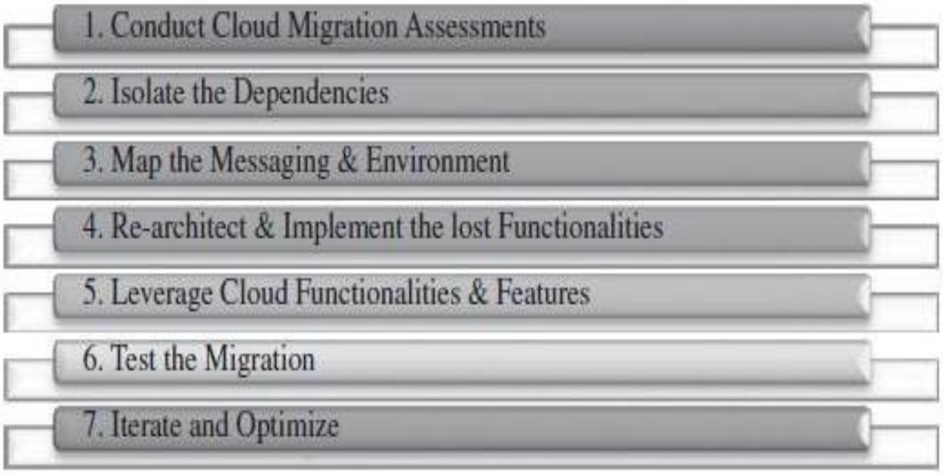
$$P \rightarrow P'_C + P'_l \rightarrow P'_{\mathrm{OFC}} + \mathrm{P'}_l$$

where

- P is the application before migration running in captive data center,
- P'_C is the application part after migration either into a (hybrid) cloud,
- P'_l is the part of application being run in the captive local data center, and

- P'_{OFC} is the application part optimized for cloud

SEVEN-STEP MODEL OF MIGRATION INTO A CLOUD



1. Conduct Cloud Migration Assessments
2. Isolate the Dependencies
3. Map the Messaging & Environment
4. Re-architect & Implement the lost Functionalities
5. Leverage Cloud Functionalities & Features
6. Test the Migration
7. Iterate and Optimize

FIGURE 2.4. The Seven-Step Model of Migration into the Cloud. (*Source:* Infosys Research.)

Iterative Step

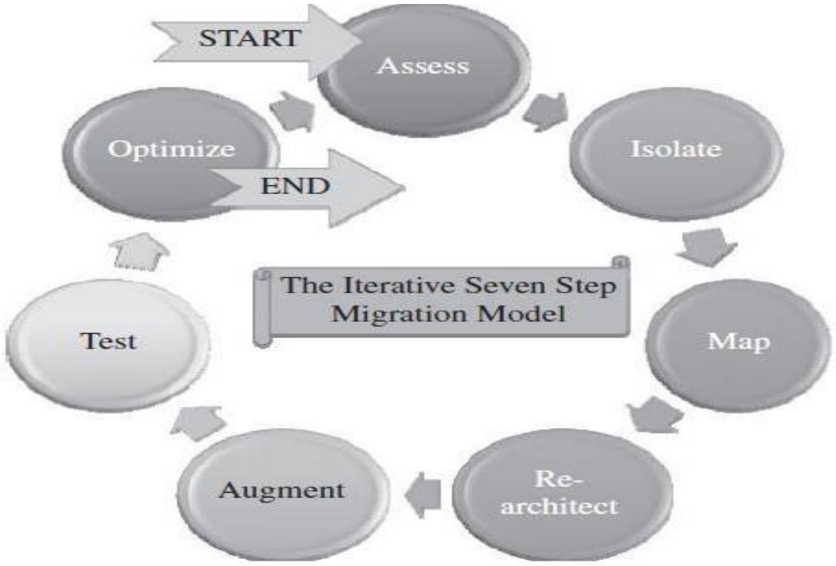


FIGURE 2.5. The iterative Seven-step Model of Migration into the Cloud. (*Source:* Infosys Research.)

Migration Risks and Mitigation

- The biggest challenge to any cloud migration project is how effectively the migration risks are identified and mitigated.
- Migration risks for migrating into the cloud fall under two broad categories:
- the general migration risks
- the security-related migration risks
- several issues identifying all possible production level deviants:
  - the business continuity and disaster recovery in the world of cloud computing service;
  - the compliance with standards and governance issues; the IP and licensing issues;
  - the quality of service (QoS) parameters as well as the corresponding SLAs committed to;
  - the ownership, transfer, and storage of data in the application;
  - the portability and interoperability issues which could help mitigate potential vendor lock-ins;

On the security front - as addressed in the guideline document published by the Cloud Security Alliance.

- Issues include
- security at various levels of the enterprise application as applicable on the cloud in addition to issues of trust and issues of privacy.
- There are several legal compliances that a migration strategy and implementation has to fulfill,
- including obtaining the right execution logs as well as retaining the rights to all audit trails at a detailed level

Challenges in the Cloud

- Security
- Costing model
- Charging model
- Service level agreement
- Cloud interoperability issue

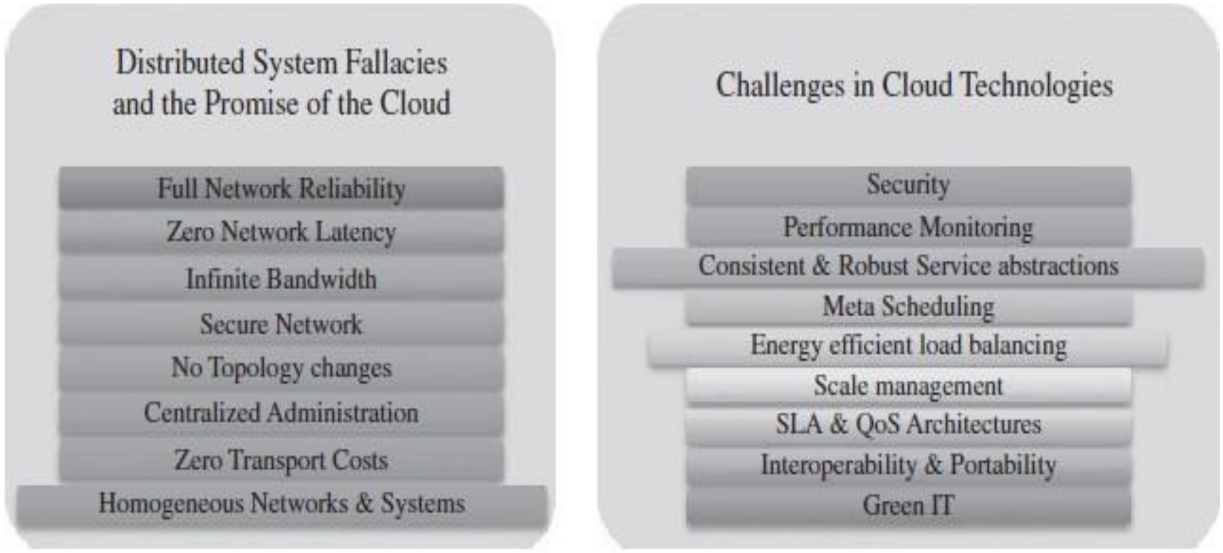

Distributed System Fallacies and the Promise of the Cloud
Full Network Reliability
Zero Network Latency
Infinite Bandwidth
Secure Network
No Topology changes
Centralized Administration
Zero Transport Costs
Homogeneous Networks & Systems
Challenges in Cloud Technologies
Security
Performance Monitoring
Consistent & Robust Service abstractions
Meta Scheduling
Energy efficient load balancing
Scale management
SLA & QoS Architectures
Interoperability & Portability
Green IT

FIGURE 2.3. 'Under the hood' challenges of the cloud computing services implementations.

– Use of cloud computing means dependence on others and that could possibly limit flexibility and innovation:
  - The others are likely become the bigger Internet companies like Google and IBM, who may monopolise the market.
  - Some argue that this use of supercomputers is a return to the time of mainframe computing that the PC was a reaction against.

– Security could prove to be a big issue:
  - It is still unclear how safe out-sourced data is and when using these services ownership of data is not always clear.

– There are also issues relating to policy and access:
  - If your data is stored abroad whose policy do you adhere to?
  - What happens if the remote server goes down?
  - How will you then access files?
  - There have been cases of users being locked out of accounts and losing access to data.

UNIT -4

INFRASTRUCTURE AS A SERVICE PROVIDERS (IAAS)

- Public Infrastructure as a Service providers commonly offer virtual servers containing one or more CPUs, OS, software stack, storage space and communication facilities
- The most relevant features are:
- (i) geographic distribution of data centers;
- (ii) variety of user interfaces and APIs to access the system;
- (iii) specialized components and services that aid particular applications (e.g., loadbalancers,)
- (iv) choice of virtualization platform and operating systems;
- (v) different billing methods and period

Geographic Presence: To improve availability and responsiveness, a provider of worldwide services would typically build several data centers distributed around the world.

- Availability zones are "distinct locations that are engineered to be insulated from failures in other availability zones and provide inexpensive, low-latency network connectivity to other availability zones in the same region."

- **User Interfaces and Access to Servers**.
- Ideally, a public IaaS provider must provide multiple access means to its cloud, thus catering for various usersand their preferences.
- Different types of user interfaces (UI) provide different levels of abstraction, the most common being
  - graphical user interfaces (GUI),
  - command-line tools (CLI), and
  - Web service (WS) APIs.

Advance Reservation of Capacity.

- **Advance reservations** allow users to request for an IaaS provider to reserve resources for a specific time frame in the future, thus ensuring that cloud resources will be available at that time
    - Amazon Reserved Instances is a form of advance reservation of capacity, allowing users to pay a fixed amount of money in advance to guarantee resource availability
- **Automatic Scaling and Load Balancing.**
- As mentioned earlier in this chapter, elasticity is a key characteristic of the cloud computing model.
- Applications often need to scale up and down to meet varying load conditions.
- **Service-Level Agreement**.
- Service-level agreements (SLAs) are offered by IaaS providers to express their commitment to delivery of a certain QoS.
- To customers it serves as a warranty.
    - Amazon EC2 states that "if the annual uptime Percentage for a customer drops below 99.95% for the service year, that customer is eligible to receive a service credit equal to 10% of their bill.3"

- **Hypervisor and Operating System Choice**:
    - Traditionally, IaaS offerings have been based on heavily customized open-source Xen deployments.
    - IaaS providers needed expertise in Linux, networking, virtualization, metering, resource management, and many other low-level aspects to successfully deploy and maintain their cloud offerings.

Public Cloud and Infrastructure Services

- **Public cloud or external cloud**
- describes cloud computing - resources are **dynamically** provisioned via publicly accessible Web applications/Web services (SOAP or RESTful interfaces) from an off-site third-party provider,
- who shares resources and bills on a fine-grained utility computing basis,

- the user pays only for the capacity of the provisioned resources at a particular time

Amazon Elastic Compute Cloud (EC2) is an IaaS service that provides elastic compute capacity in the cloud

Private Cloud and Infrastructure Services

- A **private cloud** aims at providing **public cloud functionality**, but **on private resources**, while maintaining control over an organization’s data and resources to meet security and governance’s requirements in an organization.
- Private clouds exhibit the following characteristics:
  - Allow service provisioning and compute capability for an organization’s users in a self-service manner.
  - Automate and provide well-managed virtualized environments.
  - Optimize computing resources, and servers’ utilization.
  - Support specific workloads.
- Examples are Eucalyptus and OpenNebula

“Hybrid cloud”

- in which a combination of private/internal and external cloud resources exist together by enabling outsourcing of noncritical services and functions in public cloud and keeping the critical ones internal
- Release resources from a public cloud and to handle sudden demand usage, which is called “cloud bursting

Cloud and Virtualization Standardization Efforts

- Standardization is important to ensure interoperability between
- virtualization mangement vendors,
- the virtual machines produced by each one of them,
- and cloud computing
- Distributed Management Task Force(DMTF)
- initiated the VMAN (Virtualization Management Initiative),
- delivers broadly supported interoperability and portability standards for managing the virtual computing lifecycle.

OVF (Open Virtualization Format

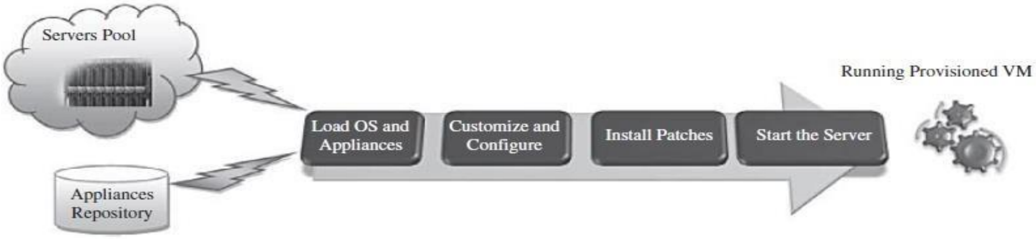
- VMAN’s OVF (Open Virtualization Format) in a collaboration between industry key players: Dell, HP, IBM, Microsoft, XenSource, and Vmware.
- OVF specification provides a common format to package and securely distribute virtual appliances across multiple virtualization platforms.
- VMAN profiles define a consistent way of managing a heterogeneous virtualized environment
- Standardization effort has been initiated by Open Grid Forum (OGF) through organizing an official new working group to deliver a standard API for cloud IaaS, the Open Cloud Computing Interface Working Group (OCCIWG)

VIRTUAL MACHINES PROVISIONING

- Typical life cycle of VM and its major possible states of operation, which make the management and automation of VMs in virtual and cloud environments easier

Process:

- Steps to Provision VM. Here, we describe the common and normal steps of provisioning a virtual server:
- Firstly, you need to select a server from a pool of available servers (physical servers with enough capacity) along with the appropriate OS template you need to provision the virtual machine.
- Secondly, you need to load the appropriate software (operating system you selected in the previous step, device drivers, middleware, and the needed applications for the service required).



-

FIGURE 5.4. Virtual machine provision process.

- Thirdly, you need to customize and configure the machine (e.g., IP address, Gateway) to configure an associated network and storage resources.
- Finally, the virtual server is ready to start with its newly loaded software

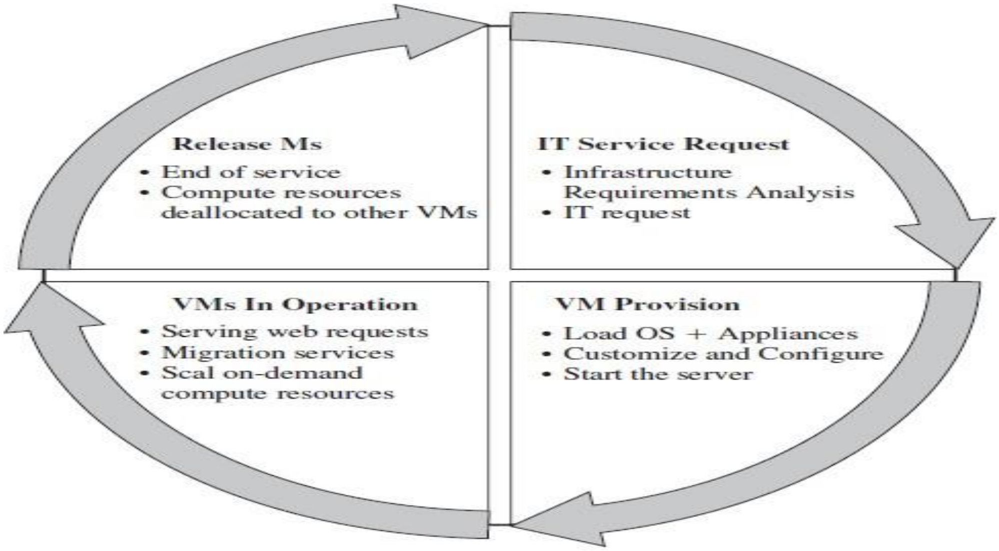


FIGURE 5.3. Virtual machine life cycle.

VIRTUAL MACHINE MIGRATION SERVICES

Migration service,

- in the context of virtual machines, is the process of moving a virtual machine from one host server or storage location to another
- There are different techniques of VM migration,
- hot/life migration,
- cold/regular migration, and
- live storage migration of a virtual machine

VM Migration, SLA and On-Demand Computing:

- virtual machines’ migration plays an important role in data centers
- once it has been detected that a particular VM is consuming more than its fair share of resources at the expense of other VMs on the same host,

- it will be eligible, for this machine, to either be moved to another underutilized host or assign more resources for it
- There should be an integration between virtualization's management tools (with its migrations and performance's monitoring capabilities), and SLA's management tools to achieve balance in resources by migrating and monitoring the workloads, and accordingly, meeting the SLA

Migration of Virtual Machines to Alternate Platforms

- Advantages of having facility in data center's technologies is
- to have the **ability to migrate virtual machines** from one platform to another
- For example, the **VMware converter that handles migrations** between ESX hosts;
- the VMware server; and the VMware workstation.
- The VMware converter can also import from other virtualization platforms, such as Microsoft virtual server machines

Deployment Scenario:

- ConVirt deployment consists of at least one ConVirt workstation,
- whereConVirt is installed and ran, which provides the main console for managing the VM life cycle, managing images, provisioning new VMs, monitoring machine resources, and so on.
- There are two essential deployment scenarios for ConVirt:
- A, **basic configuration** in which the Xen or KVM virtualization platform is on the local machine, where ConVirt is already installed; B,
- An **advanced configuration** in which the Xen or KVM is on one or more remote servers.

Installation. The installation process involves the following:

- Installing ConVirt on at least one computer.
- Preparing each managed server to be managed by ConVirt.
- We have two managing servers with the following Ips
- (managed server 1, IP:172.16.2.22; and
- managed server 2, IP:172.16.2.25) as shown in the deployment diagram

Environment, Software, and Hardware. ConVirt 1.1, Linux Ubuntu 8.10, three machines, Dell core 2 due processor, 4G RAM.

- Adding Managed Servers and Provisioning VM.
- Once the installation is done and you are ready to manage your virtual infrastructure, then you can start the ConVirt management console :
- Select any of servers' pools existing (QA Lab in our scenario) and on its context menu, select "Add Server."
- You will be faced with a message asking about the virtualization platform you want to manage (Xen or KVM), as shown in Figure
- Choose KVM, and then enter the managed server information and credentials (IP, username, and password) as shown in Figure
- Once the server is synchronized and authenticated with the management console, it will appear in the left pane/of the ConVirt,

- **Live Migration Effect** on a Running Web Server.
- Clark et al. did evaluate the above migration on an Apache 1.3 Web server; this served static content at a high rate, as illustrated in Figure 5.6.
- The throughput is achieved when continuously serving a single 512-kB file to a set of one hundred concurrent clients.
- This simple example demonstrates that a highly loaded server can be migrated with both controlled impact on live services and a short downtime

- **VMware Vmotion.**
- This allows users to
- (a) automatically optimize and allocate an entire pool of resources for maximum hardware utilization, flexibility, and availability and
- (b) perform hardware's maintenance without scheduled downtime along with migrating virtual machines away from failing or underperforming servers

- **Citrix XenServerXenMotion**.

- This is a nice feature of the Citrix XenServer product, inherited from the Xen live migrate utility, which provides the IT administrator with the facility to move a running VM from one XenServer to another in the same pool without interrupting the service

Regular/Cold Migration.

Cold migration is the migration of a powered-off virtual machine.

- Main **differences between live migration and cold migration** are that
- 1) **live migration needs a shared storage** for virtual machines in the server's pool, but cold migration does not;
- 2) live migration for a virtual machine between two hosts, there would be **certain CPU compatibility checks to be applied**; while in cold migration this checks do not apply
- The cold migration process (VMware) can be summarized as follows:
  - The **configuration files**, including the NVRAM file (BIOS settings), log files, as well as the disks of the virtual machine, are **moved** from the source host to the destination host's associated storage area.
  - The virtual machine is **registered** with the new host.
  - After the migration is completed, the **old version** of the virtual machine is **deleted** from the source host.

Live Storage Migration of Virtual Machine.

- This kind of migration constitutes moving the virtual disks or configuration file of a running virtual machine to a new data store without any interruption in the availability of the virtual machine's service

Aneka

- Manjrasoft Aneka is a .NET-based platform and framework designed for building and deploying distributed applications on clouds.
- It provides a set of APIs for transparently exploiting distributed resources and expressing the business logic of applications by using the preferred programming abstractions.
- Aneka also provides support for deploying and managing clouds.

- By using its Management Studio and a set ofWeb interfaces, it is possible to set up either public or private clouds, monitor their status, update their configuration, and perform the basic management operations.

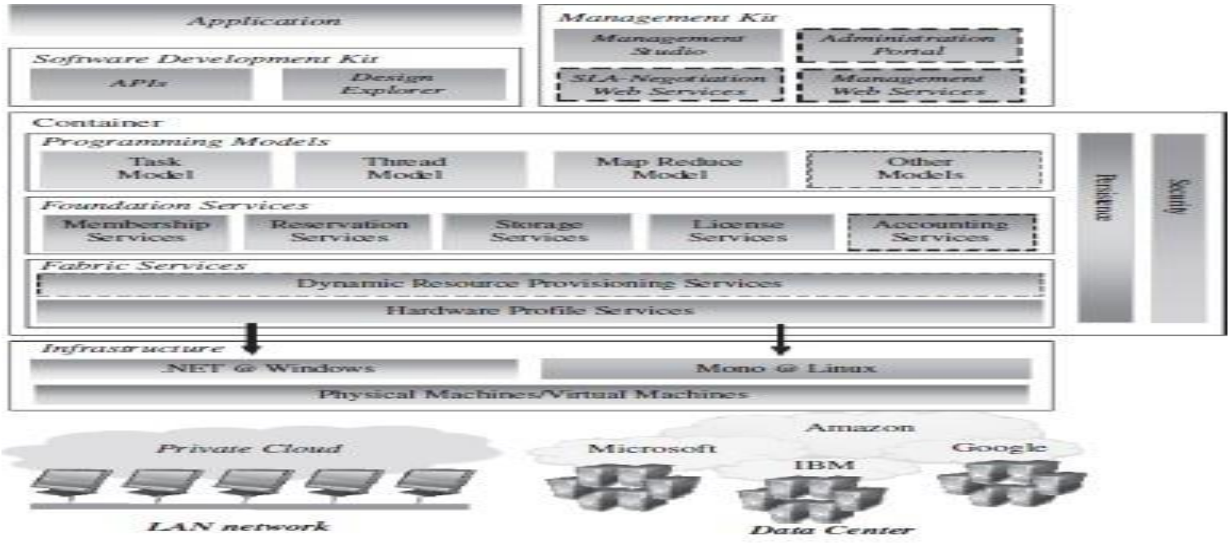


FIGURE 5.22. Manjras oft Aneka layered architecture [10].

UNIT-5

SAAS

- SaaS is a model of software deployment where an application is hosted as a service provided to customers across the Internet.
- Saas alleviates the burden of software maintenance/support but users relinquish control over software versions and requirements

SaaS Maturity Model

Level 1: Ad-Hoc/Custom – One Instance per customer

Level 2: Configurable per customer

Level 3: configurable & Multi-Tenant-Efficient

Level 4: Scalable, Configurable & Multi-Tenant-Efficient

SaaS INTEGRATION PRODUCTS AND PLATFORMS

- Cloud-centric integration solutions are being developed and demonstrated for showcasing their capabilities for integrating enterprise and cloud applications.
- Composition and collaboration will become critical and crucial for the mass adoption of clouds

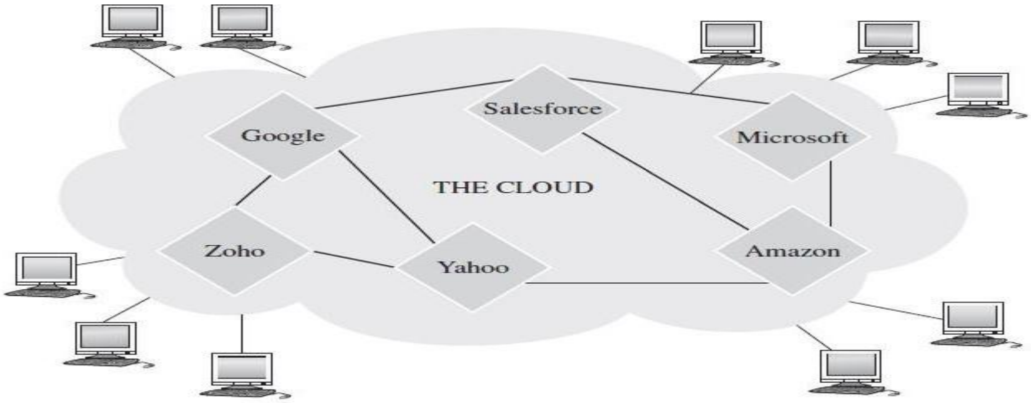


FIGURE 3.4. The Smooth and Spontaneous Cloud Interaction via Open Clouds.

Jitterbit:

- Jitterbit is a fully graphical integration solution that provides users a versatile platform
- suite of productivity tools to reduce the integration efforts sharply.
- Jitterbit can be used standalone or with existing EAI infrastructures
- Help us quickly design, implement, test, deploy, and manage the integration projects

- Two major components :
- Jitterbit Integration Environment
- An intuitive point-and-click graphical UI that enables to quickly configure, test, deploy and manage integration projects on the Jitterbit server.
- Jitterbit Integration Server
- A powerful and scalable run-time engine that processes all the integration operations, fully configurable and manageable from the Jitterbit application.

Linkage with On premise and on demand Applications

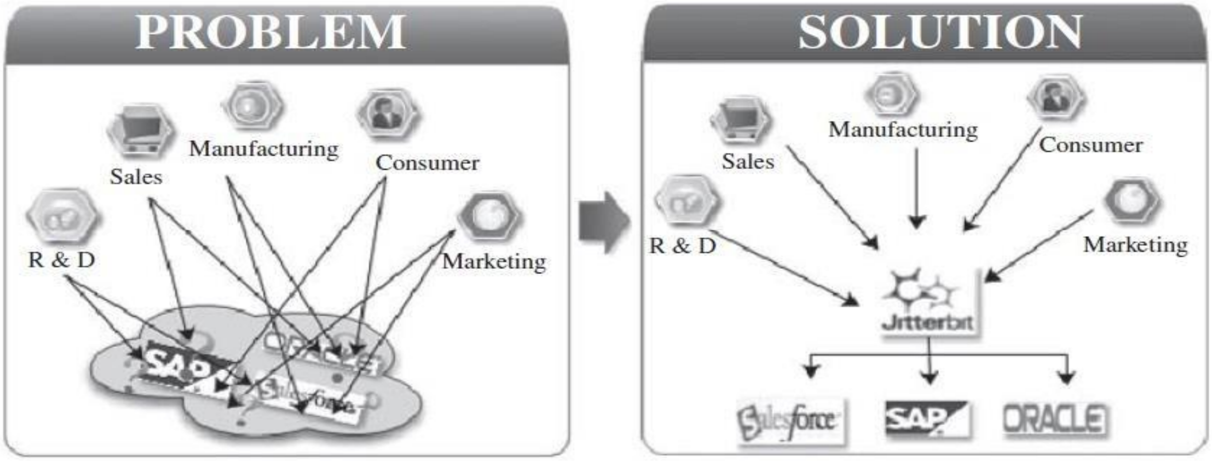


FIGURE 3.5. Linkage of On-Premise with Online and On-Demand Applications.

Google APP Engine

- The app engine is a Cloud-based platform, is quite comprehensive and combines infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS).
- The app engine supports the delivery, testing and development of software on demand in a Cloud computing environment that supports millions of users and is highly scalable.
- The company extends its platform and infrastructure to the Cloud through its app engine. It presents the platform to those who want to develop SaaS solutions at competitive costs.

Google is a leader in web-based applications,

so it’s not surprising that the company also offers cloud development services.

- These services come in the form of the Google App Engine, which enables developers to build their own web applications utilizing the same infrastructure that powers Google's powerful applications.
- The Google App Engine provides a fully integrated application environment. Using Google's development tools and computing cloud, App Engine applications are easy to build, easy to maintain, and easy to scale. All you have to do

Features of App Engine

- These are covered by the depreciation policy and the service-level agreement of the app engine. Any changes made to such a feature are backward-compatible and implementation of such a feature is usually stable. These include data storage, retrieval, and search; communications; process management; computation; app configuration and management.

- Data storage, retrieval, and search include features such as HRD migration tool, Google Cloud SQL, logs, datastore, dedicated Memcache, blobstore, Memcache and search.
- Communications include features such as XMPP. channel, URL fetch, mail, and Google Cloud Endpoints.
- Process management includes features like scheduled tasks and task queue
- Computation includes images.
- App management and configuration cover app identity, users, capabilities, traffic splitting, modules, SSL for custom domains, modules, remote access, and multitenancy

Centralizing email Communications

- The key here is to enable anywhere/anytime access to email.
- Precloud computing, your email access was via a single computer, which also stored all your email messages. For this purpose, you probably used a program like Microsoft Outlook or Outlook Express, installed on your home computer.
- To check your home email from work, it took a bit of juggling and perhaps the use of your ISP's email access web page. That web page was never in sync with the messages

on your home PC, of course, which is just the start of the problems with trying to communicate in this fashion.

- A better approach is to use a web-based email service, such as Google's Gmail (mail.google.com), Microsoft's Windows Live Hotmail (mail.live.com), or Yahoo! Mail (mail.yahoo.com). These services place your email inbox in the cloud; you can access it from any computer connected to the Internet.

Collaborating via Web-Based Communication Tools

GMAIL

- Gmail offers a few unique features that set it apart from the web-based email crowd.
- First, Gmail doesn't use folders. With Gmail you can't organize your mail into folders, as you can with the other services.
- Instead, Gmail pushes the search paradigm as the way to find the messages you want—not a surprise, given Google's search-centric business model.
- Gmail does, however, let you "tag" each message with one or more labels. This has the effect of creating virtual folders, as you can search and sort your messages by any of their labels.
- In addition, Gmail groups together related email messages in what Google calls conversations

Yahoo! Mail Yahoo! Mail (mail.yahoo.com)

- is another web mail service, provided by the popular Yahoo! search site.
- The basic Yahoo! Mail is free and can be accessed from any PC, using any web browser.
- Yahoo! also offers a paid service called Yahoo! Mail Plus that lets you send larger messages and offers offline access to your messages via POP email clients

Web Mail Services

- AOL Mail (mail.aol.com)
- BigString (www.bigstring.com) E
- xcite Mail (mail.excite.com)
- FlashMail (www.flashmail.com)

- GMX Mail (www.gmx.com)
- Inbox.com (www.inbox.com)
- Lycos Mail (mail.lycos.com)
- Mail.com (www.mail.com)
- Zoho Mail (zoho.mail.com)

Data Security

- Information in a cloud environment has much more dynamism and fluidity than information that is static on a desktop or in a network folder
- Nature of cloud computing dictates that data are fluid objects, accessible froma multitude of nodes and geographic locations and, as such, must have a datasecurity methodology that takes this into account while ensuring that this fluidity is not compromised
- The idea of content-centric or information-centric protection, being an inherent part of a data object is a development out of the idea of the "de-perimerization" of the enterprise.
- This idea was put forward by a group of Chief Information Officers (CIOs) who formed an organization called the Jericho Forum

CLOUD COMPUTING AND IDENTITY

Digital identity

- holds the key to flexible data security within a cloud Environment
- A digital identity represents who we are and how we interact with others on-line.
- **Access, identity, and risk** are three variables that can become inherently connected when applied to the security of data, because access and risk are directly proportional: As access increases, so then risk to the security of the data increases.
- Access controlled by identifying the actor attempting the access is the most logical manner of performing this operation.
- Ultimately, digital identity holds the key to securing data, if that digital identity can be programmatically linked to security policies controlling the post-access usage of data.

Identity, Reputation, and Trust

- Reputation is a real-world commodity; that is a basic requirement of human-to-human relationships
- Our basic societal communication structure is built upon the idea of reputation and trust.
- Reputation and its counter value, trust, is easily transferable to a digital realm:
    - eBay, for example, having partly built a successful business model on the strength of a ratings system, builds up the reputation of its buyers and sellers through successful (or unsuccessful) transactions.
- These types of reputation systems can be extremely useful when used with a digital identity.
- They can be used to associate varying levels of trust with that identity, which in turn can be used to define the level (granular variations) of security policy applied to data resources that the individual wishes to access

User-Centric Identity:

- Digital identities are a mechanism for identifying an individual, particularly within a cloud environment ; identity ownership being placed upon the individual is known as user-centric identity
- It allows users to consent and control how their identity (and the individual identifiers making up the identity, the claims) is used.
- This reversal of ownership away from centrally managed identity platforms (enterprise-centric) has many advantages.
- This includes the potential to improve the privacy aspects of a digital identity, by giving an individual the ability to apply permission policies based on their identity and to control which aspects of that identity are divulged
- An identity may be controllable by the end user, to the extent that the user can then decide what information is given to the party relying on the identity

Information Card:

- Information cards permit a user to present to a Web site or other service (relying party) one or more claims, in the form of a software token, which may be used to uniquely identify that user.

- They can be used in place of user name/ passwords, digital certificates, and other identification systems, when user identity needs to be established to control access to a Web site or other resource, or to permit digital signing

Information cards are part of an identity meta-system consisting of:

- 1. **Identity providers (IdP**), who provision and manage information cards,with specific claims, to users.
- 2. **Users** who own and utilize the cards to gain access to Web sites and other resources that support information cards.
- **An identity selector/service**, which is a piece of software on the user's desktop or in the cloud that allows a user to select and manage their cards.
- 4. **Relying parties.** These are the applications, services, and so on, that can use an information card to authenticate a person and to then authorize an action such as logging onto a Web site, accessing a document, signing content, and so on

Each information card is associated with a set of claims which can be used toidentify the user. These claims include identifiers such as name, email address,post code

<u>Using Information Cards to Protect Data</u>

- Information cards are built around a set of open standards devised by a consortium that includes Microsoft, IBM, Novell, and so on.
- The original remit of the cards was to create a type of single sign on system for the Internet, to help users to move away from the need to remember multiple passwords.
- However, the information card system can be used in many more ways.
- Because an information card is a type of digital identity, it can be used in the same way that other digital identities can be used.

For example, an information card can be used to digitally sign data and content and to control access to data and content. One of the more sophisticated uses of an information card is the advantage given to the cards by way of the claims system.

Cloud Computing and Data Security Risk

- Cloud computing is a development that is meant to allow more open accessibility and easier and improved data sharing.

- Data are uploaded into a cloud and stored in a data center, for access by users from that data center; or in a more fully cloud-based model, the data themselves are created in the cloud and stored and accessed from the cloud (again via a data center).
- The most obvious risk in this scenario is that associated with the storage of that data. A user uploading or creating cloud-based data include those data that are stored and maintained by a third-party cloud provider such as Google, Amazon, Microsoft, and so on.

This action has several risks associated with it:

- Firstly, it is necessary to protect the data during upload into the data center to ensure that the data do not get hijacked on the way into the database.
- Secondly, it is necessary to the stores the data in the data center to ensure that they are encrypted at all times.
- Thirdly, and perhaps less obvious, the access to those data need to be controlled; this control should also be applied to the hosting company, including the administrators of the data center.
- In addition, an area often forgotten in the application of security to a data resource is the protection of that resource during its use

Data security risks are compounded by the open nature of cloud computing.

- Access control becomes a much more fundamental issue in cloud-based systems because of the accessibility of the data
- Information-centric access control (as opposed to access control lists) can help to balance improved accessibility with risk, by associating access rules with different data objects within an open and accessible platform, without losing the Inherent usability of that platform
- A further area of risk associated not only with cloud computing, but also with traditional network computing, is the use of content after access.
- The risk is potentially higher in a cloud network, for the simple reason that the information is outside of your corporate walls

Data-centric mashupsare those

- that are used to perform business processes around data creation and dissemination—by their very nature, can be used to hijack data, leaking sensitive information and/or affecting integrity of that data
- Cloud computing, more than any other form of digital communication technology, has created a need to ensure that protection is applied at the inception of the information, in a content centric manner, ensuring that a security policy becomes an integral part of that data throughout its life cycle.

Encryption

- is a vital component of the protection policy, but further controls over the access of that data and on the use of the data must be met.
- In the case of mashups, the controlling of access to data resources, can help toalleviate the security concerns by ensuring that mashup access is authenticated.
- Linking security policies, as applied to the use of content, to the access control method offer a way of continuing protection of data, post access and throughout the life cycle; this type of data security philosophy must be incorporated into the use of cloud computing to alleviate security risks.