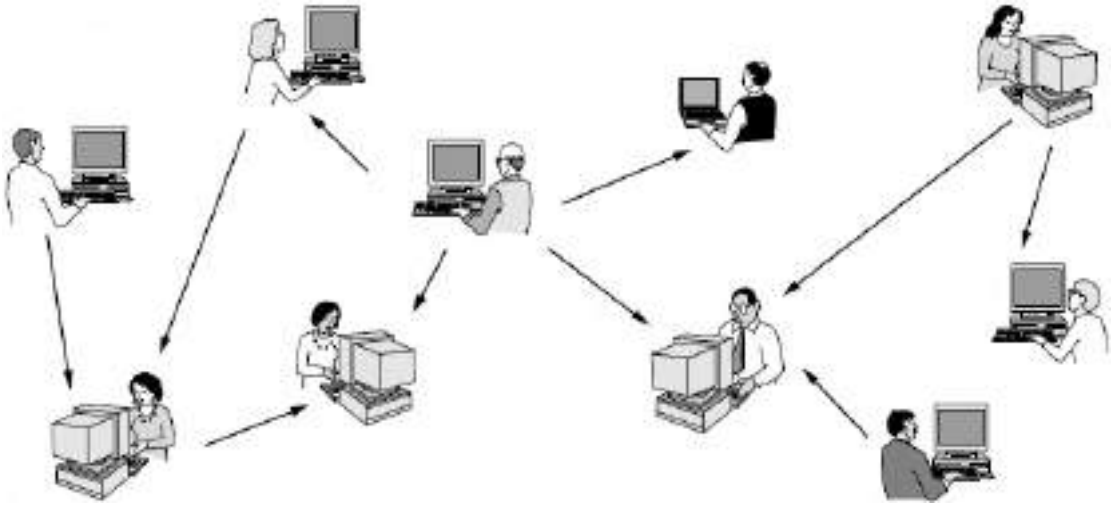
Computer Networks

NETWORK DEFINITION

- **A Computer Network** is a collection of devices connected together to provide certain services to the users.
- 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.



What is the difference between a distributed system and a computer network?

- **Distributed systems** are computing nodes that communicate with each other on purpose of processing data or running applications.

  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 (WWW). It runs on top of the Internet and presents a model in which everything looks like a document (Web page).

- **Computer networks** are nodes that communicate of purpose of exchanging data and deliver it from node to other or multiple nodes.

USES OF COMPUTER NETWORKS

Networks are an important part of everyday activities: Business, Home, Government, Education. They provide numerous advantages:

1- Share Resources from one computer to another:
   - Create files and store them in one computer, access those files from the other computers.
   - Connect a printer, scanner, or a fax machine to one computer within the network and let other computers of the network use the machines available over network.

2- Exchange of information by means of e-Mails and FTP.

3- Information sharing by using Web or Internet.

4- Interaction with other users using dynamic web pages.

5- IP phones.

6- Voice over IP (VOIP).

7- Video conferences.

8- Parallel computing.

9- Instant messaging.

10- VPNs (Virtual Private Networks) to join the individual networks at different sites into one extended network.

11- E-commerce (Electronic Commerce).

12- Social network applications.

NETWORK COMPONENTS

Network has three main components:

1- **End Devices** (hosts, servers).

2- **Network devices** (repeater, hub, bridge, switch, router, NIC, modem)
   - Devices that interconnect different computer together

3- **Connectivity** (cables, wireless)
   - Media that physically connect the computers and network devices.

Computer Networks

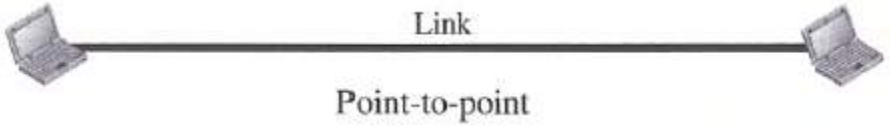
NETWORK HARDWARE

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. For visualization purposes, it is simplest to imagine any link as a line drawn between two points. For communication to occur, two devices must be connected in some way to the same link at the same time. There are two possible types of connections: point-to-point and multipoint.

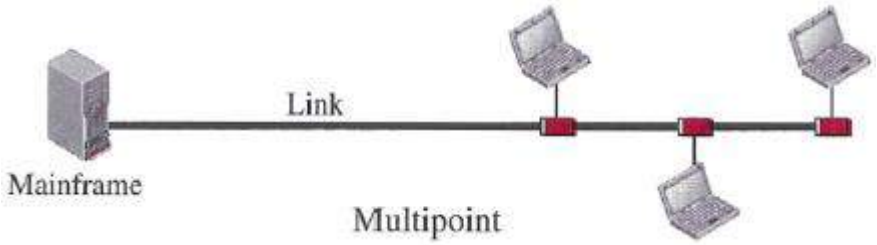
1- Point-to-Point

A point-to-point connection provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices. Most point-to-point connections use an actual length of wire or cable to connect the two ends, but other options, such as microwave or satellite links, are also possible (see the figure below). When we change television channels by infrared remote control, we are establishing a point-to-point connection between the remote control and the television's control system.



2- Multipoint

A multipoint (also called multidrop) connection is one in which more than two specific devices share a single link (see the figure below). In a multipoint environment, the capacity of the channel is shared, either spatially or temporally. If several devices can use the link simultaneously, it is a spatially shared connection. If users must take turns, it is a timeshared connection.



PHYSICAL TOPOLOGY

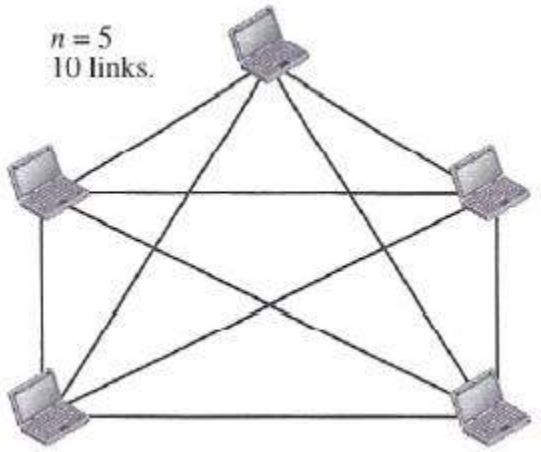
The term physical topology refers to the way in which a network is laid out physically. Two or more devices connect to a link; two or more links form a topology. The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to one another. There are four basic topologies possible: mesh, star, bus, and ring.

Computer Networks

1- Mesh Topology

In a mesh topology, every device has a dedicated point-to-point link to every other device. The term dedicated means that the link carries traffic only between the two devices it connects.

To find the number of physical links in a fully connected mesh network with n nodes, we first consider that each node must be connected to every other node. Node 1 must be connected to n - 1 nodes, node 2 must be connected to n - 1 nodes, and finally node n must be connected to n - 1 nodes. We need n (n - 1) physical links. However, if each physical link allows communication in both directions (duplex mode), we can divide the number of links by 2. In other words, we can say that in a mesh topology, we need **n (n - 1) / 2** duplex-mode links. To accommodate that many links, every device on the network must have n - 1 input/output (I/O) ports (see the figure below) to be connected to the other n - 1 stations.



Advantages of Mesh Topology

1- Each connection can carry its own data load.

2- It is robust.

3- Fault is diagnosed easily.

4- Provides security and privacy.

Disadvantages of Mesh Topology

1- Installation and configuration is difficult.

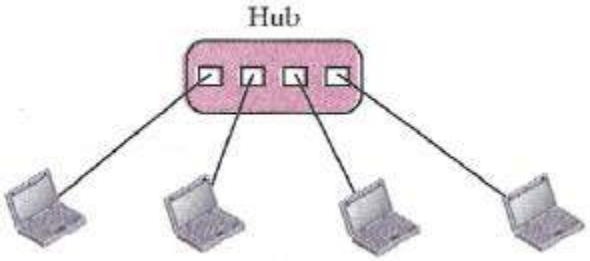
2- Cabling cost is more.

3- Bulk wiring is required.

One practical example of a mesh topology is the connection of **telephone regional offices** in which each regional office needs to be connected to every other regional office.

2- Star Topology

In a star topology, each device has a dedicated point-to-point link only to a central controller, usually called a hub. The devices are not directly linked to one another. Unlike a mesh topology, a star topology does not allow direct traffic between devices. The controller acts as an exchange: If one device wants to send data to another, it sends the data to the controller, which then relays the data to the other connected device (see The figure below).



Advantages of Star Topology

1- Easy to troubleshoot.
2- Easy to setup and modify.
3- Only that node is affected which has failed, rest of the nodes can work smoothly.

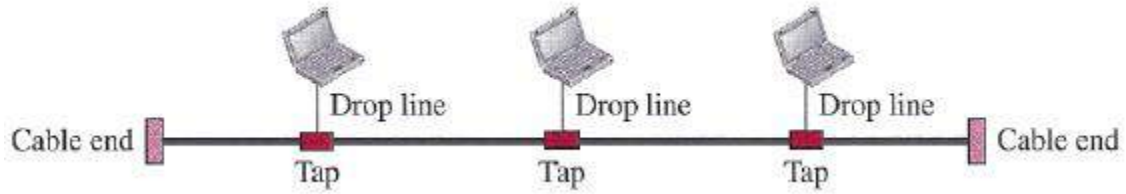
Disadvantages of Star Topology

1- Cost of installation is high.
2- If the hub fails, then the whole network is stopped because all the nodes depend on the hub.
3- Performance is based on the hub that is it depends on its capacity

The star topology is used in **local-area networks (LANs)**.

3- Bus Topology

The preceding examples all describe point-to-point connections. A bus topology, on the other hand, is multipoint. One long cable acts as a backbone to link all the devices in a network (see the figure below).



Nodes are connected to the bus cable by drop lines and taps.

- A drop line is a connection running between the device and the main cable.
- A tap is a connector that either splices into the main cable or punctures the sheathing of a cable to create a contact with the metallic core.

As a signal travels along the backbone, some of its energy is transformed into heat. Therefore, it becomes weaker and weaker as it travels farther and farther. For this reason, there is a limit on the number of taps a bus can support and on the distance between those taps.

Advantages of Bus Topology

1- It is cost effective.
2- Cable required is least compared to other network topology.
3- Used in small networks.
4- It is easy to understand.
5- Easy to expand joining two cables together.

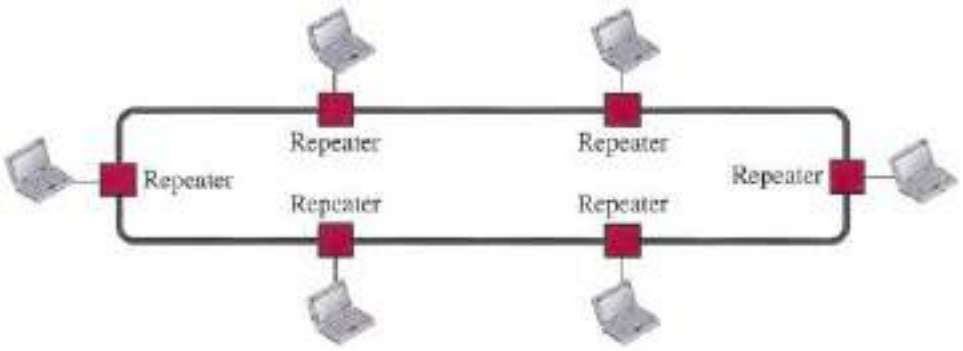
Disadvantages of Bus Topology

1- Cables fails then whole network fails.
2- If network traffic is heavy or nodes are more the performance of the network decreases.
3- Cable has a limited length.
4- It is slower than the ring topology.

Bus topology was the one of the first topologies used in the design of early local area networks. Traditional Ethernet LANs can use a bus topology.

4- Ring Topology

In a ring topology, each device has a dedicated point-to-point connection with only the two devices on either side of it. A signal is passed along the ring in one direction, from device to device, until it reaches its destination. Each device in the ring incorporates a repeater. When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along (see the figure below).



Generally, in a ring a signal is circulating at all times. If one device does not receive a signal within a specified period, it can issue an alarm. The alarm alerts the network operator to the problem and its location.

Advantages of Ring Topology

1- Transmitting network is not affected by high traffic or by adding more nodes.
2- Cheap to install and expand

Disadvantages of Ring Topology

1- Troubleshooting is difficult in ring topology.
2- Adding or deleting the computers disturbs the network activity.
3- Failure of one computer disturbs the whole network

Ring topology was prevalent when IBM introduced its local-area network, Token Ring. Today, the need for higher-speed LANs has made this topology less popular.

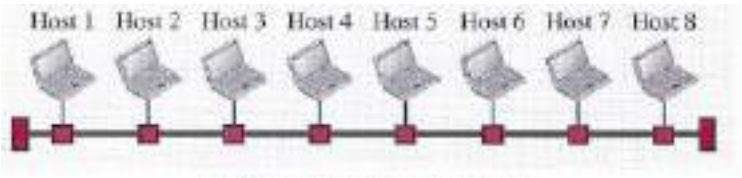
NETWORK TYPES

The criteria of distinguishing one type of network from another is difficult and sometimes confusing. We use a few criteria such as **size, geographical coverage, and ownership** to make this distinction.

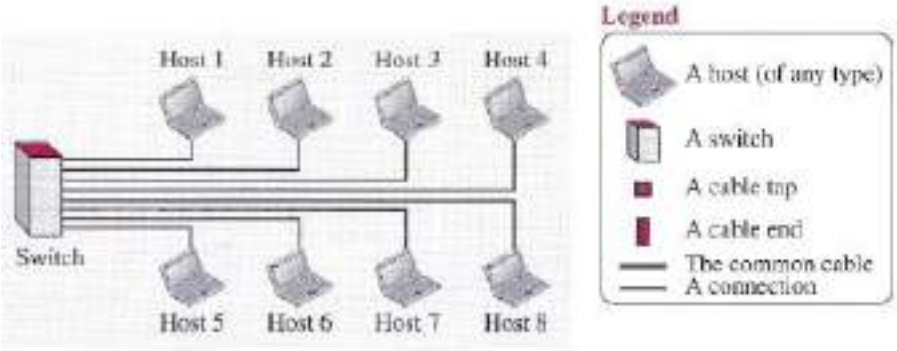
1- Local Area Network

- A local area network (LAN) is usually privately owned and connects some hosts in a single office, building, or campus.
- Depending on the needs of an organization, a LAN can be as simple as two PCs and a printer in someone's home office, or it can extend throughout a company and include audio and video devices.
- Each host in a LAN has an identifier, an address, that uniquely defines the host in the LAN. A packet sent by a host to another host carries both the source host's and the destination host's addresses.
- In the past, all hosts in a network were connected through a common cable, which meant that a packet sent from one host to another was received by all hosts. The intended recipient kept the packet; the others dropped the packet. Today, most LANs use a smart connecting switch, which is able to recognize the destination address of the packet and guide the packet to its destination without sending it to all other hosts.
- The switch alleviates the traffic in the LAN and allows more than one pair to communicate with each other at the same time if there is no common source and destination among them.
- Note that the above definition of a LAN does not define the minimum or maximum number of hosts in a LAN. The figure below shows a LAN using either a common cable or a switch.

Computer Networks



LAN with a common cable (past)



LAN with a switch (today)

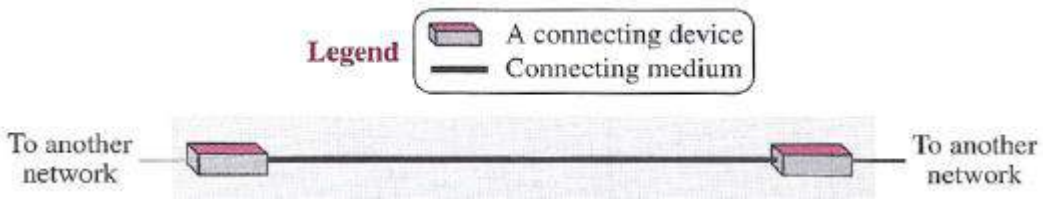
- When LANs were used in isolation (which is rare today), they were designed to allow resources to be shared between the hosts. LANs today are connected to each other and to WANs to create communication at a wider level.

2- Wide Area Network

- A wide area network (WAN) is also an interconnection of devices capable of communication.
- However, there are some differences between a LAN and a WAN.
- A LAN is normally limited in size, spanning an office, a building, or a campus; a WAN has a wider geographical span, spanning a town, a state, a country, or even the world.
- A LAN interconnects hosts; a WAN interconnects connecting devices such as switches, routers, or modems.
- A LAN is normally privately owned by the organization that uses it; a WAN is normally created and run by communication companies and leased by an organization that uses it.
- We see two distinct examples of WANs today: point-to-point WANs and switched WANs.

Point-to-Point WAN

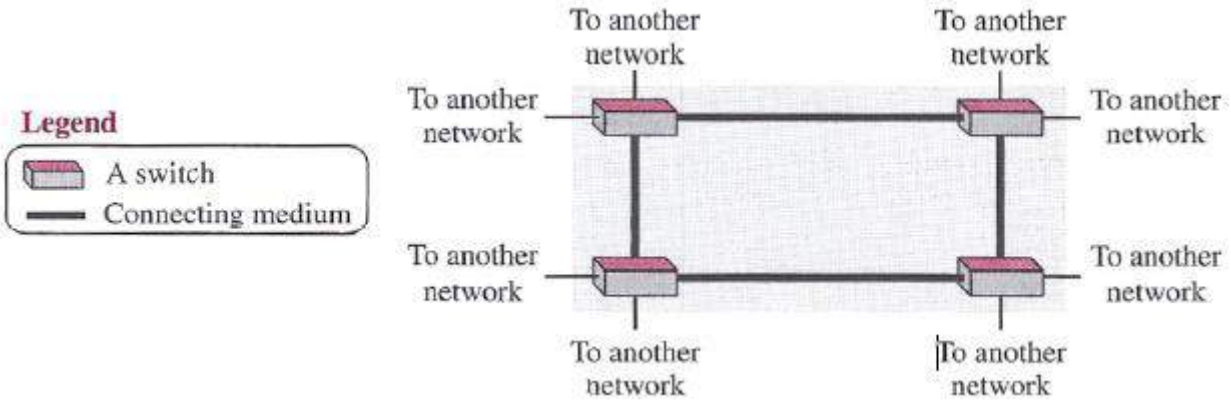
A point-to-point WAN is a network that connects two communicating devices through a transmission media (cable or air). The figure below shows an example of a point-to-point WAN.



Computer Networks

Switched WAN

A switched WAN is a network with more than two ends. A switched WAN is used in the backbone of global communication today. We can say that a switched WAN is a combination of several point-to-point WANs that are connected by switches. The figure below shows an example of a switched WAN.

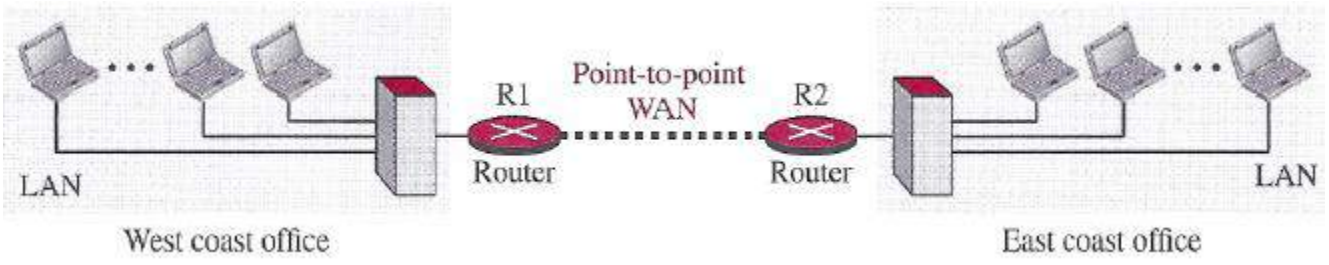


3- Metropolitan Area Networks

A metropolitan area network (MAN) is a network with a size between a LAN and a WAN. It normally covers the area inside a town or a city. It is designed for customers who need a high-speed connectivity, normally to the Internet, and have endpoints spread over a city or part of city. A good example of a MAN is the part of the telephone company network that can provide a high-speed DSL line to the customer.

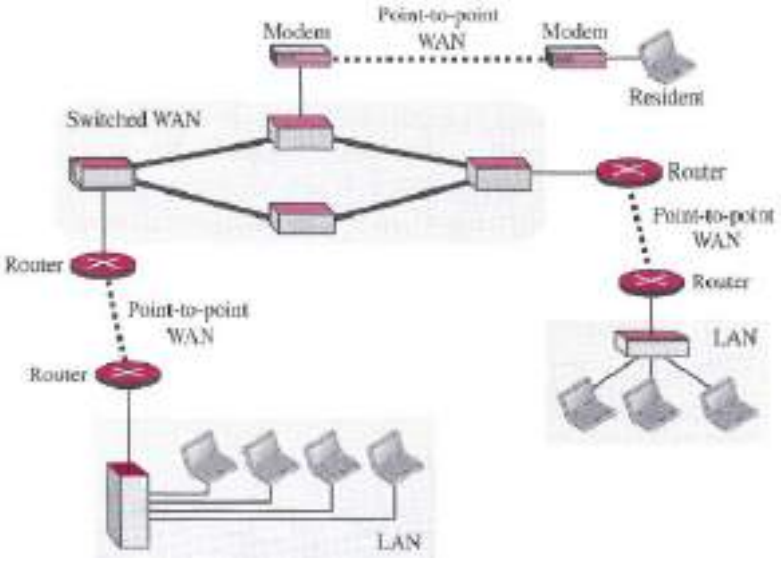
4- Internetwork

Today, it is very rare to see a LAN, a MAN, or a WAN in isolation; they are connected to one another. When two or more networks are connected, they make an internetwork, or internet (with lowercase i). As an example, assume that an organization has two offices, one on the east coast and the other on the west coast. Each office has a LAN that allows all employees in the office to communicate with each other. To make the communication between employees at different offices possible, the management leases a point-to-point dedicated WAN from a service provider, such as a telephone company, and connects the two LANs. Now the company has an internetwork, or a private internet. Communication between offices is now possible. The figure below shows this internet.



When a host in the west coast office sends a message to another host in the same office, the router blocks the message, but the switch directs the message to the destination. On the other hand, when a host on the west coast sends a message to a host on the east coast, router R1 routes the packet to router R2, and the packet reaches the destination.

The figure below shows another internet with several LANs and WANs connected. One of the WANs is a switched WAN with four switches.



5- The Internet

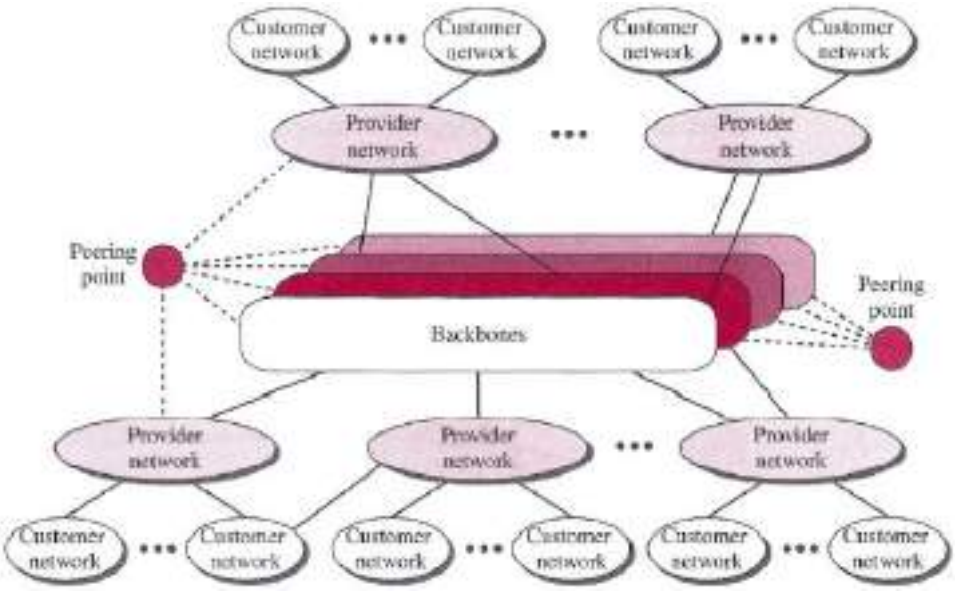
As we discussed before, an internet (note the lowercase i) is two or more networks that can communicate with each other. The most notable internet is called the Internet (uppercase I), and is composed of thousands of interconnected networks. The figure below shows a conceptual (not geographical) view of the Internet.

The figure shows the Internet as several backbones, provider networks, and customer networks.

- At the top level, the backbones are large networks owned by some communication companies such as Sprint, Verizon (MCI), AT&T, and NTT. The backbone networks are connected through some complex switching systems, called **peering points**.
- At the second level, there are smaller networks, called provider networks, that use the services of the backbones for a fee. The provider networks are connected to backbones and sometimes to other provider networks.
- The customer networks are networks at the edge of the Internet that actually use the services provided by the Internet. They pay fees to provider networks for receiving services.

Backbones and provider networks are also called **Internet Service Providers (ISPs)**. The backbones are often referred to as international ISPs; the provider networks are often referred to as national or regional ISPs.

Computer Networks



ACCESSING THE INTERNET

The Internet today is an internetwork that allows any user to become part of it. The user, however, needs to be physically connected to an ISP. The physical connection is normally done through a point-to-point WAN.

1- Using Telephone Networks

Today most residences and small businesses have telephone service, which means they are connected to a telephone network. Since most telephone networks have already connected themselves to the Internet, one option for residences and small businesses to connect to the Internet is to change the voice line between the residence or business and the telephone center to a point-to-point WAN. This can be done in two ways.

- **Dial-up service:** The first solution is to add to the telephone line a modem that converts data to voice. The software installed on the computer dials the ISP and imitates making a telephone connection. Unfortunately, the dial-up service is very slow, and when the line is used for Internet connection, it cannot be used for telephone (voice) connection. It is only useful for small residences.

- **DSL Service:** Since the advent of the Internet, some telephone companies have upgraded their telephone lines to provide higher speed Internet services to residences or small businesses. The DSL service also allows the line to be used simultaneously for voice and data communication.

2- Using Cable Networks

More and more residents over the last two decades have begun using cable TV services instead of antennas to receive TV broadcasting. The cable companies have been upgrading their cable networks and connecting to the Internet. A residence or a small business can be connected to the Internet by using this service. It provides a higher speed connection, but the speed varies depending on the number of neighbors that use the same cable.

3- Using Wireless Networks

Wireless connectivity has recently become increasingly popular. A household or a small business can use a combination of wireless and wired connections to access the Internet. With the growing wireless WAN access, a household or a small business can be connected to the Internet through a wireless WAN.

4- Direct Connection to the Internet

A large organization or a large corporation can itself become a local ISP and be connected to the Internet. This can be done if the organization or the corporation leases a high-speed WAN from a carrier provider and connects itself to a regional ISP. For example, a large university with several campuses can create an internetwork and then connect the internetwork to the Internet,

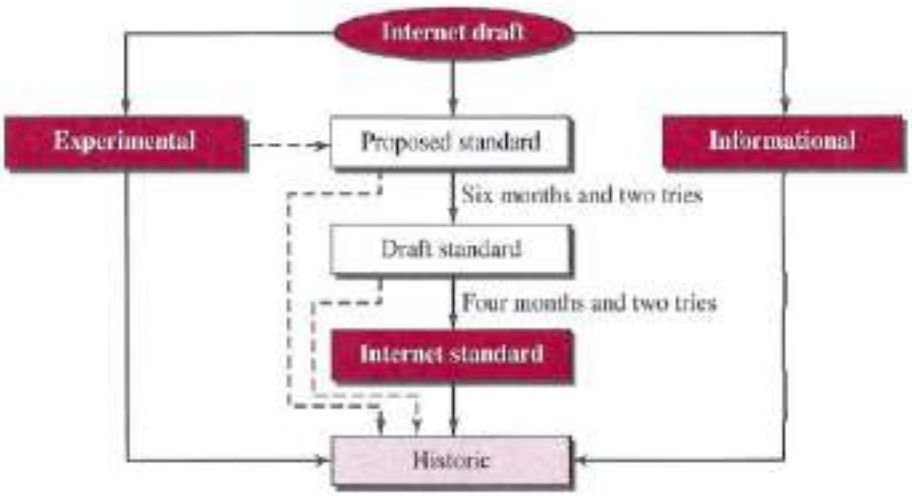
INTERNET STANDARDS

An Internet standard is a thoroughly tested specification that is useful to and adhered to by those who work with the Internet. It is a formalized regulation that must be followed. There is a strict procedure by which a specification attains Internet standard status. A specification begins as an Internet draft. An Internet draft is a working document (a work in progress) with no official status and a six-month lifetime. Upon recommendation from the Internet authorities, a draft may be published as a Request for Comment (RFC). Each RFC is edited, assigned a number, and made available to all interested parties. RFCs go through maturity levels and are categorized according to their requirement level.

Maturity Levels

An RFC, during its lifetime, falls into one of six maturity levels: proposed standard, draft standard, Internet standard, historic, experimental, and informational (see the figure below).

Computer Networks



- **Proposed Standard**: A proposed standard is a specification that is stable, well understood, and of sufficient interest to the Internet community. At this level, the specification is usually tested and implemented by several different groups.
- **Draft Standard**: A proposed standard is elevated to draft standard status after at least two successful independent and interoperable implementations. Barring difficulties, a draft standard, with modifications if specific problems are encountered, normally becomes an Internet standard.
- **Internet Standard**: A draft standard reaches Internet standard status after demonstrations of successful implementation.
- **Historic**: The historic RFCs are significant from a historical perspective. They either have been superseded by later specifications or have never passed the necessary maturity levels to become an Internet standard.
- **Experimental**: An RFC classified as experimental describes work related to an experimental situation that does not affect the operation of the Internet. Such an RFC should not be implemented in any functional Internet service.
- **Informational**: An RFC classified as informational contains general, historical, or tutorial information related to the Internet. It is usually written by someone in a non-Internet organization, such as a vendor.

Requirement Levels

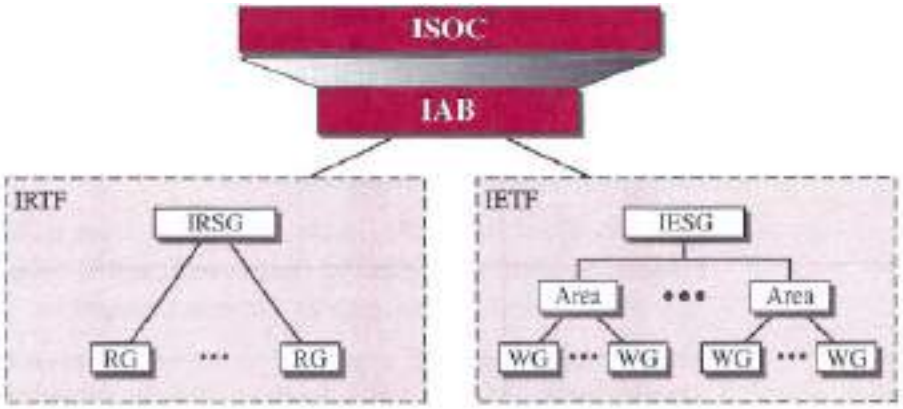
RFCs are classified into five requirement levels: required, recommended, elective, limited use, and not recommended.

- **Required**: An RFC is labeled required if it must be implemented by all Internet systems to achieve minimum conformance. For example, IP and ICMP are required protocols.

- **Recommended**: An RFC labeled recommended is not required for minimum conformance; it is recommended because of its usefulness. For example, FTP and TELNET are recommended protocols.
- **Elective**: An RFC labeled elective is not required and not recommended. However, a system can use it for its own benefit.
- **Limited Use**: An RFC labeled limited use should be used only in limited situations. Most of the experimental RFCs fall under this category.
- **Not Recommended**: An RFC labeled not recommended is inappropriate for general use. Normally a historic (deprecated) RFC may fall under this category.

INTERNET ADMINISTRATION

The Internet, with its roots primarily in the research domain, has evolved and gained a broader user base with significant commercial activity. Various groups that coordinate Internet issues have guided this growth and development. The figure below shows the general organization of Internet administration.



ISOC

The Internet Society (ISOC) is an international, nonprofit organization formed in 1992 to provide support for the Internet standards process. ISOC accomplishes this through maintaining and supporting other Internet administrative bodies such as IAB, IETF, IRTF, and IANA. ISOC also promotes research and other scholarly activities relating to the Internet.

IAB

The Internet Architecture Board (IAB) is the technical advisor to the ISOC. The main purposes of the IAB are to oversee the continuing development of the TCP/IP Protocol Suite and to serve in a technical advisory capacity to research members of the Internet community. IAB accomplishes this through its two primary components, the Internet Engineering Task Force (IETF) and the Internet Research Task Force (IRTF). Another responsibility of the IAB is the editorial management of the RFCs. IAB is also the external liaison between the Internet and other standards organizations and forums.

IETF

The Internet Engineering Task Force (IETF) is a forum of working groups managed by the Internet Engineering Steering Group (IESG). IETF is responsible for identifying operational problems and proposing solutions to these problems. IETF also develops and reviews specifications intended as Internet standards. The working groups are collected into areas, and each area concentrates on a specific topic. Currently nine areas have been defined. The areas include applications, protocols, routing, network management next generation (IPng), and security.

IRTF

The Internet Research Task Force (IRTF) is a forum of working groups managed by the Internet Research Steering Group (IRSG). IRTF focuses on long-term research topics related to Internet protocols, applications, architecture, and technology.

NETWORK SOFTWARE

Protocol layering

To reduce their design complexity, most networks are organized as a stack of layers or levels, each one built upon the one below it. The number of layers, the name of each layer, the contents of each layer, and the function of each layer differ from network to network. The purpose of each layer is to offer certain services to the higher layers while shielding those layers from the details of how the offered services are actually implemented. In a sense, each layer is a kind of virtual machine, offering certain services to the layer above it.

This concept is actually a familiar one and is used throughout computer science, where it is variously known as information hiding, abstract data types, data encapsulation, and object-oriented programming. The fundamental idea is that a particular piece of software (or hardware) provides a service to its users but keeps the details of its internal state and algorithms hidden from them.

When layer n on one machine carries on a conversation with layer n on another machine, the rules and conventions used in this conversation are collectively known as the layer n protocol. Basically, a protocol is an agreement between the communicating parties on how communication is to proceed and it defines the rules that both the sender and receiver and all intermediate devices need to follow to be able to communicate effectively. When communication is simple, we may need only one simple protocol; when the communication is complex, we may need to divide the task between different layers, in which case we need a protocol at each layer, or **protocol layering**.

Scenarios

Let us develop two simple scenarios to better understand the need for protocol layering.

- **First Scenario**

In the first scenario, communication is so simple that it can occur in only one layer. Assume Maria and Ann are neighbors with a lot of common ideas. Communication between Maria and Ann takes place in one layer, face to face, in the same language, as shown in the figure below.

Even in this simple scenario, we can see that a set of rules needs to be followed.

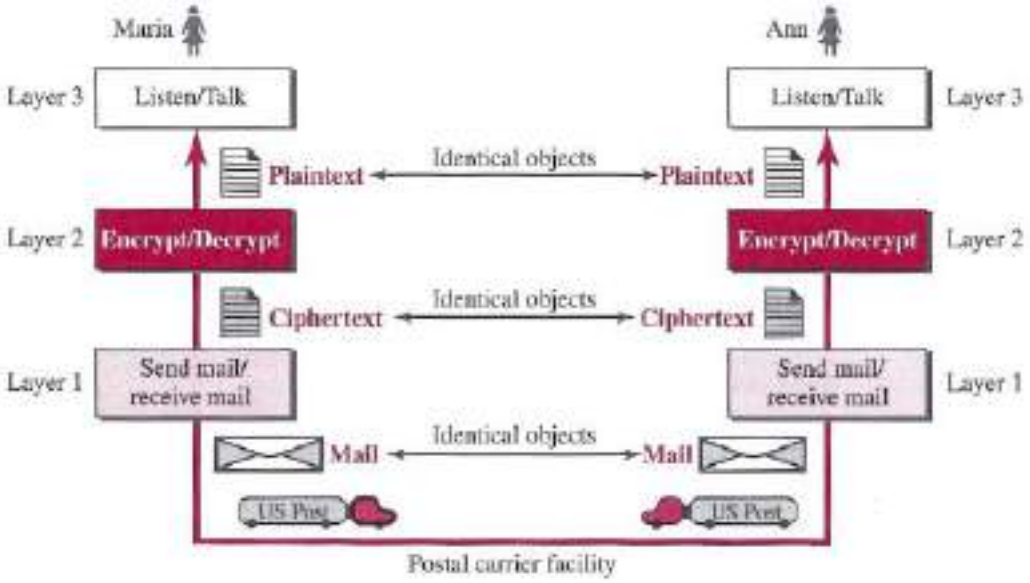
- First, Maria and Ann know that they should greet each other when they meet.
- Second, they know that they should confine their vocabulary to the level of their friendship.
- Third, each party knows that she should refrain from speaking when the other party is speaking.

- Fourth, each party knows that the conversation should be a dialog, not a monolog: both should have the opportunity to talk about the issue.
- Fifth, they should exchange some nice words when they leave.

We can see that the protocol used by Maria and Ann is different from the communication between a professor and the students in a lecture hall. The communication in the second case is mostly monolog; the professor talks most of the time unless a student has a question, a situation in which the protocol dictates that she should raise her hand and wait for permission to speak. In this case, the communication is normally very formal and limited to the subject being taught.

- **Second Scenario**

In the second scenario, we assume that Ann is offered a higher-level position in her company, but needs to move to another branch located in a city very far from Maria. The two friends still want to continue their communication and exchange ideas because they have come up with an innovative project to start a new business when they both retire. They decide to continue their conversation using regular mail through the post office. However, they do not want their ideas to be revealed by other people if the letters are intercepted. They agree on an encryption/decryption technique. The sender of the letter encrypts it to make it unreadable by an intruder; the receiver of the letter decrypts it to get the original letter. We assume that Maria and Ann use one technique that makes it hard to decrypt the letter if one does not have the key for doing so. Now we can say that the communication between Maria and Ann takes place in three layers, as shown in the figure below. We assume that Ann and Maria each have three machines (or robots) that can perform the task at each layer.



Let us assume that Maria sends the first letter to Ann. Maria talks to the machine at the third layer as though the machine is Ann and is listening to her. The third layer machine listens to what Maria says

and creates the plaintext (a letter in English), which is passed to the second layer machine. The second layer machine takes the plaintext, encrypts it, and creates the ciphertext, which is passed to the first layer machine. The first layer machine, presumably a robot, takes the ciphertext, puts it in an envelope, adds the sender and receiver addresses, and mails it.

At Ann's side, the first layer machine picks up the letter from Ann's mail box, recognizing the letter from Maria by the sender address. The machine takes out the ciphertext from the envelope and delivers it to the second layer machine. The second layer machine decrypts the message, creates the plaintext, and passes the plaintext to the third-layer machine. The third layer machine takes the plaintext and reads it as though Maria is speaking.

Protocol layering enables us to divide a complex task into several smaller and simpler tasks. For example, in the figure above, we could have used only one machine to do the job of all three machines. However, if Maria and Ann decide that the encryption/decryption done by the machine is not enough to protect their secrecy, they would have to change the whole machine. In the present situation, they need to change only the second layer machine; the other two can remain the same. This is referred to as modularity. Modularity in this case means independent layers. A layer (module) can be defined as a black box with inputs and outputs, without concern about how inputs are changed to outputs. If two machines provide the same outputs when given the same inputs, they can replace each other. For example, Ann and Maria can buy the second layer machine from two different manufacturers. As long as the two machines create the same ciphertext from the same plaintext and vice versa, they do the job.

Advantages of protocol layering

- One of the advantages of protocol layering is that it allows us to separate the services from the implementation. A layer needs to be able to receive a set of services from the lower layer and to give the services to the upper layer; we don't care about how the layer is implemented. For example, Maria may decide not to buy the machine (robot) for the first layer; she can do the job herself. As long as Maria can do the tasks provided by the first layer, in both directions, the communication system works.
- Another advantage of protocol layering, which cannot be seen in our simple examples but reveals itself when we discuss protocol layering in the Internet, is that communication does not always use only two end systems; there are intermediate systems that need only some layers, but not all layers. If we did not use protocol layering, we would have to make each intermediate system as complex as the end systems, which makes the whole system more expensive.

Is there any disadvantage to protocol layering? One can argue that having a single layer makes the job easier. There is no need for each layer to provide a service to the upper layer and give service to the lower layer. For example, Ann and Maria could find or build one machine that could do all three tasks. However, as mentioned above, if one day they found that their code was broken, each would have to replace the whole machine with a new one instead of just changing the machine in the second layer.

Principles of Protocol Layering

Let us discuss two principles of protocol layering.

First Principle

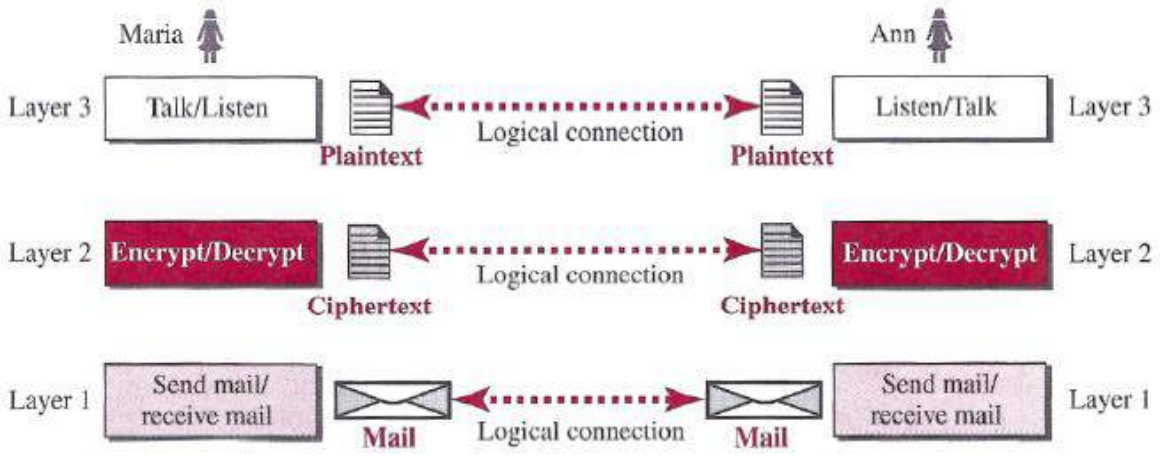
The first principle dictates that if we want bidirectional communication, we need to make each layer so that it is able to perform two opposite tasks, one in each direction. For example, the third layer task is to listen (in one direction) and talk (in the other direction). The second layer needs to be able to encrypt and decrypt. The first layer needs to send and receive mail.

Second Principle

The second principle that we need to follow in protocol layering is that the two objects under each layer at both sites should be identical. For example, the object under layer 3 at both sites should be a plaintext letter. The object under layer 2 at both sites should be a cipher text letter. The object under layer 1 at both sites should be a piece of mail.

Logical Connections

After following the above two principles, we can think about logical connection between each layer as shown in the figure below. This means that we have layer-to-layer communication. Maria and Ann can think that there is a logical (imaginary) connection at each layer through which they can send the object created from that layer.



Computer Networks

The entities comprising the corresponding layers on different machines are called peers. The peers may be software processes, hardware devices, or even human beings. In other words, it is the peers that communicate by using the protocol to talk to each other.

In reality, no data are directly transferred from layer n on one machine to layer n on another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the physical medium through which actual communication occurs.

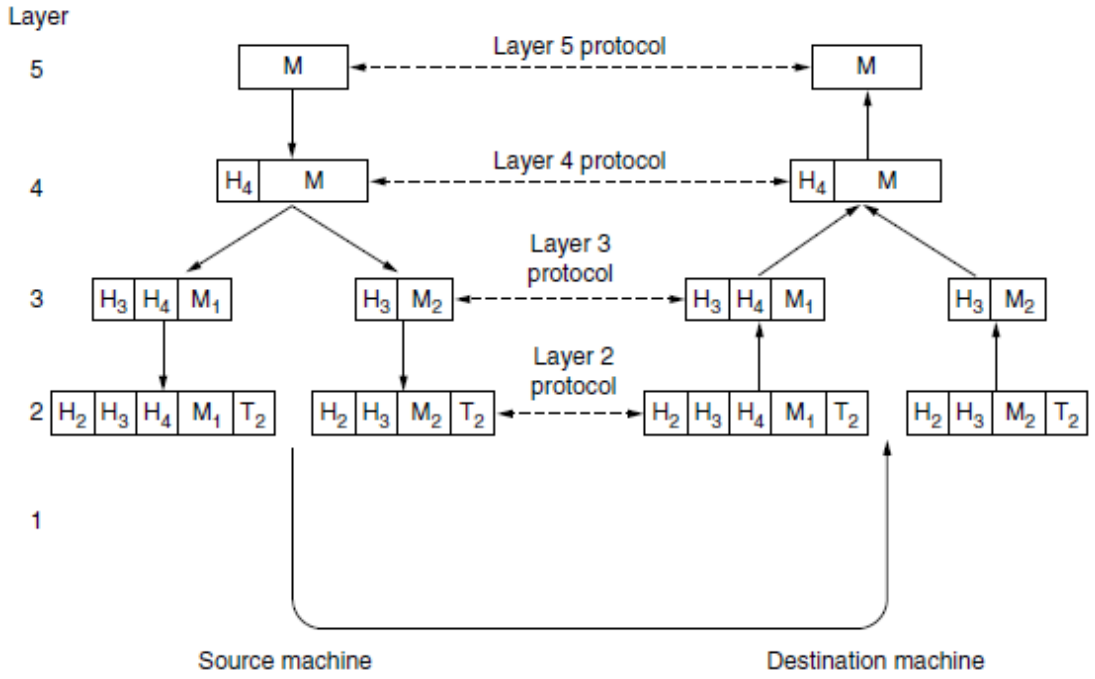
Between each pair of adjacent layers is an interface. The interface defines which primitive operations and services the lower layer makes available to the upper one.

A set of layers and protocols is called a network architecture.

A list of the protocols used by a certain system, one protocol per layer, is called a protocol stack.

Now let us understand how to provide communication to the top layer of the five-layer network as shown in Figure below.

A message, M, is produced by an application process running in layer 5 and given to layer 4 for transmission. Layer 4 puts a header in front of the message to identify the message and passes the result to layer 3. The header includes control information, such as addresses, to allow layer 4 on the destination machine to deliver the message. Other examples of control information used in some layers are sequence numbers (in case the lower layer does not preserve message order), sizes, and times.



In many networks, no limit is placed on the size of messages transmitted in the layer 4 protocol but there is nearly always a limit imposed by the layer 3 protocol. Consequently, layer 3 must break up the incoming messages into smaller units, packets, prepending a layer 3 header to each packet. In this example, M is split into two parts, M1 and M2, that will be transmitted separately. Layer 3 decides which of the outgoing lines to use and passes the packets to layer 2.

Layer 2 adds to each piece not only a header but also a trailer, and gives the resulting unit to layer 1 for physical transmission. At the receiving machine the message moves upward, from layer to layer, with headers being stripped off as it progresses. None of the headers for layers below n are passed up to layer n.

Network Criteria

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

1- 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.

2- 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.

3- Security

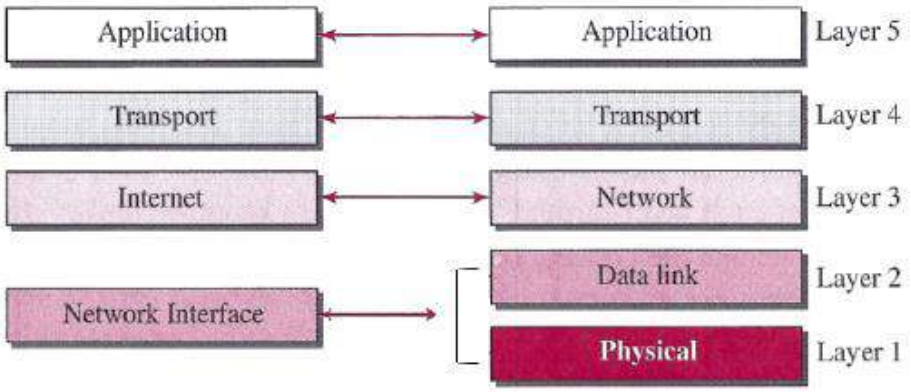
Network security issues include protecting data from unauthorized access, protecting data from damage and development, and implementing policies and procedures for recovery from breaches and data losses.

Computer Networks

NETWORK MODELS

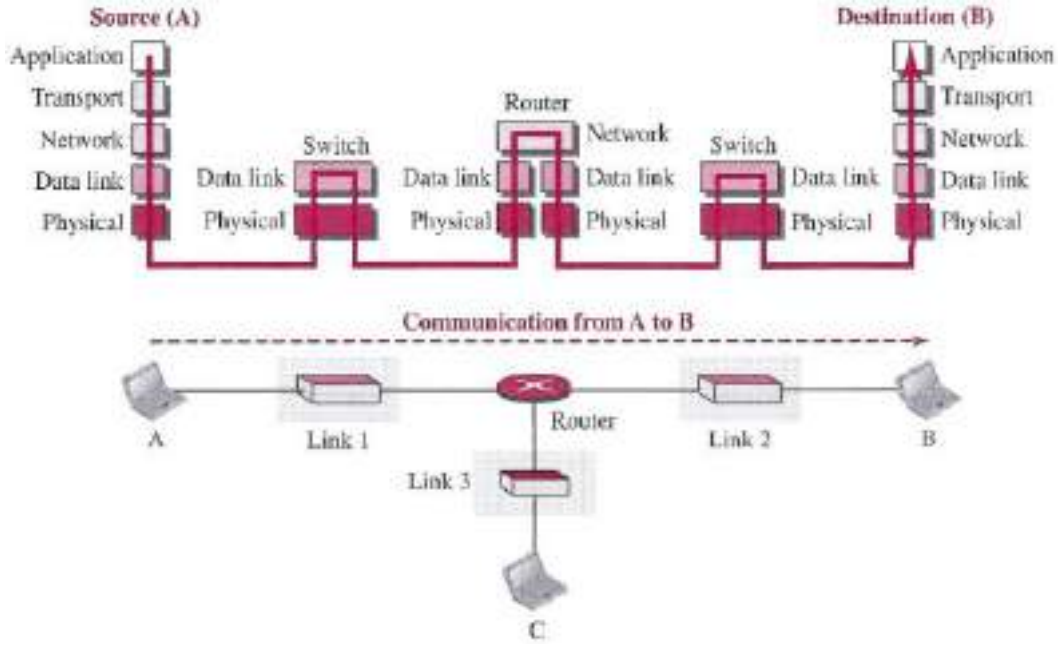
TCP/IP Protocol Suite

TCP/IP (Transmission Control Protocol/Internet Protocol) is a protocol suite (a set of protocols organized in different layers) used in the Internet today. It is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality. The term hierarchical means that each upper level protocol is supported by the services provided by one or more lower level protocols. The original TCP/IP protocol suite was defined as four software layers built upon the hardware. Today, however, TCP/IP is thought of as a five-layer model. The figure below shows both configurations.



Layered Architecture

To show how the layers in the TCP/IP protocol suite are involved in communication between two hosts, we assume that we want to use the suite in a small internet made up of three LANs (links), each with a link-layer switch. We also assume that the links are connected by one router, as shown in the figure below.



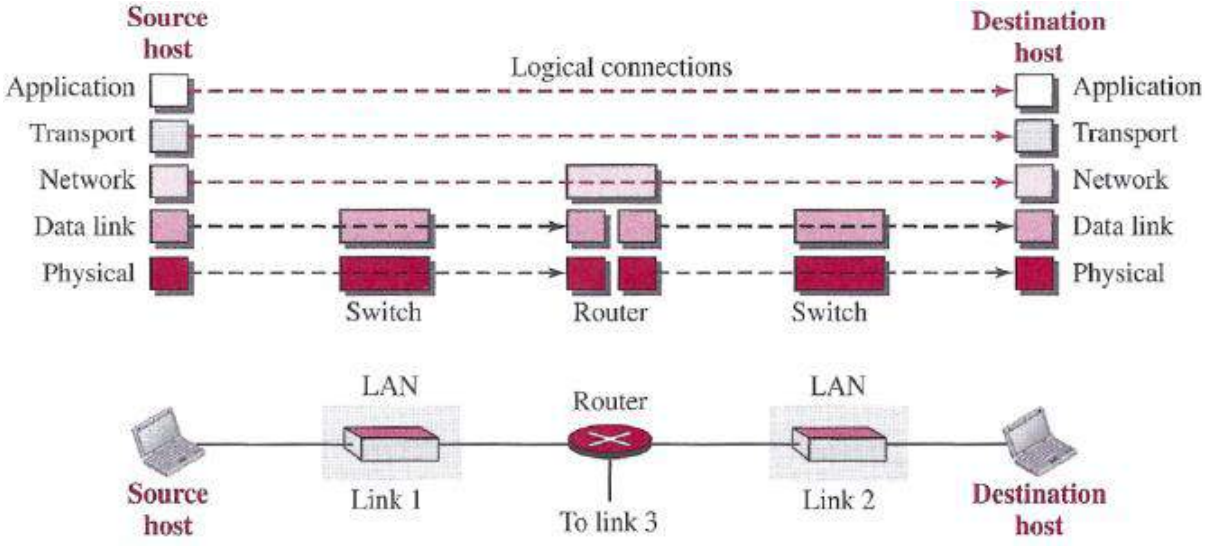
Let us assume that computer A communicates with computer B. As the figure shows, we have five communicating devices in this communication: source host (computer A), the link-layer switch in link1,

the router, the link-layer switch in link2, and the destination host (computer B). Each device is involved with a set of layers depending on the role of the device in the internet.

- The two hosts are involved in all five layers; the source host needs to create a message in the application layer and send it down the layers so that it is physically sent to the destination host. The destination host needs to receive the communication at the physical layer and then deliver it through the other layers to the application layer.
- The router is involved in only three layers; there is no transport or application layer in a router as long as the router is used only for routing. Although a router is always involved in one network layer, it is involved in n combinations of link and physical layers in which n is the number of links the router is connected to. The reason is that each link may use its own data-link or physical protocol. For example, in the above figure, the router is involved in three links, but the message sent from source A to destination B is involved in two links. Each link may be using different link-layer and physical-layer protocols; the router needs to receive a packet from link 1 based on one pair of protocols and deliver it to link 2 based on another pair of protocols.
- A link-layer switch in a link, however, is involved only in two layers, data-link and physical. Although each switch in the above figure has two different connections, the connections are in the same link, which uses only one set of protocols. This means that, unlike a router, a link-layer switch is involved only in one data-link and one physical layer.

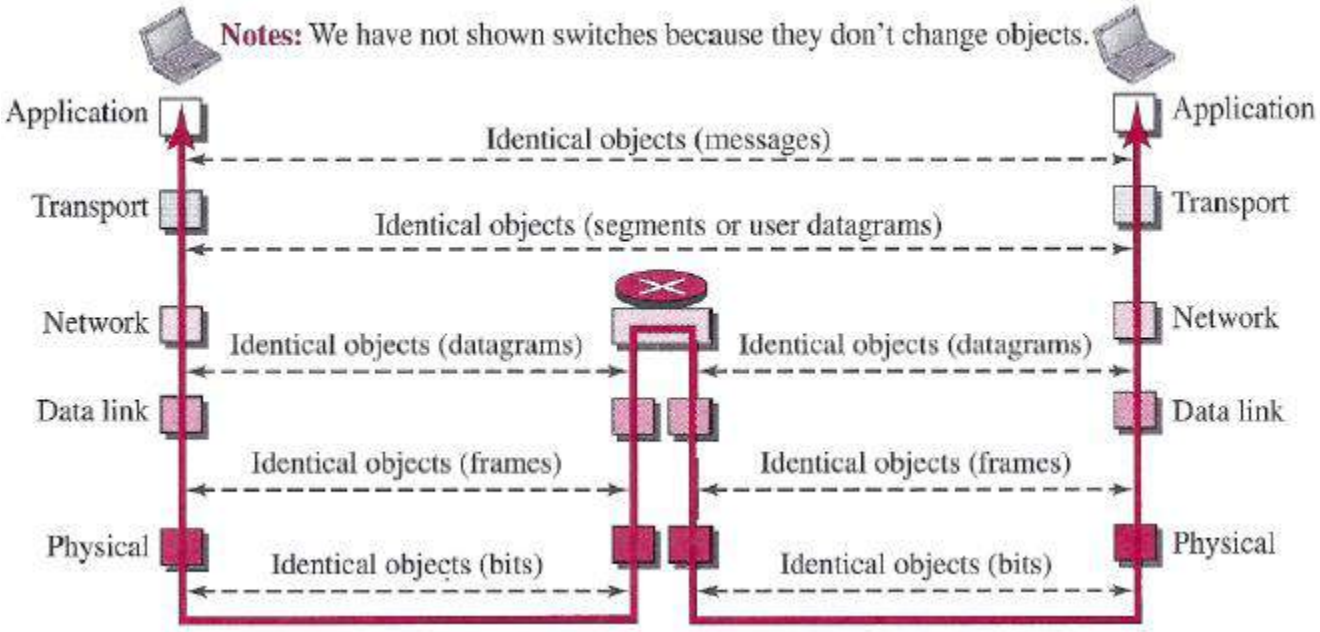
Layers in the TCP/IP Protocol Suite

After the above introduction, we briefly discuss the functions and duties of layers in the TCP/IP protocol suite. Each layer is discussed in detail later. To better understand the duties of each layer, we need to think about the logical connections between layers. The figure below shows logical connections in our simple internet.



Using logical connections makes it easier for us to think about the duty of each layer. As the figure shows, the duty of the application, transport, and network layers is end-to-end. However, the duty of the data-link and physical layers is hop-to-hop, in which a hop is a host or router. In other words, the domain of duty of the top three layers is the internet, and the domain of duty of the two lower layers is the link.

Another way of thinking of the logical connections is to think about the data unit created from each layer. In the top three layers, the data unit (packets) should not be changed by any router or link-layer switch. In the bottom two layers, the packet created by the host is changed only by the routers, not by the link-layer switches. The figure below shows the second principle discussed previously for protocol layering. We show the identical objects below each layer related to each device.



Note that, although the logical connection at the network layer is between the two hosts, we can only say that identical objects exist between two hops in this case because a router may fragment the packet at the network layer and send more packets than received. Note that the link between two hops does not change the object.

Description of Each Layer

1- Physical Layer

We can say that the physical layer is responsible for carrying individual bits in a frame across the link. Although the physical layer is the lowest level in the TCP/IP protocol suite, the communication between two devices at the physical layer is still a logical communication because there is another, hidden layer,

the transmission media, under the physical layer. Two devices are connected by a transmission medium (cable or air). We need to know that the transmission medium does not carry bits; it carries electrical or optical signals. So the bits received in a frame from the data-link layer are transformed and sent through the transmission media, but we can think that the logical unit between two physical layers in two devices is a bit. There are several protocols that transform a bit to a signal.

2- Data-link Layer

We have seen that an internet is made up of several links (LANs and WANs) connected by routers. There may be several overlapping sets of links that a datagram can travel from the host to the destination. The routers are responsible for choosing the best links. However, when the next link to travel is determined by the router, the data-link layer is responsible for taking the datagram and moving it across the link. The link can be a wired LAN with a link-layer switch, a wireless LAN, a wired WAN, or a wireless WAN. We can also have different protocols used with any link type. In each case, the data-link layer is responsible for moving the packet through the link. TCP/IP does not define any specific protocol for the data-link layer. It supports all the standard and proprietary protocols. Any protocol that can take the datagram and carry it through the link suffices for the network layer. The data-link layer takes a datagram and encapsulates it in a packet called «frame». Each link-layer protocol may provide a different service. Some link-layer protocols provide complete error detection and correction; some provide only error correction.

3- Network Layer

The network layer is responsible for creating a connection between the source computer and the destination computer. The communication at the network layer is host-to-host. However, since there can be several routers from the source to the destination, the routers in the path are responsible for choosing the best route for each packet. We can say that the network layer is responsible for host-to-host communication and routing the packet through possible routes.

Again, we may ask ourselves why we need the network layer. We could have added the routing duty to the transport layer and dropped this layer.

- One reason, as we said before, is the separation of different tasks between different layers.
- The second reason is that the routers do not need the application and transport layers. Separating the tasks allows us to use fewer protocols on the routers.

The network layer in the Internet includes the main protocol, Internet Protocol (IP), that defines the format of the packet, called a datagram at the network layer. IP also defines the format and the

structure of addresses used in this layer. IP is also responsible for routing a packet from its source to its destination, which is achieved by each router forwarding the datagram to the next router in its path. IP is a connectionless protocol that provides no flow control, no error control, and no congestion control services. This means that if any of these services is required for an application, the application should rely only on the transport-layer protocol. The network layer also includes unicast (one-to-one) and multicast (one-to-many) routing protocols. A routing protocol does not take part in routing (it is the responsibility of IP), but it creates forwarding tables for routers to help them in the routing process.

The network layer also has some auxiliary protocols that help IP in its delivery and routing tasks. The Internet Control Message Protocol (ICMP) helps IP to report some problems when routing a packet. The Internet Group Management Protocol (IGMP) is another protocol that helps IP in multicasting. The Dynamic Host Configuration Protocol (DHCP) helps IP to get the network-layer address for a host. The Address Resolution Protocol (ARP) is a protocol that helps IP to find the link-layer address of a host or a router when its network-layer address is given.

4- Transport Layer

The logical connection at the transport layer is also end-to-end. The transport layer at the source host gets the message from the application layer, encapsulates it in a transport layer packet (called a segment or a user datagram in different protocols) and sends it, through the logical (imaginary) connection, to the transport layer at the destination host. In other words, the transport layer is responsible for giving services to the application layer: to get a message from an application program running on the source host and deliver it to the corresponding application program on the destination host.

We may ask why we need an end-to-end transport layer when we already have an end-to-end application layer. The reason is the separation of tasks and duties, which we discussed earlier. The transport layer should be independent of the application layer. In addition, we will see that we have more than one protocol in the transport layer, which means that each application program can use the protocol that best matches its requirement. As we said, there are a few transport-layer protocols in the Internet, each designed for some specific task.

The main protocol, Transmission Control Protocol (TCP), is a connection-oriented protocol that first establishes a logical connection between transport layers at two hosts before transferring data. It creates a logical pipe between two TCPs for transferring a stream of bytes. TCP provides flow control (matching the sending data rate of the source host with the receiving data rate of the destination host to prevent overwhelming the destination), error control (to guarantee that the segments arrive at the

destination without error and resending the corrupted ones), and congestion control to reduce the loss of segments due to congestion in the network.

The other common protocol, User Datagram Protocol (UDP), is a connectionless protocol that transmits user datagrams without first creating a logical connection. In UDP, each user datagram is an independent entity without being related to the previous or the next one (the meaning of the term connectionless). UDP is a simple protocol that does not provide flow, error, or congestion control. Its simplicity, which means small overhead, is attractive to an application program that needs to send short messages and cannot afford the retransmission of the packets involved in TCP, when a packet is corrupted or lost.

A new protocol, Stream Control Transmission Protocol (SCTP) is designed to respond to new applications that are emerging in the multimedia.

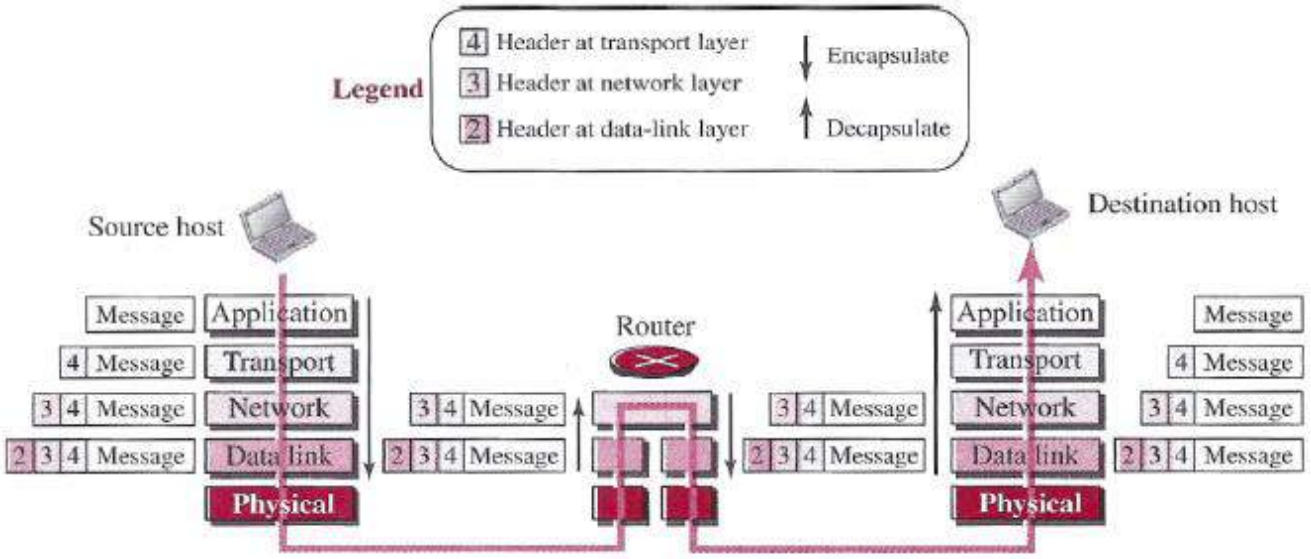
5- Application Layer

The logical connection between the two application layers is end-to-end. The two application layers exchange messages between each other as though there were a bridge between the two layers. However, we should know that the communication is done through all the layers. Communication at the application layer is between two processes (two programs running at this layer). To communicate, a process sends a request to the other process and receives a response. Process-to-process communication is the duty of the application layer. The application layer in the Internet includes many predefined protocols, but a user can also create a pair of processes to be run at the two hosts. The Hypertext Transfer Protocol (HTTP) is a vehicle for accessing the World Wide Web (WWW). The Simple Mail Transfer Protocol (SMTP) is the main protocol used in electronic mail (e-mail) service. The File Transfer Protocol (FTP) is used for transferring files from one host to another. The Terminal Network (TELNET) and Secure Shell (SSH) are used for accessing a site remotely. The Simple Network Management Protocol (SNMP) is used by an administrator to manage the Internet at global and local levels. The Domain Name System (DNS) is used by other protocols to find the network-layer address of a computer.

Computer Networks

Encapsulation and Decapsulation

One of the important concepts in protocol layering in the Internet is encapsulation/decapsulation. The figure below shows this concept for the small internet.



We have not shown the layers for the link-layer switches because no encapsulation/decapsulation occurs in this device. In the figure above, we show the encapsulation in the source host, decapsulation in the destination host, and encapsulation and decapsulation in the router.

Encapsulation at the Source Host

At the source, we have only encapsulation.

1- At the application layer, the data to be exchanged is referred to as a message. A message normally does not contain any header or trailer, but if it does, we refer to the whole as the message. The message is passed to the transport layer.

2- The transport layer takes the message as the payload, the load that the transport layer should take care of. It adds the transport layer header to the payload, which contains the identifiers of the source and destination application programs that want to communicate plus some more information that is needed for the end-to-end delivery of the message, such as information needed for flow, error control, or congestion control. The result is the transport-layer packet, which is called the segment (in TCP) and the user datagram (in UDP). The transport layer then passes the packet to the network layer.

3- The network layer takes the transport-layer packet as data or payload and adds its own header to the payload. The header contains the addresses of the source and destination hosts and some more information used for error checking of the header, fragmentation information, and so on. The

result is the network-layer packet, called a datagram. The network layer then passes the packet to the data-link layer.

4- The data-link layer takes the network-layer packet as data or payload and adds its own header, which contains the link-layer addresses of the host or the next hop (the router). The result is the link-layer packet, which is called a frame. The frame is passed to the physical layer for transmission.

Decapsulation and Encapsulation at the Router

At the router, we have both decapsulation and encapsulation because the router is connected to two or more links.

1- After the set of bits are delivered to the data-link layer, this layer decapsulates the datagram from the frame and passes it to the network layer.

2- The network layer only inspects the source and destination addresses in the datagram header and consults its forwarding table to find the next hop to which the datagram is to be delivered. The contents of the datagram should not be changed by the network layer in the router unless there is a need to fragment the datagram if it is too big to be passed through the next link. The datagram is then passed to the data-link layer of the next link.

3- The data-link layer of the next link encapsulates the datagram in a frame and passes it to the physical layer for transmission.

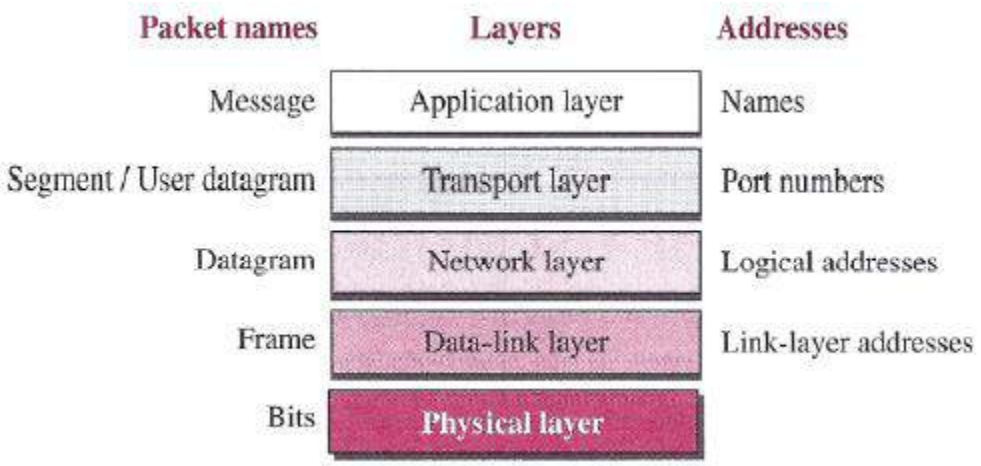
Decapsulation at the Destination Host

At the destination host, each layer only decapsulates the packet received, removes the header, and delivers the payload to the next-higher layer protocol until the message reaches the application layer. It is necessary to say that decapsulation in the host involves error checking.

Addressing

It is worth mentioning another concept related to protocol layering in the Internet, addressing. As we discussed before, we have logical communication between pairs of layers in this model. Any communication that involves two parties needs two addresses: source address and destination address. Although it looks as if we need five pairs of addresses, one pair per layer, we normally have only four because the physical layer does not need addresses; the unit of data exchange at the physical layer is a bit, which definitely cannot have an address. The figure below shows the addressing at each layer.

Computer Networks

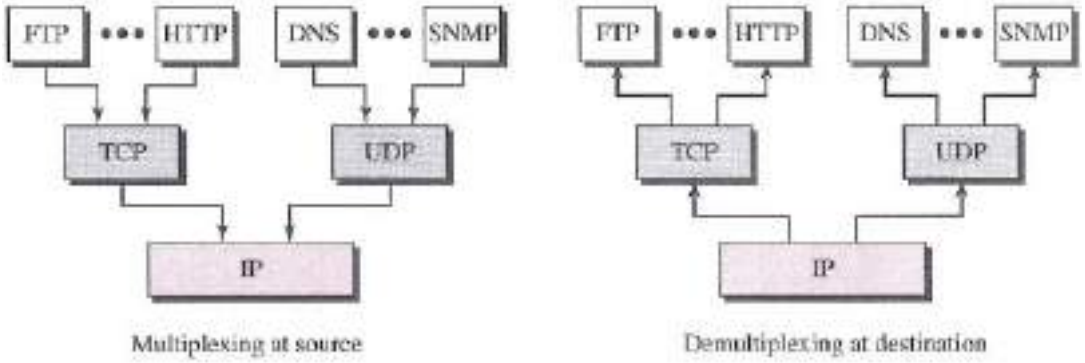


As the figure shows, there is a relationship between the layer, the address used in that layer, and the packet name at that layer.

- At the application layer, we normally use names to define the site that provides services, such as someorg.com, or the e-mail address, such as somebody@coldmail.com.
- At the transport layer, addresses are called port numbers, and these define the application-layer programs at the source and destination. Port numbers are local addresses that distinguish between several programs running at the same time.
- At the network-layer, the addresses are global, with the whole Internet as the scope. A network-layer address uniquely defines the connection of a device to the Internet.
- The link-layer addresses, sometimes called MAC addresses, are locally defined addresses, each of which defines a specific host or router in a network (LAN or WAN).

Multiplexing and Demultiplexing

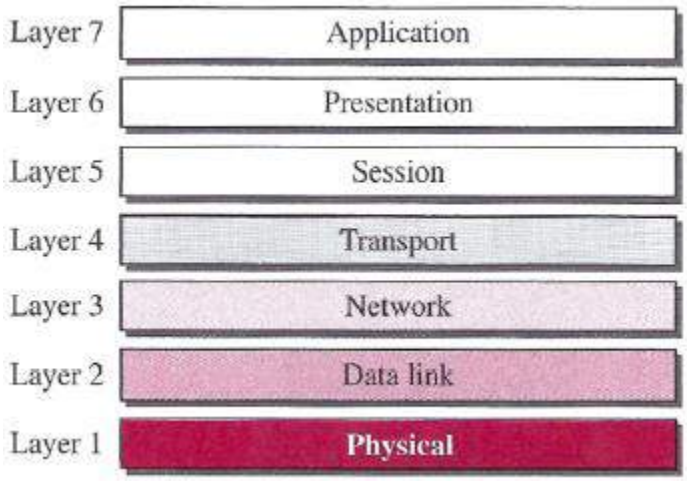
Since the TCP/IP protocol suite uses several protocols at some layers, we can say that we have multiplexing at the source and demultiplexing at the destination. Multiplexing in this case means that a protocol at a layer can encapsulate a packet from several next-higher layer protocols (one at a time); demultiplexing means that a protocol can decapsulate and deliver a packet to several next-higher layer protocols (one at a time). The figure below shows the concept of multiplexing and demultiplexing at the three upper layers.



To be able to multiplex and demultiplex, a protocol needs to have a field in its header to identify to which protocol the encapsulated packets belong. At the transport layer, either UDP or TCP can accept a message from several application-layer protocols. At the network layer, IP can accept a segment from TCP or a user datagram from UDP. IP can also accept a packet from other protocols such as ICMP, IGMP, and so on. At the data-link layer, a frame may carry the payload coming from IP or other protocols such as ARP.

THE OSI MODEL

Although, when speaking of the Internet, everyone talks about the TCP/IP protocol suite, this suite is not the only suite of protocols defined. Established in 1947, the International Organization for Standardization (ISO) is a multinational body dedicated to worldwide agreement on international standards. Almost three-fourths of the countries in the world are represented in the ISO. An ISO standard that covers all aspects of network communications is the Open Systems Interconnection (OSI) model. It was first introduced in the late 1970s. An open system is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture. The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software. The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable. The OSI model was intended to be the basis for the creation of the protocols in the OSI stack. The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network (see the figure below).



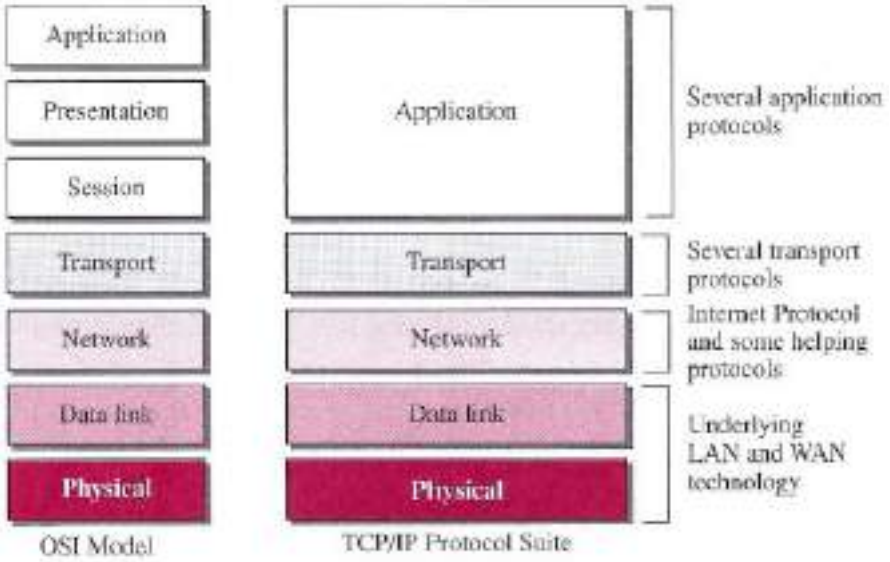
The seven layers are divided into two groups.

- The top three layers define how the applications within the end stations will communicate with each other and with users.
- The bottom four layers define how data is transmitted end to end.

The user interfaces with the computer at the Application layer and also that the upper layers are responsible for applications communicating between hosts. Remember that none of the upper layers knows anything about networking or network addresses. That's the responsibility of the four bottom layers. The four bottom layers define how data is transferred through physical media, switches, and routers. These bottom layers also determine how to rebuild a data stream from a transmitting host to a destination host's application.

OSI versus TCP/IP

When we compare the two models, we find that two layers, session and presentation, are missing from the TCP/IP protocol suite. These two layers were not added to the TCP/IP protocol suite after the publication of the OSI model. The application layer in the suite is usually considered to be the combination of three layers in the OSI model, as shown in the figure below.



Two reasons were mentioned for this decision.

- First, TCP/IP has more than one transport-layer protocol. Some of the functionalities of the session layer are available in some of the transport-layer protocols.
- Second, the application layer is not only one piece of software. Many applications can be developed at this layer. If some of the functionalities mentioned in the session and presentation layers are needed for a particular application, they can be included in the development of that piece of software.

Lack of OSI Model's Success

The OSI model appeared after the TCP/IP protocol suite. Most experts were at first excited and thought that the TCP/IP protocol would be fully replaced by the OSI model. This did not happen for several reasons, but we describe only three, which are agreed upon by all experts in the field.

- First, OSI was completed when TCP/IP was fully in place and a lot of time and money had been spent on the suite; changing it would cost a lot.
- Second, some layers in the OSI model were never fully defined. For example, although the services provided by the presentation and the session layers were listed in the document, actual protocols for these two layers were not fully defined, nor were they fully described, and the corresponding software was not fully developed.
- Third, when OSI was implemented by an organization in a different application, it did not show a high enough level of performance to entice the Internet authority to switch from the TCP/IP protocol suite to the OSI model.

Computer Networks

PHYSICAL LAYER

Physical layer is the only layer which actually deals with the physical connectivity of two different stations. Physical layer provides its services to Data-link layer. Data-link layer hands over frames to physical layer. One of the major functions of the physical layer is to move data in the form of electromagnetic signals across a transmission medium. Whether you are collecting numerical statistics from another computer, sending animated pictures from a design workstation, or causing a bell to ring at a distant control center, you are working with the transmission of data across network connections. Generally, the data usable to a person or application are not in a form that can be transmitted over a network. For example, a photograph must first be changed to a form that transmission media can accept. For transmission, data needs to be changed to signals.

Physical layer is concerned with transmitting raw bits over a communication channel. The design issues have to do with making sure that when one side sends a 1 bit, it is received by the other side as 1 bit and not as 0 bit. In physical layer we deal with the communication medium used for transmission.

Bandwidth: Simply means how many bits can be transmitted per second in the communication channel. In technical terms it indicates the width of frequency spectrum.

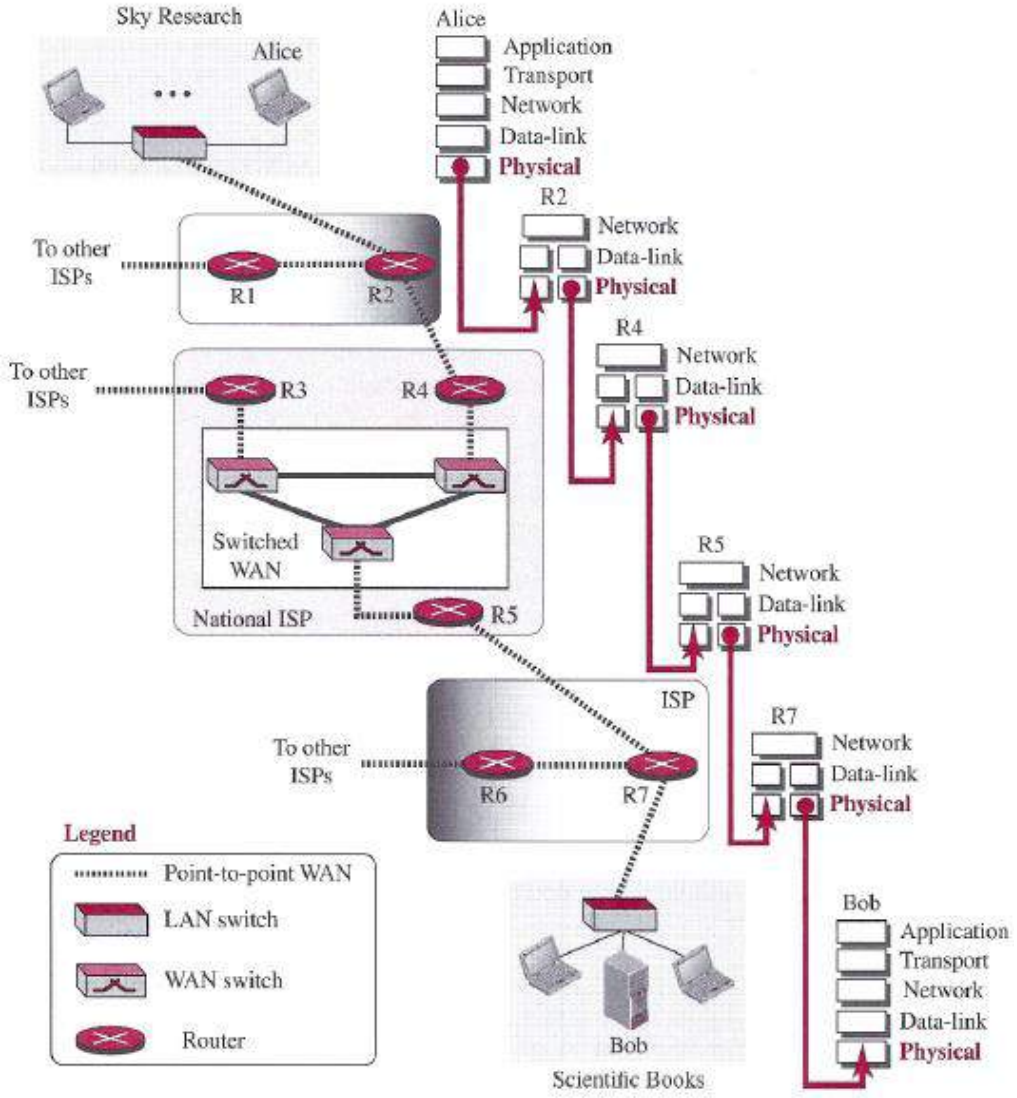
DATA AND SIGNALS

The figure below shows a scenario in which a scientist working in a research company, Sky Research, needs to order a book related to her research from an online bookseller, Scientific Books.

We can think of five different levels of communication between Alice, the computer on which our scientist is working, and Bob, the computer that provides online service. Communication at application, transport, network, or data-link is logical; communication at the physical layer is physical. For simplicity, we have shown only host-to-router, router-to-router, and router-to-host, but the switches are also involved in the physical communication.

Although Alice and Bob need to exchange data, communication at the physical layer means exchanging signals. Data need to be transmitted and received, but the media have to change data to signals. Both data and the signals that represent them can be either analog or digital in form.

Computer Networks



Analog and Digital Data

Data can be analog or digital. The term analog data refers to information that is continuous; digital data refers to information that has discrete states. For example, an analog clock that has hour, minute, and second hands gives information in a continuous form; the movements of the hands are continuous. On the other hand, a digital clock that reports the hours and the minutes will change suddenly from 8:05 to 8:06.

Analog data, such as the sounds made by a human voice, take on continuous values. When someone speaks, an analog wave is created in the air. This can be captured by a microphone and converted to an analog signal or sampled and converted to a digital signal.

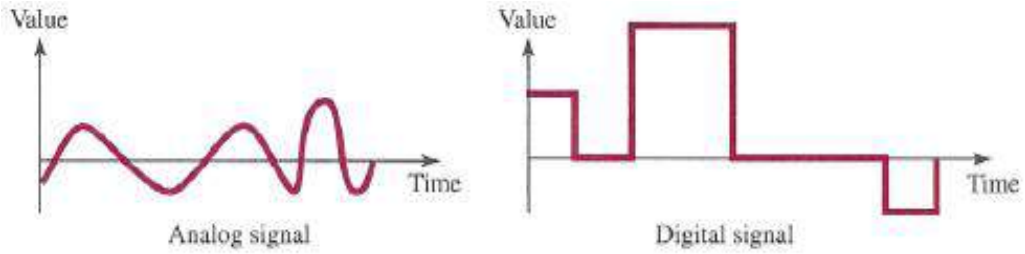
Digital data take on discrete values. For example, data are stored in computer memory in the form of 0s and 1s. They can be converted to a digital signal or modulated into an analog signal for transmission across a medium.

Analog and Digital Signals

Like the data they represent, signals can be either analog or digital.

- An analog signal has infinitely many levels of intensity over a period of time. As the wave moves from value A to value B, it passes through and includes an infinite number of values along its path.
- A digital signal, on the other hand, can have only a limited number of defined values. Although each value can be any number, it is often as simple as 1 and 0.

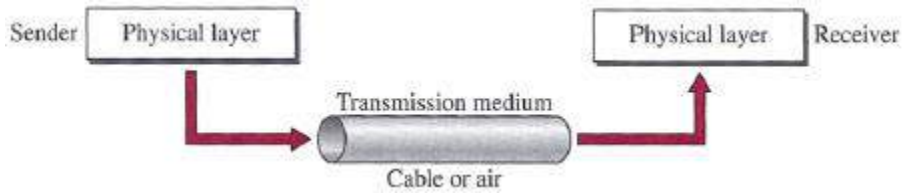
The simplest way to show signals is by plotting them on a pair of perpendicular axes. The vertical axis represents the value or strength of a signal. The horizontal axis represents time. The figure below illustrates an analog signal and a digital signal. The curve representing the analog signal passes through an infinite number of points. The vertical lines of the digital signal, however, demonstrate the sudden jump that the signal makes from value to value.



Computer Networks

Transmission Media

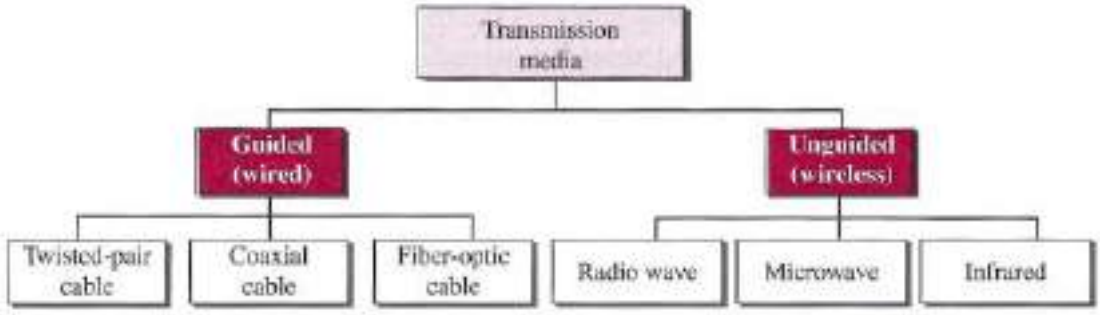
Transmission media are actually located below the physical layer and are directly controlled by the physical layer. We could say that transmission media belong to layer zero. The figure below shows the position of transmission media in relation to the physical layer.



A transmission medium can be broadly defined as anything that can carry information from a source to a destination. For example, the transmission medium for two people having a dinner conversation is the air.

In data communications the definition of the information and the transmission medium is more specific. The transmission medium is usually free space, metallic cable, or fiber-optic cable. The information is usually a signal that is the result of a conversion of data from another form.

In telecommunications, transmission media can be divided into two broad categories: guided and unguided. Guided media include twisted-pair cable, coaxial cable, and fiber-optic cable. Unguided medium is free space. The figure below shows this taxonomy.



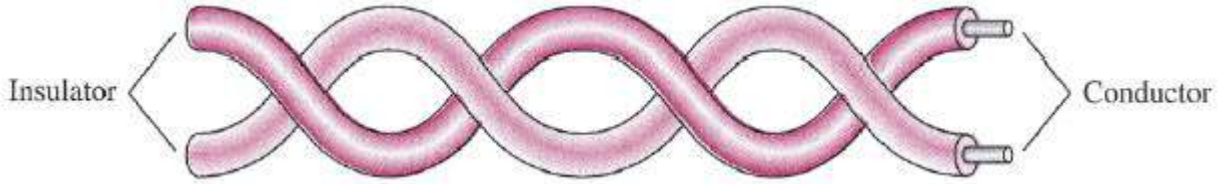
GUIDED MEDIA

Guided media, which are those that provide a conduit from one device to another, include twisted-pair cable, coaxial cable, and fiber-optic cable. A signal traveling along any of these media is directed and contained by the physical limits of the medium. Twisted-pair and coaxial cable use metallic (copper) conductors that accept and transport signals in the form of electric current. Optical fiber is a cable that accepts and transports signals in the form of light

Computer Networks

1- Twisted-Pair Cable

A twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together, as shown in the figure below.

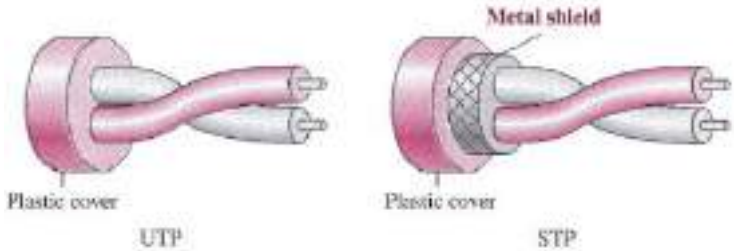


One of the wires is used to carry signals to the receiver, and the other is used only as a ground reference. The receiver uses the difference between the two. In addition to the signal sent by the sender on one of the wires, interference (noise) and crosstalk may affect both wires and create unwanted signals. If the two wires are parallel, the effect of these unwanted signals is not the same in both wires because they are at different locations relative to the noise or crosstalk sources (e.g., one is closer and the other is farther). This results in a difference at the receiver.

By twisting the pairs, a balance is maintained. For example, suppose in one twist, one wire is closer to the noise source and the other is farther; in the next twist, the reverse is true. Twisting makes it probable that both wires are equally affected by external influences (noise or crosstalk). This means that the receiver, which calculates the difference between the two, receives no unwanted signals. The unwanted signals are mostly canceled out. From the above discussion, it is clear that the number of twists per unit of length (e.g., inch) has some effect on the quality of the cable.

Unshielded Versus Shielded Twisted-Pair Cable

The most common twisted-pair cable used in communications is referred to as unshielded twisted-pair (UTP). IBM has also produced a version of twisted-pair cable for its use, called shielded twisted-pair (STP). STP cable has a metal foil or braided-mesh covering that encases each pair of insulated conductors. Although metal casing improves the quality of cable by preventing the penetration of noise or crosstalk, it is bulkier and more expensive. The figure below shows the difference between UTP and STP. Our discussion focuses primarily on UTP because STP is seldom used outside of IBM.



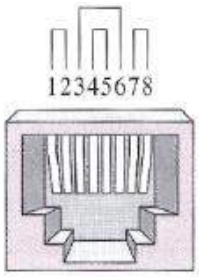
Categories

The Electronic Industries Association (EIA) has developed standards to classify unshielded twisted-pair cable into seven categories. Categories are determined by cable quality, with 1 as the lowest and 7 as the highest. Each EIA category is suitable for specific uses.

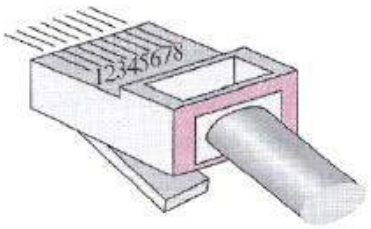
Category	Specification	Max Length	Data Rate (Mbps)	Use
1	Unshielded twisted-pair used in telephone	-	< 0.1	Telephone
2	Unshielded twisted-pair originally used in T lines	-	2	T-1 lines
3	Improved CAT 2 used in LANs	100 m	10	LANs
4	Improved CAT 3 used in Token Ring networks	100 m	20	LANs
5	Cable wire is normally 24 AWG with a jacket 100 LANs and outside sheath	100 m	100	LANs
5E	An extension to category 5 that includes extra features to minimize the crosstalk and electromagnetic interference	100 m	125	LANs
6	A new category with matched components coming from the same manufacturer. The cable must be tested at a 200-Mbps data rate	100 m	200	LANs
7	Sometimes called SSTP (shielded screen twisted-pair). Each pair is individually wrapped in a helical metallic foil followed by a metallic foil shield in addition to the outside sheath. The shield decreases the effect of crosstalk and increases the data rate.	100 m	600	LANs

Connectors

The most common UTP connector is RJ45 (RJ stands for registered jack), as shown in the figure below. The RJ45 is a keyed connector, meaning the connector can be inserted in only one way.



RJ-45 Female



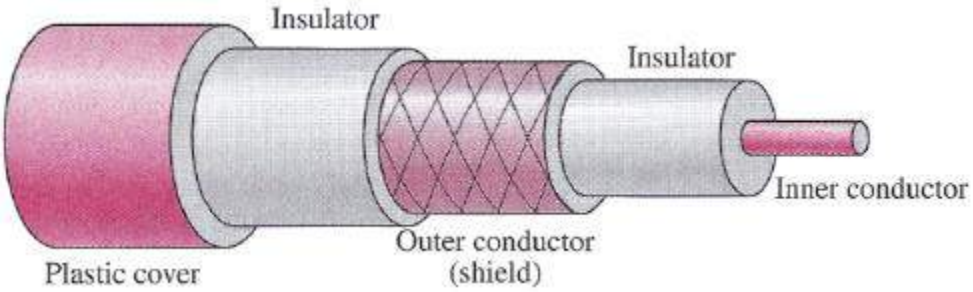
RJ-45 Male

Applications

- Telephone lines.
- The DSL lines.
- Local-area networks, such as 10Base-T and 100Base-T.

2- Coaxial Cable

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



Coaxial Cable Standards

Coaxial cables are categorized by their Radio Government (RG) ratings. Each RG number denotes a unique set of physical specifications, including the wire gauge of the inner conductor, the thickness and type of the inner insulator, the construction of the shield, and the size and type of the outer casing. Each cable defined by an RG rating is adapted for a specialized function, as shown in the table below.

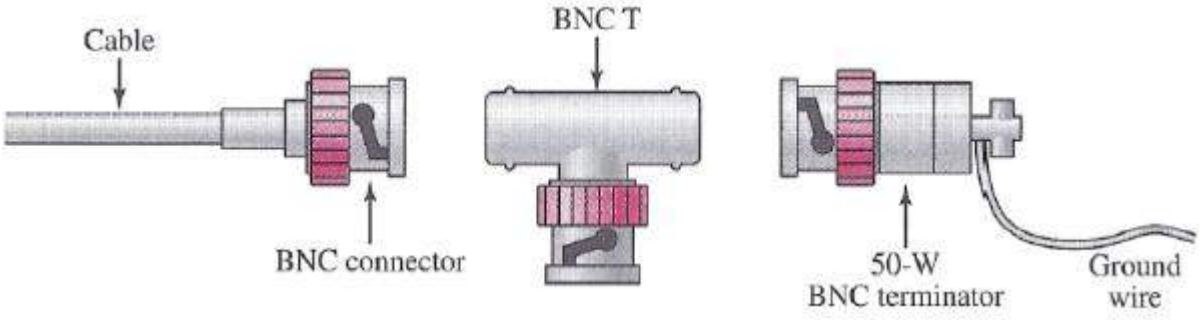
Category	Impedance	Use
RG-59	75 Ω	Cable TV
RG-58	50 Ω	Thin Ethernet
RG-11	50 Ω	Thick Ethernet

There are two types of coaxial cabling: Thinnet and Thicknet.

- **Thinnet** is a flexible coaxial cable about ¼ inch thick. Thinnet is used for short-distance. The maximum length of thinnet is 185 meters.
- **Thicknet** coaxial is thicker cable than thinnet. Thicknet cable is about ½ inch thick and can support data transfer over longer distances than thinnet. Thicknet has a maximum cable length of 500 meters and usually is used as a backbone to connect several smaller thinnet-based networks.

Coaxial Cable Connectors

To connect coaxial cable to devices, we need coaxial connectors. The most common type of connector used today is the Bayonet Neill–Concelman (BNC) connector. The figure below shows three popular types of these connectors: the BNC connector, the BNC T connector, and the BNC terminator.



- The BNC connector is used to connect the end of the cable to a device, such as a TV set.
- The BNC T connector is used in Ethernet networks to branch out to a connection to a computer or other device.
- The BNC terminator is used at the end of the cable to prevent the reflection of the signal.

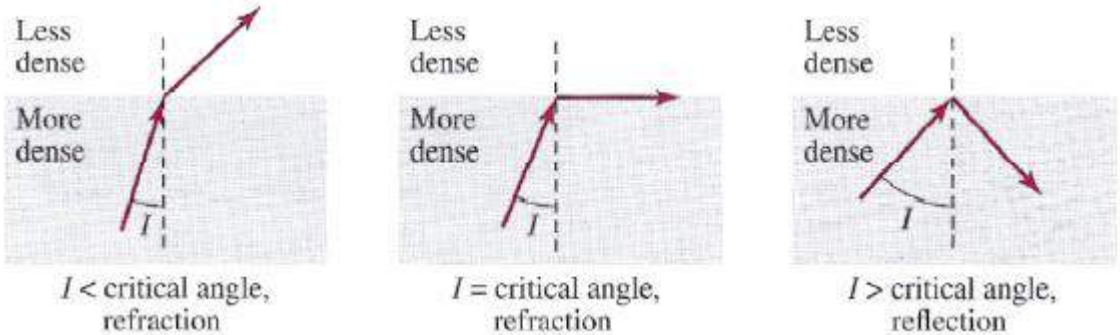
Applications

- Analog telephone networks.
- Cable TV networks.
- Traditional Ethernet LANs.

3- Fiber-Optic Cable

A fiber-optic cable is made of glass or plastic and transmits signals in the form of light. To understand optical fiber, we first need to explore several aspects of the nature of light.

Light travels in a straight line as long as it is moving through a single uniform substance. If a ray of light traveling through one substance suddenly enters another substance (of a different density), the ray changes direction. The figure below shows how a ray of light changes direction when going from a more dense to a less dense substance.

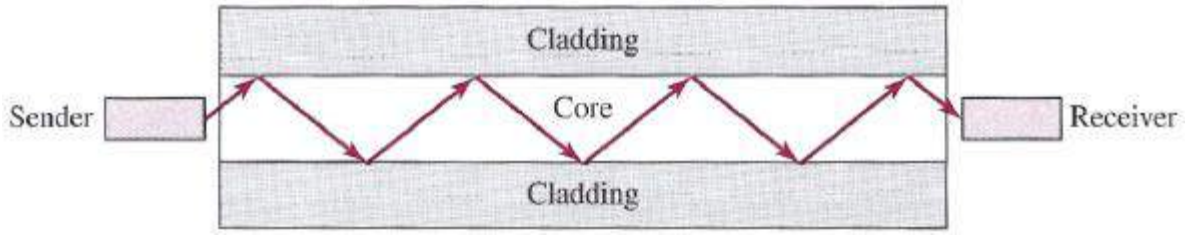


As the figure shows:

- If the angle of incidence I (the angle the ray makes with the line perpendicular to the interface between the two substances) is less than the critical angle, the ray refracts and moves closer to the surface.
- If the angle of incidence is equal to the critical angle, the light bends along the interface.
- If the angle is greater than the critical angle, the ray reflects (makes a turn) and travels again in the denser substance.

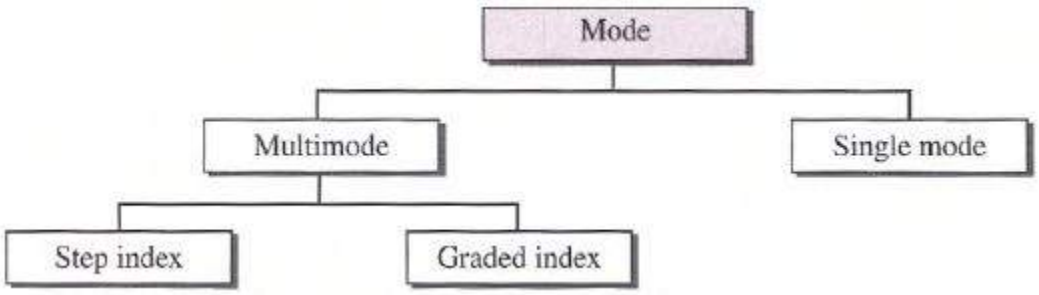
Note that the critical angle is a property of the substance, and its value differs from one substance to another.

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. The difference in density of the two materials must be such that a beam of light moving through the core is reflected off the cladding instead of being refracted into it. See the figure below.



Propagation Modes

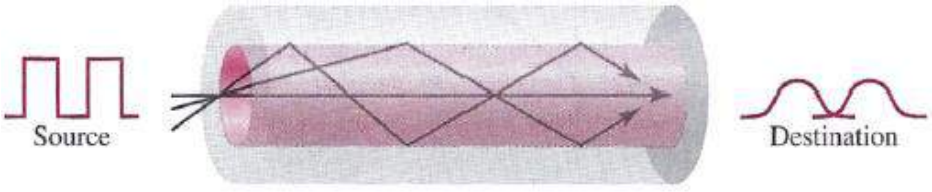
Current technology supports two modes (multimode and single mode) for propagating light along optical channels, each requiring fiber with different physical characteristics. Multimode can be implemented in two forms: step-index or graded-index.



Multimode

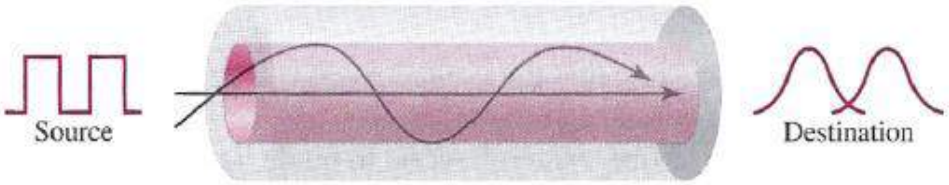
Multimode is so named because multiple beams from a light source move through the core in different paths. How these beams move within the cable depends on the structure of the core.

In multimode step-index fiber, the density of the core remains constant from the center to the edges. A beam of light moves through this constant density in a straight line until it reaches the interface of the core and the cladding. At the interface, there is an abrupt change due to a lower density; this alters the angle of the beam's motion. 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.



Multimode, step index

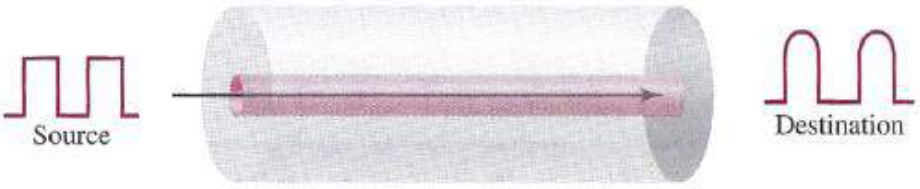
A second type of fiber, called multimode graded-index fiber, decreases this distortion of the signal through the cable. The word index here refers to the index of refraction. As we saw above, the index of refraction is related to density. A graded index fiber, therefore, is one with varying densities. Density is highest at the center of the core and decreases gradually to its lowest at the edge. The figure below shows the impact of this variable density on the propagation of light beams.



Multimode, graded index

Single-Mode

Single-mode uses step-index fiber and a highly focused source of light that limits beams to a small range of angles, all close to the horizontal. The single-mode fiber itself is manufactured with a much smaller diameter than that of multimode fiber, and with substantially lower density (index of refraction). The decrease in density results in a critical angle that is close enough to $90°$ to make the propagation of beams almost horizontal. In this case, propagation of different beams is almost identical, and delays are negligible. All the beams arrive at the destination "together" and can be recombined with little distortion to the signal.



Single mode

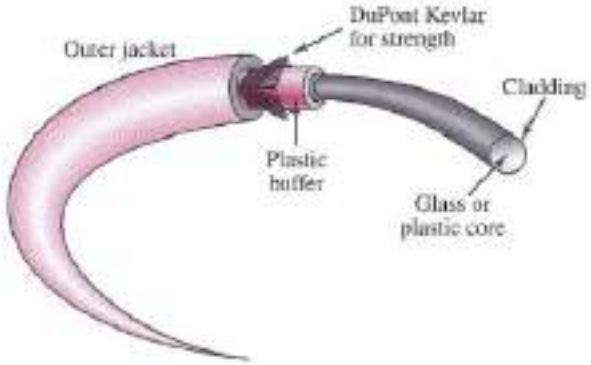
Fiber Sizes

Optical fibers are defined by the ratio of the diameter of their core to the diameter of their cladding, both expressed in micrometers. The common sizes are shown in the table below. Note that the last size listed is for single-mode only.

Type	Core (μm)	Cladding (μm)	Mode
50/125	50	125	Multimode, graded index
62.5/125	62.5	125	Multimode, graded index
100/125	100	125	Multimode, graded index
7/125	7	125	Single mode

Cable Composition

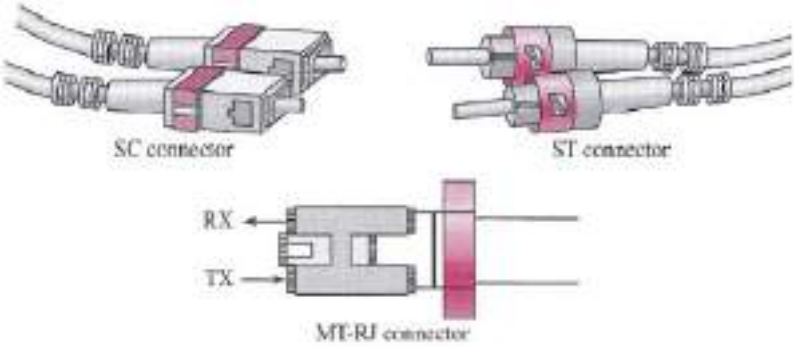
The figure below shows the composition of a typical fiber-optic cable. The outer jacket is made of either PVC or Teflon. Inside the jacket are Kevlar strands to strengthen the cable. Kevlar is a strong material used in the fabrication of bulletproof vests. Below the Kevlar is another plastic coating to cushion the fiber. The fiber is at the center of the cable, and it consists of cladding and core.



Fiber-Optic Cable Connectors

There are three types of connectors for fiber-optic cables, as shown in the figure below.

- The subscriber channel (SC) connector is used for cable TV. It uses a push/pull locking system.
- The straight-tip (ST) connector is used for connecting cable to networking devices. It uses a bayonet locking system and is more reliable than SC.
- MT-RJ is a connector that is the same size as RJ45.



Applications

- The SONET network.
- Cable TV.
- Local-area networks such as 100Base-FX network (Fast Ethernet) and 1000Base-X.

Advantages of Optical Fiber

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

- Higher bandwidth
- Less signal attenuation
- Immunity to electromagnetic interference.
- Resistance to corrosive materials.
- Light weight
- Greater immunity to tapping

Disadvantages of Optical Fiber

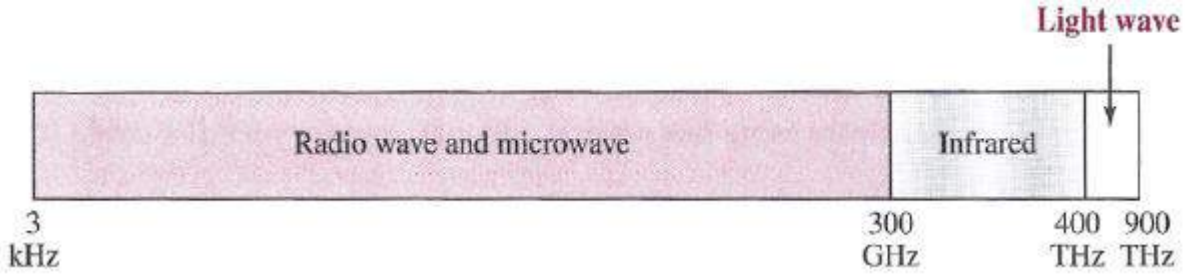
There are some disadvantages in the use of optical fiber.

- Installation and maintenance: Fiber-optic cable is a relatively new technology. Its installation and maintenance require expertise that is not yet available everywhere.
- Unidirectional light propagation: Propagation of light is unidirectional. If we need bidirectional communication, two fibers are needed.
- 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 medium transport electromagnetic waves without using a physical conductor. This type of communication is often referred to as wireless communication. Signals are normally broadcast through free space and thus are available to anyone who has a device capable of receiving them.

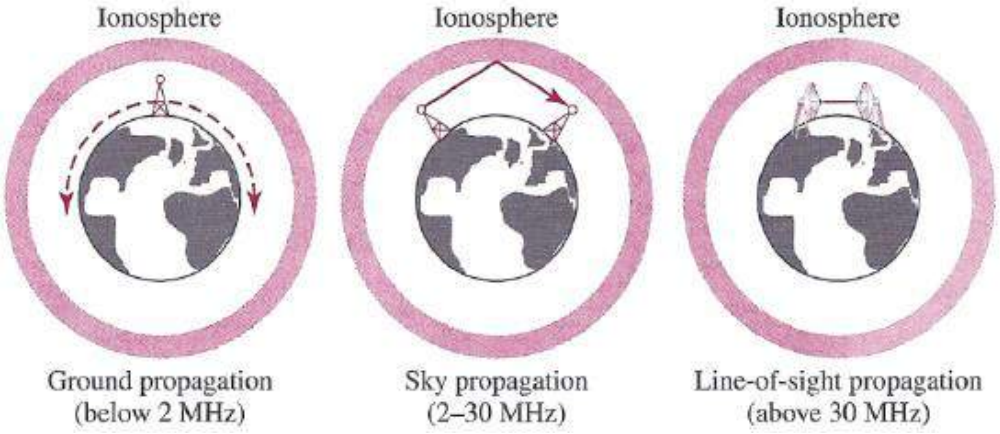
The figure below shows the part of the electromagnetic spectrum, ranging from 3 kHz to 900 THz, used for wireless communication.



Computer Networks

Unguided signals can travel from the source to the destination in several ways:

- Ground propagation
- Sky propagation
- Line-of-sight propagation



In ground propagation, radio waves travel through the lowest portion of the atmosphere, hugging the earth. These low-frequency signals emanate in all directions from the transmitting antenna and follow the curvature of the planet. Distance depends on the amount of power in the signal: The greater the power, the greater the distance.

In sky propagation, higher-frequency radio waves radiate upward into the ionosphere (the layer of atmosphere where particles exist as ions) where they are reflected back to earth. This type of transmission allows for greater distances with lower output power.

In line-of-sight propagation, very high-frequency signals are transmitted in straight lines directly from antenna to antenna. Antennas must be directional, facing each other, and either tall enough or close enough together not to be affected by the curvature of the earth. Line-of-sight propagation is tricky because radio transmissions cannot be completely focused.

The section of the electromagnetic spectrum defined as radio waves and microwaves is divided into eight ranges, called bands, each regulated by government authorities. These bands are rated from very low frequency (VLF) to extremely high frequency (EHF). The table below lists these bands, their ranges, propagation methods, and some applications.

Band	Range	Propagation	Application
very low frequency (VLF)	3–30 kHz	Ground	Long-range radio navigation
low frequency (LF)	30–300 kHz	Ground	Radio beacons and navigational locators
middle frequency (MF)	300 kHz–3 MHz	Sky	AM radio
high frequency (HF)	3–30 MHz	Sky	Citizens band (CB), ship/ aircraft
very high frequency (VHF)	30–300 MHz	Sky and line-of-sight	VHF TV, FM radio
Ultrahigh frequency (UHF)	300 MHz–3 GHz	Line-of-sight	UHF TV, cellular phones, paging, satellite
Superhigh frequency (SF)	3–30 GHz	Line-of-sight	Satellite
Extremely high frequency(EHF)	30–300GHz	Line-of-sight	Radar, satellite

,

We can divide wireless transmission into three broad groups: radio waves, microwaves, and infrared waves.

1- Radio Waves

Although there is no clear-cut demarcation between radio waves and microwaves, electromagnetic waves ranging in frequencies between 3 kHz and 1 GHz are normally called radio waves; waves ranging in frequencies between 1 and 300 GHz are called microwaves.

However, the behavior of the waves, rather than the frequencies, is a better criterion for classification. Radio waves, for the most part, are omnidirectional. 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 omnidirectional 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.

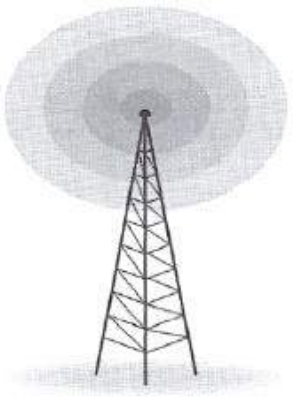
Radio waves, particularly those waves that propagate in the sky mode, can travel long distances. This makes radio waves a good candidate for long-distance broadcasting such as AM radio. Radio waves, particularly those of low and medium frequencies, can penetrate walls. This characteristic can be both an advantage and a disadvantage. It is an advantage because, for example, an AM radio can receive signals inside a building. It is a disadvantage because we cannot isolate a communication to just inside or outside a building.

The radio wave band is relatively narrow, just under 1 GHz, compared to the microwave band. When this band is divided into sub-bands, the sub-bands are also narrow, leading to a low data rate for

digital communications. Almost the entire band is regulated by authorities (e.g., the FCC in the United States). Using any part of the band requires permission from the authorities.

Omnidirectional 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. The figure below shows an omnidirectional antenna.



Applications

The omnidirectional 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.

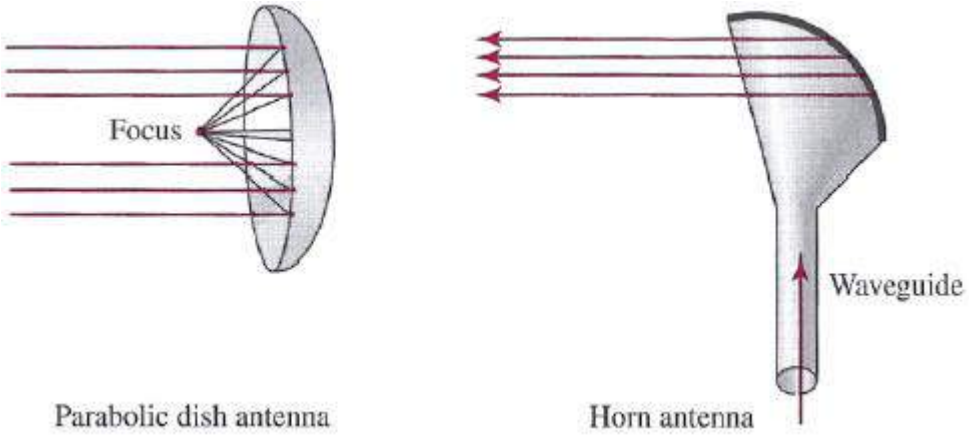
2- Microwaves

Electromagnetic waves having frequencies between 1 and 300 GHz are called microwaves. Microwaves are unidirectional. When an antenna transmits microwaves, they can be narrowly focused. This means that 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. The following describes some characteristics of microwave propagation:

- Microwave propagation is line-of-sight. Since the towers with the mounted antennas need to be in direct sight of each other, towers that are far apart need to be very tall. The curvature of the earth as well as other blocking obstacles do not allow two short towers to communicate by using microwaves. Repeaters are often needed for long distance communication.
- Very high-frequency microwaves cannot penetrate walls. This characteristic can be a disadvantage if receivers are inside buildings.
- The microwave band is relatively wide, almost 299 GHz. Therefore, wider sub-bands can be assigned, and a high data rate is possible. Use of certain portions of the band requires permission from authorities.

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 parabolic dish antenna is based on the geometry of a parabola: Every line parallel to the line of symmetry (line of sight) reflects off the curve at angles such that all the lines intersect in a common point called the focus. The parabolic dish works as a funnel, catching a wide range of waves and directing them to a common point. In this way, more of the signal is recovered than would be possible with a single-point receiver.

Outgoing transmissions are broadcast through a horn aimed at the dish. The microwaves hit the dish and are deflected outward in a reversal of the receipt path. A horn antenna looks like a gigantic scoop. Outgoing transmissions are broadcast up a stem (resembling a handle) and deflected outward in a series of narrow parallel beams by the curved head. Received transmissions are collected by the scooped shape of the horn, in a manner similar to the parabolic dish, and are deflected down into the stem.

Applications

Microwaves, due to their unidirectional properties, are very useful when unicast (one-to-one) communication is needed between the sender and the receiver. They are used in cellular phones, satellite networks, and wireless LANs.

3- Infrared

Infrared waves, with frequencies from 300 GHz to 400 THz (wavelengths from 1 mm to 770 nrn), 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 of the remote by our

neighbors. However, this same characteristic makes 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

The infrared band, almost 400 THz, has an excellent potential for data transmission. Such a wide bandwidth can be used to transmit digital data with a very high data rate. The Infrared Data Association (IrDA), an association for sponsoring the use of infrared waves, has established standards for using these signals for communication between devices such as keyboards, mice, PCs, and printers. For example, some manufacturers provide a special port called the IrDA port that allows a wireless keyboard to commnicate with a PC. The standard originally defined a data rate of 75 kbps for a distance up to 8 m. The recent standard defines a data rate of 4 Mbps. Infrared signals defined by IrDA transmit through line of sight; the IrDA port on the keyboard needs to point to the PC for transmission to occur.

Computer Networks

MULTIPLEXING

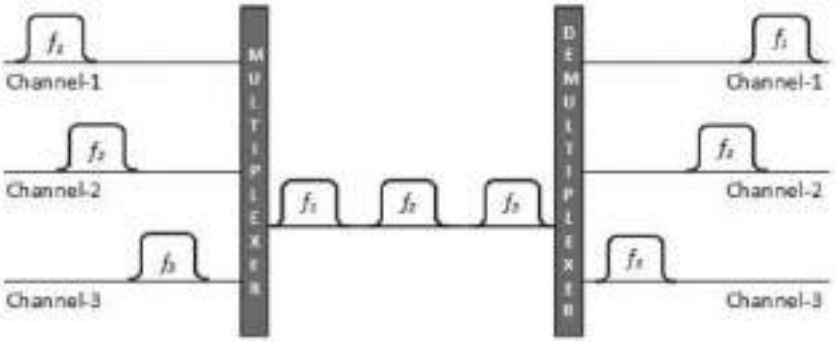
Multiplexing is a technique by which different analog and digital streams of transmission can be simultaneously processed over a shared link. Multiplexing divides the high capacity medium into low capacity logical medium which is then shared by different streams.

Communication is possible over the air (radio frequency), using a physical media (cable), and light (optical fiber). All mediums are capable of multiplexing.

When multiple senders try to send over a single medium, a device called Multiplexer divides the physical channel and allocates one to each. On the other end of communication, a De-multiplexer receives data from a single medium, identifies each, and sends to different receivers.

1- Frequency Division Multiplexing

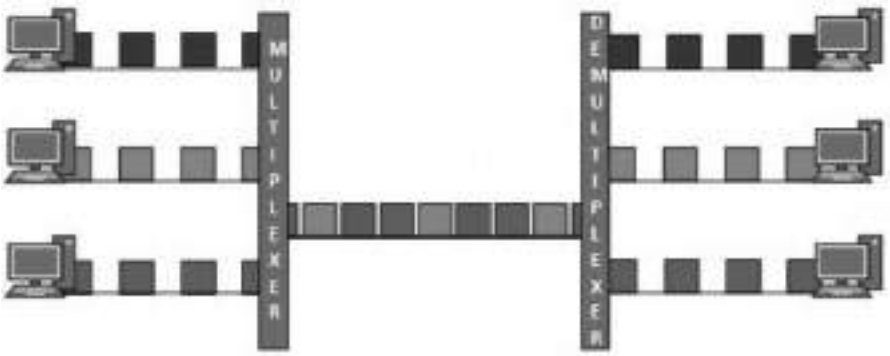
When the carrier is frequency, FDM is used. FDM is an analog technology. FDM divides the spectrum or carrier bandwidth in logical channels and allocates one user to each channel. Each user can use the channel frequency independently and has exclusive access of it. All channels are divided in such a way that they do not overlap with each other. Channels are separated by guard bands. Guard band is a frequency which is not used by either channel.



2- Time Division Multiplexing

TDM is applied primarily on digital signals but can be applied on analog signals as well. In TDM the shared channel is divided among its user by means of time slot. Each user can transmit data within the provided time slot only. Digital signals are divided in frames, equivalent to time slot i.e. frame of an optimal size which can be transmitted in given time slot.

TDM works in synchronized mode. Both ends, i.e. Multiplexer and De-multiplexer are timely synchronized, and both switch to next channel simultaneously.



When channel A transmits its frame at one end, the De-multiplexer provides media to channel A on the other end. As soon as the channel A's time slot expires, this side switches to channel B. On the other end, the De-multiplexer works in a synchronized manner and provides media to channel B. Signals from different channels travel the path in interleaved manner.

3- Wavelength Division Multiplexing

Light has different wavelength (colors). In fiber optic mode, multiple optical carrier signals are multiplexed into an optical fiber by using different wavelengths. This is an analog multiplexing technique and is done conceptually in the same manner as FDM but uses light as signals.

Further, on each wavelength time division multiplexing can be incorporated to accommodate more data signals.

4- Code Division Multiplexing

Multiple data signals can be transmitted over a single frequency by using Code Division Multiplexing. FDM divides the frequency in smaller channels but CDM allows its users to full bandwidth and transmit signals all the time using a unique code. CDM uses orthogonal codes to spread signals.

Each station is assigned with a unique code, called chip. Signals travel with these codes independently, inside the whole bandwidth. The receiver knows in advance the chip code signal it has to receive.

SWITCHING

Switching is process to forward packets coming in from one port to a port leading towards the destination. When data comes on a port it is called ingress, and when data leaves a port or goes out it is called egress. A communication system may include number of switches and nodes. At broad level, switching can be divided into two major categories:

1- Connectionless: The data is forwarded on behalf of forwarding tables. No previous handshaking is required and acknowledgements are optional.

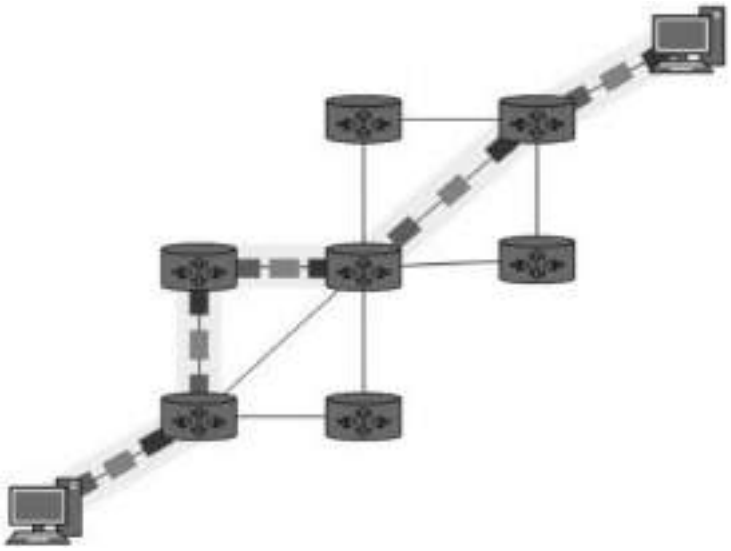
2- Connection Oriented: Before switching data to be forwarded to destination, there is a need to pre-establish circuit along the path between both endpoints. Data is then forwarded on that circuit. After the transfer is completed, circuits can be kept for future use or can be turned down immediately.

1- Circuit Switching

When two nodes communicate with each other over a dedicated communication path, it is called circuit switching. There is a need of pre-specified route from which data travels and no other data is permitted. In circuit switching to transfer the data, circuit must be established so that the data transfer can take place.

Circuits can be permanent or temporary. Applications which use circuit switching may have to go through three phases:

1. Establish a circuit
2. Transfer the data
3. Disconnect the circuit

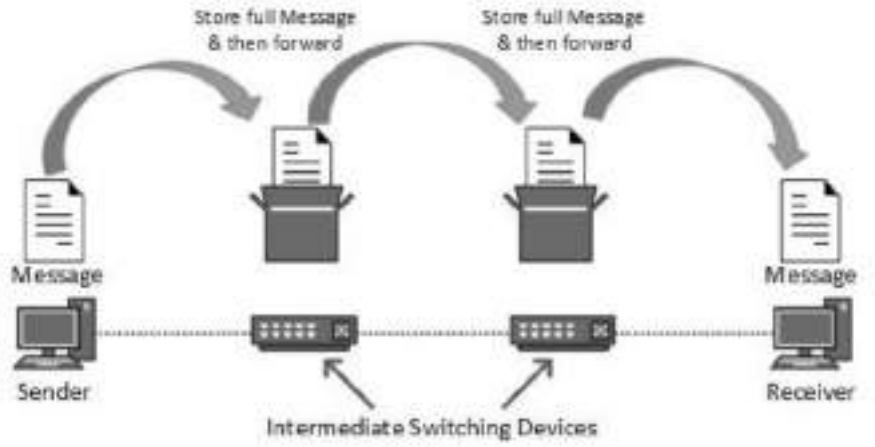


Circuit switching was designed for voice applications. Telephone is the best suitable example of circuit switching. Before a user can make a call, a virtual path between caller and callee is established over the network.

2- Message Switching

This technique was somewhere in middle of circuit switching and packet switching. In message switching, the whole message is treated as a data unit and is switching / transferred in its entirety.

A switch working on message switching, first receives the whole message and buffers it until there are resources available to transfer it to the next hop. If the next hop is not having enough resource to accommodate large size message, the message is stored and switch waits.



This technique was considered substitute to circuit switching. As in circuit switching the whole path is blocked for two entities only. Message switching is replaced by packet switching. Message switching has the following drawbacks:

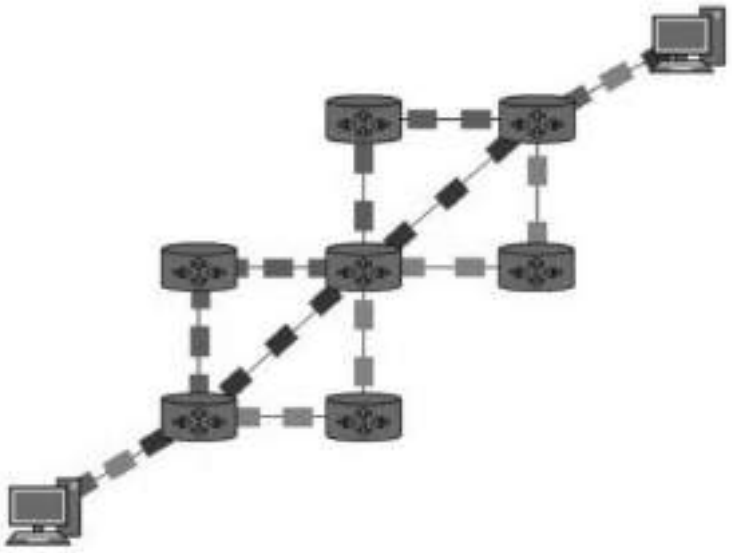
- Every switch in transit path needs enough storage to accommodate entire message.
- Because of store-and-forward technique and waits included until resources are available, message switching is very slow.
- Message switching was not a solution for streaming media ((audio, video) and real-time applications.

Computer Networks

3- Packet Switching

Shortcomings of message switching gave birth to an idea of packet switching. The entire message is broken down into smaller chunks called packets. The switching information is added in the header of each packet and transmitted independently.

It is easier for intermediate networking devices to store small size packets and they do not take much resources either on carrier path or in the internal memory of switches.



Packet switching enhances line efficiency as packets from multiple applications can be multiplexed over the carrier. The internet uses packet switching technique. Packet switching enables the user to differentiate data streams based on priorities. Packets are stored and forwarded according to their priority to provide quality of service.

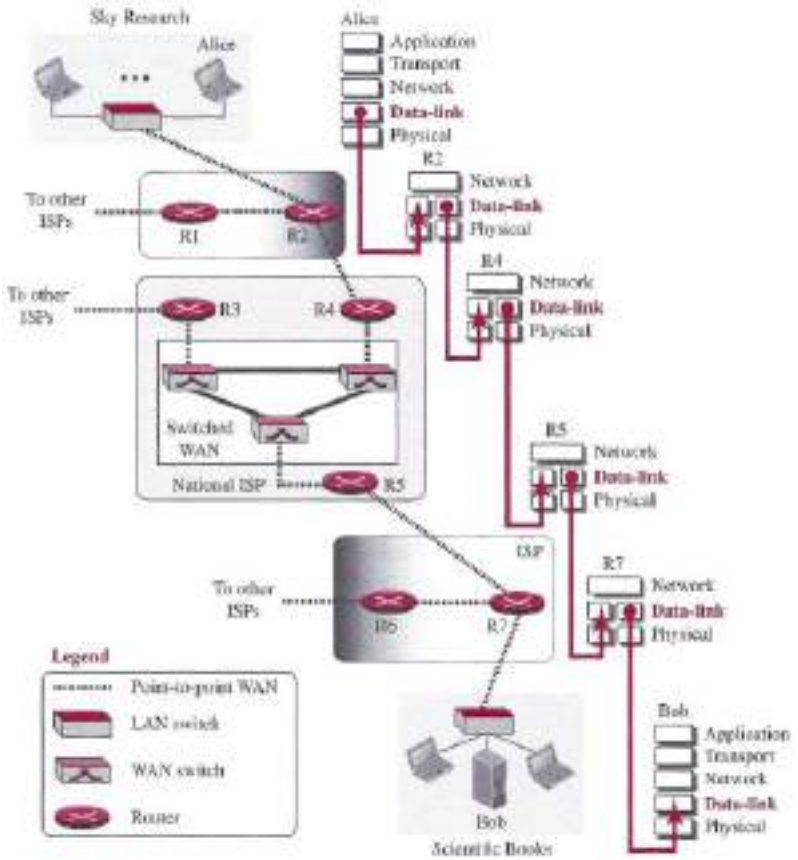
Computer Networks

Data link layer

Data Link Layer is second layer of TCP/IP model. This layer is one of the most complicated layers and has complex functionalities and liabilities. Data link layer hides the details of underlying hardware and represents itself to upper layer as the medium to communicate.

Data link layer works between two hosts which are directly connected in some sense. This direct connection could be point to point or broadcast.

The figure below shows the same scenario we discussed previously, but we are now interested in communication at the data-link layer. Communication at the data-link layer is made up of five separate logical connections between the data-link layers in the path.

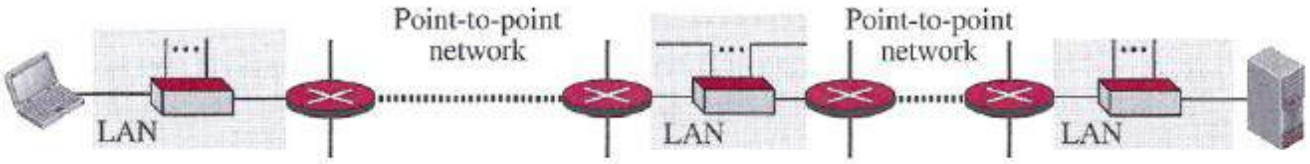


The data-link layer at Alice's computer communicates with the data-link layer at router R2. The data-link layer at router R2 communicates with the data-link layer at router R4, and so on. Finally, the data-link layer at router R7 communicates with the data-link layer at Bob's computer. Only one data-link layer is involved at the source or the destination, but two data-link layers are involved at each router. The reason is that Alice's and Bob's computers are each connected to a single network, but each router takes input from one network and sends output to another network. Note that although switches are also involved in the data-link-layer communication, for simplicity we have not shown them in the figure.

Computer Networks

Nodes and Links

Communication at the data-link layer is node-to-node. A data unit from one point in the Internet needs to pass through many networks (LANs and WANs) to reach another point. These LANs and WANs are connected by routers. It is customary to refer to the two end hosts and the routers as nodes and the networks in between as links. The figure below is a simple representation of links and nodes when the path of the data unit is only six nodes.



A small part of the Internet

Nodes and links

The first node is the source host; the last node is the destination host. The other four nodes are four routers. The first, the third, and the fifth links represent the three LANs; the second and the fourth links represent the two WANs.

Services

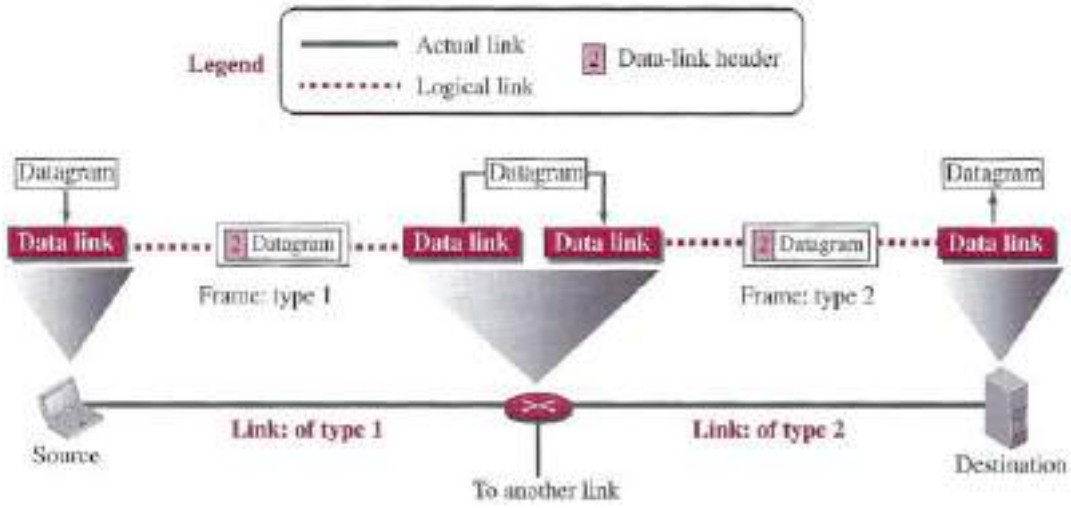
The data-link layer is located between the physical and the network layers. The datalink layer provides services to the network layer; it receives services from the physical layer. Let us discuss services provided by the data-link layer.

The duty scope of the data-link layer is node-to-node. When a packet is travelling in the Internet, the data-link layer of a node (host or router) is responsible for delivering a datagram to the next node in the path. For this purpose, the data-link layer of the sending node needs to encapsulate the datagram received from the network in a frame, and the data-link layer of the receiving node needs to decapsulate the datagram from the frame. In other words, the data-link layer of the source host needs only to encapsulate, the data-link layer of the destination host needs to decapsulate, but each intermediate node needs to both encapsulate and decapsulate. One may ask why we need encapsulation and decapsulation at each intermediate node. The reason is that each link may be using

a different protocol with a different frame format. Even if one link and the next are using the same protocol, encapsulation and decapsulation are needed because the link-layer addresses are normally different.

An analogy may help in this case. Assume a person needs to travel from her home to her friend's home in another city. The traveler can use three transportation tools. She can take a taxi to go to the train station in her own city, then travel on the train from her own city to the city where her friend lives, and finally reach her friend's home using another taxi. Here we have a source node, a destination node, and two intermediate nodes. The traveler needs to get into the taxi at the source node, get out of the taxi and get into the train at the first intermediate node (train station in the city where she lives), get out of the train and get into another taxi at the second intermediate node (train station in the city where her friend lives), and finally get out of the taxi when she arrives at her destination. A kind of encapsulation occurs at the source node, encapsulation and decapsulation occur at the intermediate nodes, and decapsulation occurs at the destination node. Our traveler is the same, but she uses three transporting tools to reach the destination.

The figure below shows the encapsulation and decapsulation at the data-link layer. For simplicity, we have assumed that we have only one router between the source and destination. The datagram received by the data-link layer of the source host is encapsulated in a frame. The frame is logically transported from the source host to the router. The frame is decapsulated at the data-link layer of the router and encapsulated at another frame. The new frame is logically transported from the router to the destination host. Note that, although we have shown only two data-link layers at the router, the router actually has three data-link layers because it is connected to three physical links.



With the contents of the above figure in mind, we can list the services provided by a data-link layer as shown below.

1- Framing

Definitely, the first service provided by the data-link layer is framing. The data-link layer at each node needs to encapsulate the datagram (packet received from the network layer) in a frame before sending it to the next node. The node also needs to decapsulate the datagram from the frame received on the logical channel. Although we have shown only a header for a frame, we will see in future chapters that a frame may have both a header and a trailer. Different data-link layers have different formats for framing.

2- Flow Control

Whenever we have a producer and a consumer, we need to think about flow control. If the producer produces items that cannot be consumed, accumulation of items occurs. The sending data-link layer at the end of a link is a producer of frames; the receiving data-link layer at the other end of a link is a consumer. If the rate of produced frames is higher than the rate of consumed frames, frames at the receiving end need to be buffered while waiting to be consumed (processed). Definitely, we cannot have an unlimited buffer size at the receiving side. We have two choices. The first choice is to let the receiving data-link layer drop the frames if its buffer is full. The second choice is to let the receiving data-link layer send a feedback to the sending data-link layer to ask it to stop or slow down. Different data-link-layer protocols use different strategies for flow control. Since flow control also occurs at the transport layer, with a higher degree of importance.

3- Error Control

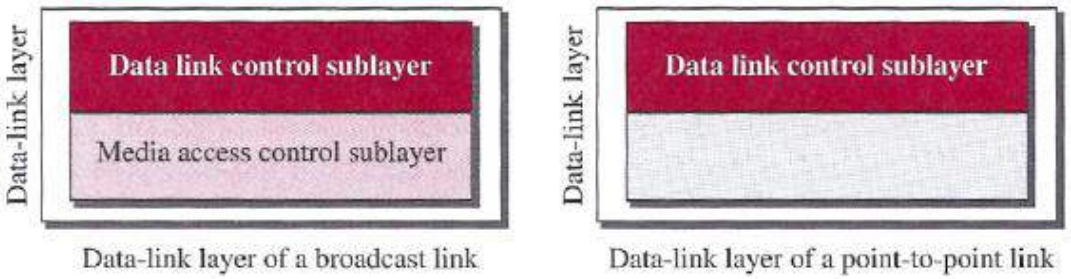
At the sending node, a frame in a data-link layer needs to be changed to bits, transformed to electromagnetic signals, and transmitted through the transmission media. At the receiving node, electromagnetic signals are received, transformed to bits, and put together to create a frame. Since electromagnetic signals are susceptible to error, a frame is susceptible to error. The error needs first to be detected. After detection, it needs to be either corrected at the receiver node or discarded and retransmitted by the sending node. Since error detection and correction is an issue in every layer (node-to-node or host-to-host).

Two Categories of Links

Although two nodes are physically connected by a transmission medium such as cable or air, we need to remember that the data-link layer controls how the medium is used. We can have a data-link layer that uses the whole capacity of the medium; we can also have a data-link layer that uses only part of the capacity of the link. In other words, we can have a **point-to-point link or a broadcast link**. In a point-to-point link, the link is dedicated to the two devices; in a broadcast link, the link is shared between several pairs of devices. For example, when two friends use the traditional home phones to chat, they are using a point-to-point link; when the same two friends use their cellular phones, they are using a broadcast link (the air is shared among many cell phone users).

Two Sublayers

To better understand the functionality of and the services provided by the link layer, we can divide the data-link layer into two sublayers: data link control (DLC) and media access control (MAC). The data link control sublayer deals with all issues common to both point-to-point and broadcast links; the media access control sublayer deals only with issues specific to broadcast links. In other words, we separate these two types of links at the data-link layer, as shown in the figure below.



Data-link layer of a broadcast link Data-link layer of a point-to-point link

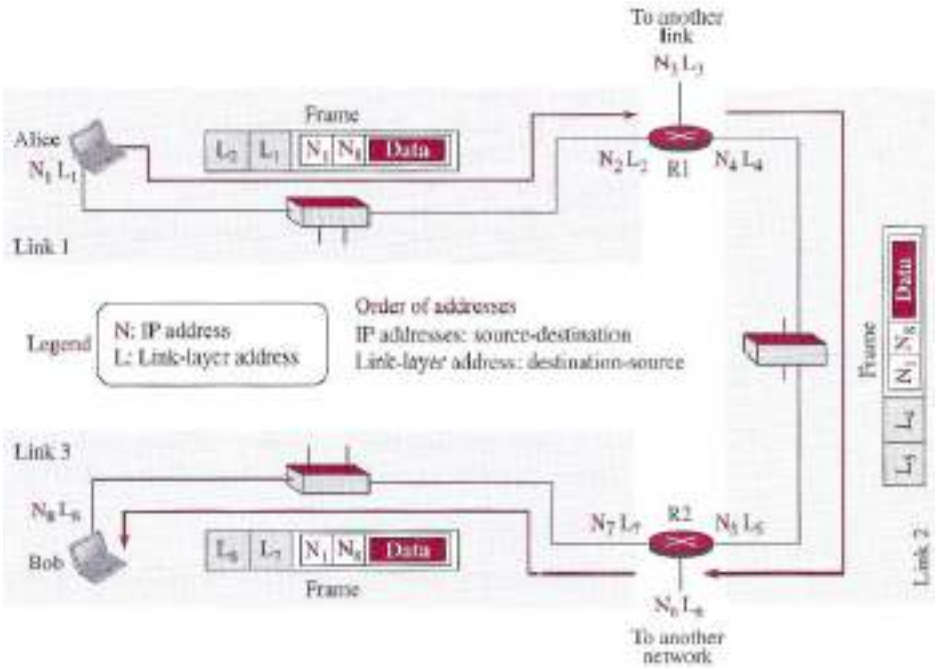
LINK-LAYER ADDRESSING

The next issue we need to discuss about the data-link layer is the link-layer addresses. We will discuss later IP addresses as the identifiers at the network layer that define the exact points in the Internet where the source and destination hosts are connected. However, in a connectionless internetwork such as the Internet we cannot make a datagram reach its destination using only IP addresses. The reason is that each datagram in the Internet, from the same source host to the same destination host, may take a different path. The source and destination IP addresses define the two ends but cannot define which links the datagram should pass through.

We need to remember that the IP addresses in a datagram should not be changed. If the destination IP address in a datagram changes, the packet never reaches its destination; if the source IP address in a datagram changes, the destination host or a router can never communicate with the source if a response needs to be sent back or an error needs to be reported back to the source.

The above discussion shows that we need another addressing mechanism in a connectionless internetwork: the link-layer addresses of the two nodes. A link-layer address is sometimes called a link address, sometimes a physical address, and sometimes a MAC address.

Since a link is controlled at the data-link layer, the addresses need to belong to the data-link layer. When a datagram passes from the network layer to the data-link layer, the datagram will be encapsulated in a frame and two data-link addresses are added to the frame header. These two addresses are changed every time the frame moves from one link to another. The figure below demonstrates the concept in a small internet.



Computer Networks

In the internet in the figure, we have three links and two routers. We also have shown only two hosts: Alice (source) and Bob (destination). For each host, we have shown two addresses, the IP addresses (N) and the link-layer addresses (L). Note that a router has as many pairs of addresses as the number of links the router is connected to. We have shown three frames, one in each link. Each frame carries the same datagram with the same source and destination addresses (N1 and N8), but the link-layer addresses of the frame change from link to link. In link 1, the link-layer addresses are L1 and L2. In link 2, they are L4 and L5. In link 3, they are L7 and Ls. Note that the IP addresses and the link-layer addresses are not in the same order. For IP addresses, the source address comes before the destination address; for link-layer addresses, the destination address comes before the source. The datagrams and frames are designed in this way, and we follow the design. We may raise several questions:

- If the IP address of a router does not appear in any datagram sent from a source to a destination, why do we need to assign IP addresses to routers?

  The answer is that in some protocols a router may act as a sender or receiver of a datagram. For example, in routing protocols, a router is a sender or a receiver of a message. The communications in these protocols are between routers.
- Why do we need more than one IP address in a router, one for each interface?

  The answer is that an interface is a connection of a router to a link. We will see that an IP address defines a point in the Internet at which a device is connected. A router with n interfaces is connected to the Internet at n points. This is the situation of a house at the corner of a street with two gates; each gate has the address related to the corresponding street.
- How are the source and destination IP addresses in a packet determined?

  The answer is that the host should know its own IP address, which becomes the source IP address in the packet. The application layer uses the services of DNS to find the destination address of the packet and passes it to the network layer to be inserted in the packet.
- How are the source and destination link-layer addresses determined for each link?

  Again, each hop (router or host) should know its own link-layer address. The destination link-layer address is determined by using the Address Resolution Protocol (ARP).

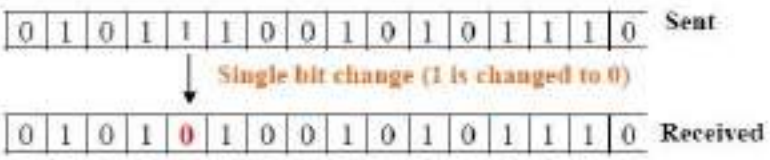
ERROR DETECTION AND CORRECTION

Networks must be able to transfer data from one device to another with acceptable accuracy. For most applications, a system must guarantee that the data received are identical to the data transmitted. Any time data are transmitted from one node to the next, they can become corrupted in passage. Many factors can alter one or more bits of a message. Some applications require a mechanism for detecting and correcting errors. Some applications can tolerate a small level of error, for example, random errors in audio or video transmissions may be tolerable, but when we transfer text, we expect a very high level of accuracy. At the data-link layer, if a frame is corrupted between the two nodes, it needs to be corrected before it continues its journey to other nodes. However, most link-layer protocols simply discard the frame and let the upper-layer protocols handle the retransmission of the frame. Some multimedia applications, however, try to correct the corrupted frame.

Types of Errors

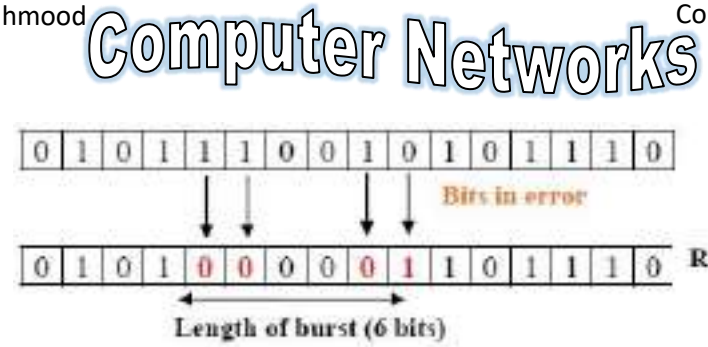
Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference. This interference can change the shape of the signal. These errors can be divided into two types:

A- Single-bit error: The term single-bit error means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1.



Single bit errors are least likely type of errors in serial data transmission. To see why, imagine a sender sends data at 10 Mbps. This means that each bit lasts only for 0.1 μsec (micro-second). For a single bit error to occur noise must have duration of only 0.1 μsec (micro-second), which is very rare. However, a single-bit error can happen if we are having a parallel data transmission. For example, if 16 wires are used to send all 16 bits of a word at the same time and one of the wires is noisy, one bit is corrupted in each word.

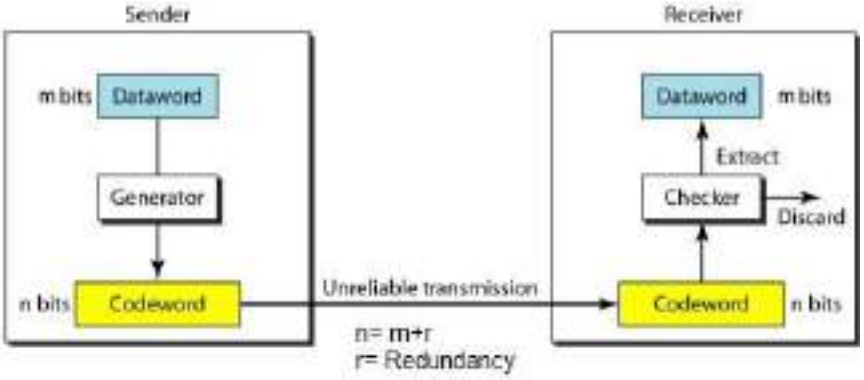
B- Burst error: The term burst error means that two or more bits in the data unit have changed from 0 to 1 or vice-versa. Note that burst error doesn't necessary means that error occurs in consecutive bits. The length of the burst error is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not be corrupted.



Burst errors are mostly likely to happen in serial transmission. The duration of the noise is normally longer than the duration of a single bit, which means that the noise affects data; it affects a set of bits as shown in figure above. The number of bits affected depends on the data rate and duration of noise.

Redundancy

The central concept in detecting or correcting errors is redundancy. To be able to detect or correct errors, we need to send some extra bits with our data. These redundant bits are added by the sender and removed by the receiver. Their presence allows the receiver to detect or correct corrupted bits.



Detection versus Correction

The correction of errors is more difficult than the detection.

- In error detection, we are only looking to see if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of corrupted bits. A single-bit error is the same for us as a burst error.
- In error correction, we need to know the exact number of bits that are corrupted and, more importantly, their location in the message. The number of errors and the size of the message are important factors. If we need to correct a single error in an 8-bit data unit, we need to consider eight possible error locations; if we need to correct two errors in a data unit of the same size, we need to consider 28 (permutation of 8 by 2) possibilities. You can imagine the receiver's difficulty in finding 10 errors in a data unit of 1000 bits.

Computer Networks

1- Error Detecting Codes

Popular techniques are:

A- Simple Parity check

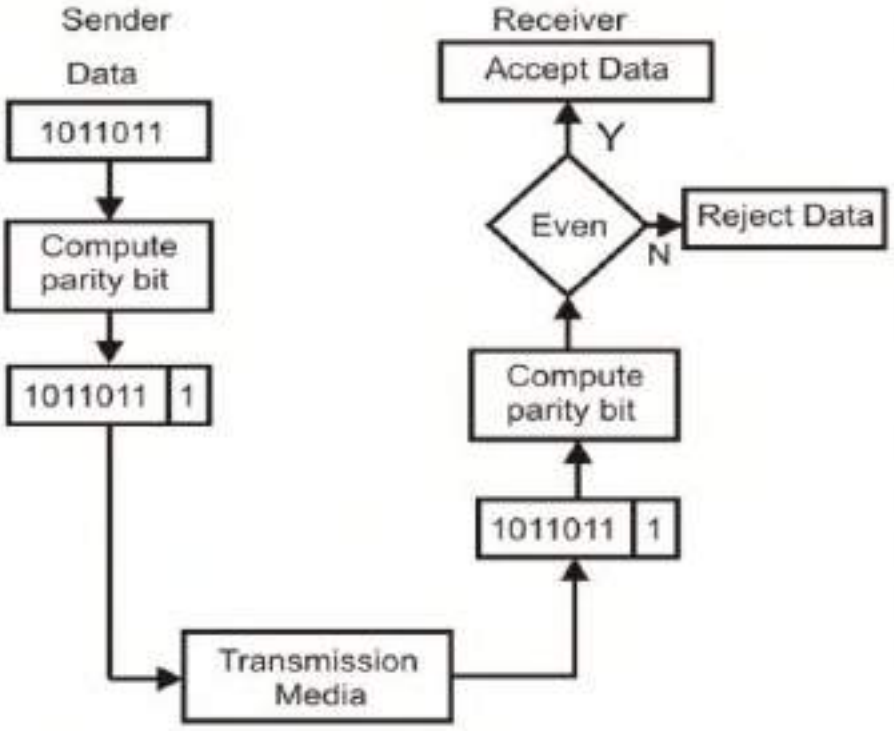
B- Two-dimensional Parity check

C- Checksum

D- Cyclic redundancy check

A- Simple Parity Checking or One-dimension Parity Check

The most common and least expensive mechanism for error- detection is the simple parity check. In this technique, a redundant bit called parity bit, is appended to every data unit so that the number of 1s in the unit (including the parity becomes) even. 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 an odd number of 1's and 0 is added if it contains an even number of 1's. At the receiving end the parity bit is computed from the received data bits and compared with the received parity bit, as shown in figure below. This scheme makes the total number of 1's even, that is why it is called even parity checking. Considering a 4-bit word, different combinations of the data words and the corresponding code words are given in table below.



Possible 4-bit data words and corresponding code words

Decimal value	Data Block	Parity bit	Code word
0	0000	0	00000
1	0001	1	00011
2	0010	1	00101
3	0011	0	00110
4	0100	1	01001
5	0101	0	01010
6	0110	0	01100
7	0111	1	01111
8	1000	1	10001
9	1001	0	10010
10	1010	0	10100
11	1011	1	10111
12	1100	0	11000
13	1101	1	11011
14	1110	1	11101
15	1111	0	11110

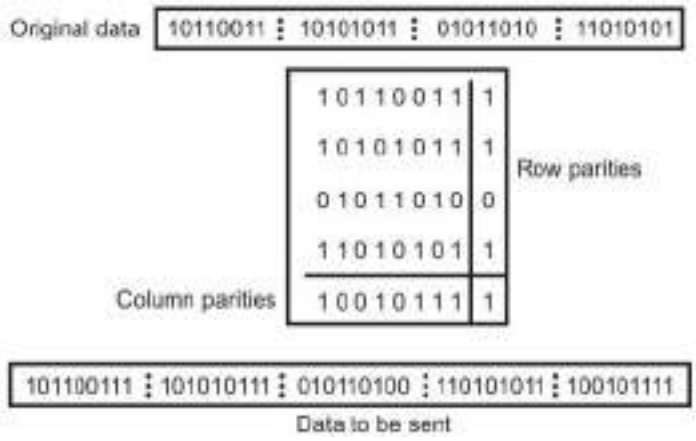
Note that for the sake of simplicity, we are discussing here the even-parity checking, where the number of 1's should be an even number. It is also possible to use odd-parity checking, where the number of 1's should be odd.

Performance:

If two errors occur in the code word, it becomes another valid member of the set and the decoder will see only another valid code word and know nothing of the error. Thus errors in more than one bit cannot be detected. In fact, it can be shown that a single parity check code can detect only odd number of errors in a code word.

B- Two-dimension Parity Check

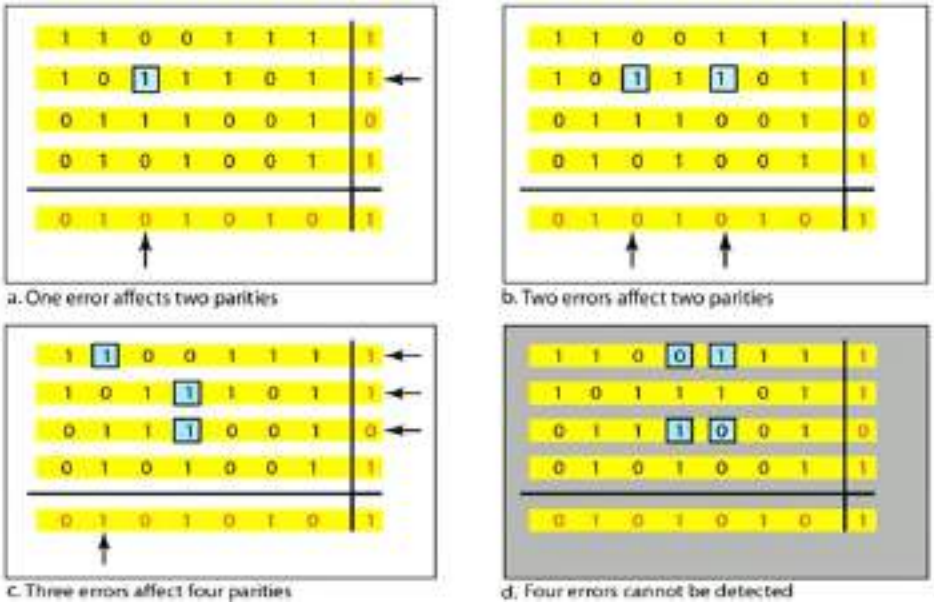
Performance can be improved by using two-dimensional parity check, which organizes the block of bits in the form of a table. Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are also calculated for all columns then both are sent along with the data. At the receiving end these are compared with the parity bits calculated on the received data. This is illustrated in figure below.



Performance

Two- Dimension Parity Checking increases the likelihood of detecting burst errors. As shown in figures (a, b and c) below that a 2-D Parity check of n bits can detect a burst error of n bits. A burst error of more than n bits is also detected by 2-D Parity check with a high-probability.

There is, however, one pattern of error that remains elusive. If two bits in one data unit are damaged and two bits in exactly same position in another data unit are also damaged, the 2-D Parity check checker will not detect an error as shown in figure d.



C- Checksum

In checksum error detection scheme, the data is divided into k segments each 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 as shown in example below.

Example: Suppose the following block of 32 bit is to be sent using a checksum of 8 bit: 10110011 10101011 01011010 11010101

```
k=4, m=8
10110011
10101011
--------
01011110
       1
--------
01011111
01011010
--------
10111001
11010101
--------
10001110
       1
--------
Sum :  10001111
Checksum 01110000
```

The pattern sent is: 10110011 10101011 01011010 11010101 01110000

At the receiver's end, all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented. If the result is zero, the received data is accepted; otherwise discarded, as shown below.

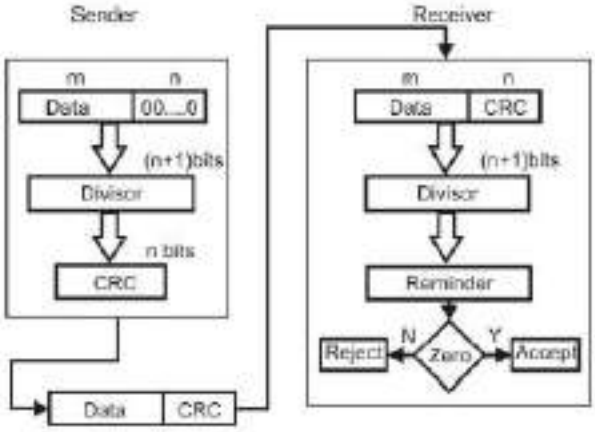
```
Received data
10110011
10101011
--------
01011110
       1
--------
01011111
01011010
--------
10111001
11010101
--------
10001110
       1
--------
10001111
01110000
--------
Sum:  11111111
Complement = 00000000
Conclusion = Accept data
```

Performance

The checksum detects all errors involving an odd number of bits. It also detects most errors involving even number of bits.

D- Cyclic Redundancy Checks (CRC)

This Cyclic Redundancy Check also known as a polynomial code is the most powerful and easy to implement technique. Unlike checksum scheme, which is based on addition, CRC is based on binary division. In CRC, a sequence of redundant bits, called **cyclic redundancy check bits**, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number. At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted. A remainder indicates that the data unit has been damaged in transit and therefore must be rejected. The generalized technique can be explained as shown in flowchart below.



Example: Obtain the 4-bit CRC code word for the data bit sequence 111011001 using the generator 10101?

$$
\begin{array}{r|l}
 & 110101001 \\
\hline
10101 & 1110110010000 \\
 & 10101 \\
\hline
 & 10001 \\
 & 10101 \\
\hline
 & 10000 \\
 & 10101 \\
\hline
 & 10110 \\
 & 10101 \\
\hline
 & 11000 \\
 & 10101 \\
\hline
 & 1101 = \mathbf{CRC}
\end{array}
$$

Transmitted frame: 1110110011101

2- Error Correcting Codes

The techniques that we have discussed so far can detect errors, but do not correct them. Error Correction can be handled in two ways.

- One is when an error is discovered; the receiver can have the sender retransmit the entire data unit. This is known as backward error correction.
- In the other, receiver can use an error-correcting code, which automatically corrects certain errors. This is known as forward error correction.

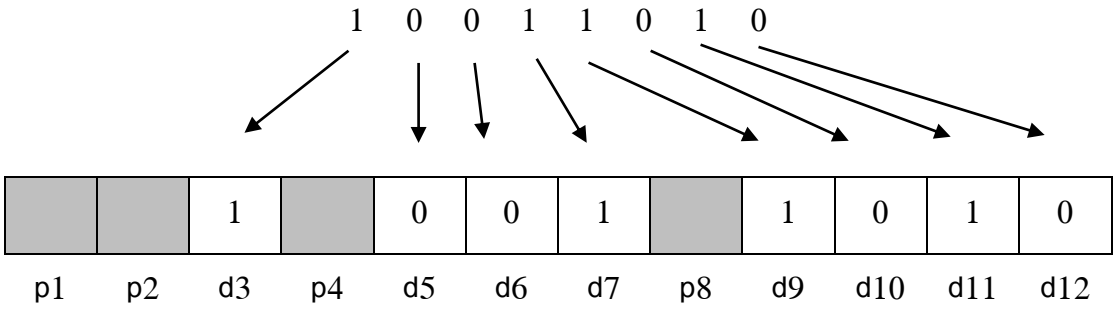
In theory it is possible to correct any number of errors automatically. Error-correcting codes are more sophisticated than error detecting codes and require more redundant bits. The number of bits required to correct multiple-bit or burst error is so high that in most of the cases it is inefficient to do so. For this reason, most error correction is limited to one, two or at the most three-bit errors.

Hamming Codes

The most common types of error-correcting codes used in RAM are based on the codes devised by R. W. Hamming. In the Hamming code, k parity bits are added to an n-bit data word, forming a new word of n + k bits. The bit positions are numbered in sequence from 1 to n + k. Those positions numbered with powers of two are reserved for the parity bits. The remaining bits are the data bits. The code can be used with words of any length.

Before giving the general characteristics of the Hamming code, we will illustrate its operation with a data word of eight bits. Consider, for example, the 8-bit data word 10011010.

Now we are going to create data including parity bits. All bit positions that are power of 2 is our parity bits (i.e. 1,2,4,8,16 … etc.)



Each parity bit calculates the parity for some of the bits in the code word.

The position of parity bit determines the sequence of bits that it alternately checks and skips

P1: check 1 bit, skip 1 bit, check 1 bit, skip 1 bit, etc. (1,3,5,7,9,11,13, 15, …)

		1		0	0	1		1	0	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

? 1 0 1 1 1

Set a parity bit to 1 if the total number of ones in the positions it checks is odd.

Set a parity bit to 0 if the total number of ones in the positions it checks is even.

Here Four 1's (Even), so set P1 to 0

0		1		0	0	1		1	0	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

P2: check 2 bit, skip 2 bit, check 2 bit, skip 2 bit, etc. (2,3,6,7,10, 11, …)

0		1		0	0	1		1	0	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

? 1 0 1 0 1

Here Three 1's (Odd), so set P2 to 1

0	1	1		0	0	1		1	0	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

P4: check 4 bit, skip 4 bit, check 4 bit, skip 4 bit, etc. (4,5,6,7,12,13,14,15, …)

0	1	1		0	0	1		1	0	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

? 0 0 1 0

Here one 1's (Odd), so set P4 to 1

0	1	1	1	0	0	1		1	0	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

P8: check 8 bit, skip 8 bit, check 8 bit, skip 8 bit, etc. (8–15,24–31,40–47, …)

0	1	1	1	0	0	1		1	0	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

? 1 0 1 0

Here tow 1's (Even), so set P4 to 0

0	1	1	1	0	0	1	0	1	0	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

So, transmitted frame: 011100101010

For example, consider the following case will happen at receiver:

Error

0	1	1	1	0	0	1	0	1	1	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

Now we are going to find the bad bit:

P1: check 1 bit, skip 1 bit, check 1 bit, skip 1 bit, etc. (1,3,5,7,9,11,13, 15, …)

0	1	1	1	0	0	1	0	1	1	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

? 1 0 1 1 1

Here Four 1's (Even), so P1 is 0

✓

0	1	1	1	0	0	1	0	1	1	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

Computer Networks

P2: check 2 bit, skip 2 bit, check 2 bit, skip 2 bit, etc. (2,3,6,7,10, 11, …)

0	1	1	1	0	0	1	0	1	1	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

? 1 0 1 1 1

Here Four 1's (Even), so P2 is 0

✓ ×

0	1	1	1	0	0	1	0	1	1	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

P4: check 4 bit, skip 4 bit, check 4 bit, skip 4 bit, etc. (4,5,6,7,12,13,14,15, …)

0	1	1	1	0	0	1	0	1	1	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

? 0 0 1 0

Here one 1's (Odd), so P4 is 1

✓ × ✓

0	1	1	1	0	0	1	0	1	1	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

Computer Networks

P8: check 8 bit, skip 8 bit, check 8 bit, skip 8 bit, etc. (8−15,24−31,40−47, …)

0	1	1	1	0	0	1	0	1	1	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

? 1 1 1 0

Here Three 1's (Odd), so P8 is 1

✓ × ✓ ×

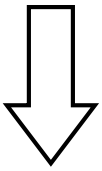
0	1	1	1	0	0	1	0	1	1	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

Here P2 and P8 are incorrect, so 2+8=10th bit is bad bit

✓ × ✓ × Error

0	1	1	1	0	0	1	0	1	1	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

The error will then be corrected



No Error

0	1	1	1	0	0	1	0	1	0	1	0
p1	p2	d3	p4	d5	d6	d7	p8	d9	d10	d11	d12

Computer Networks

DATA LINK CONTROL (DLC) SUBLAYER

DLC services

The data link control (DLC) deals with procedures for communication between two adjacent nodes (node-to-node communication) no matter whether the link is dedicated or broadcast. Data link control functions include framing and flow and error control.

1- Framing

Data transmission in the physical layer means moving bits in the form of a signal from the source to the destination. The physical layer provides bit synchronization to ensure that the sender and receiver use the same bit durations and timing.

The data-link layer, on the other hand, needs to pack bits into frames, so that each frame is distinguishable from another. Our postal system practices a type of framing. The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter. In addition, each envelope defines the sender and receiver addresses, which is necessary since the postal system is a many-to-many carrier facility.

Framing in the data-link layer separates a message from one source to a destination by adding a sender address and a destination address. The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt.

Although the whole message could be packed in one frame, that is not normally done. One reason is that a frame can be very large, making flow and error control very inefficient. When a message is carried in one very large frame, even a single-bit error would require the retransmission of the whole frame. When a message is divided into smaller frames, a single-bit error affects only that small frame.

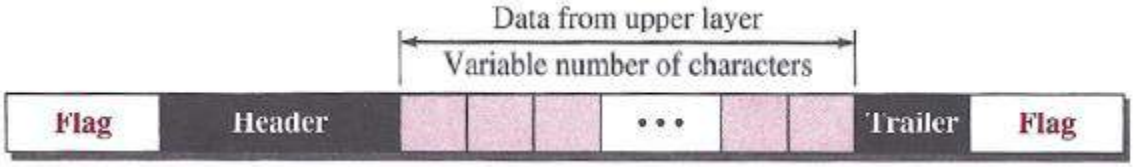
Frame Size

Frames can be of fixed or variable size. In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter. An example of this type of framing is the ATM WAN, which uses frames of fixed size called cells.

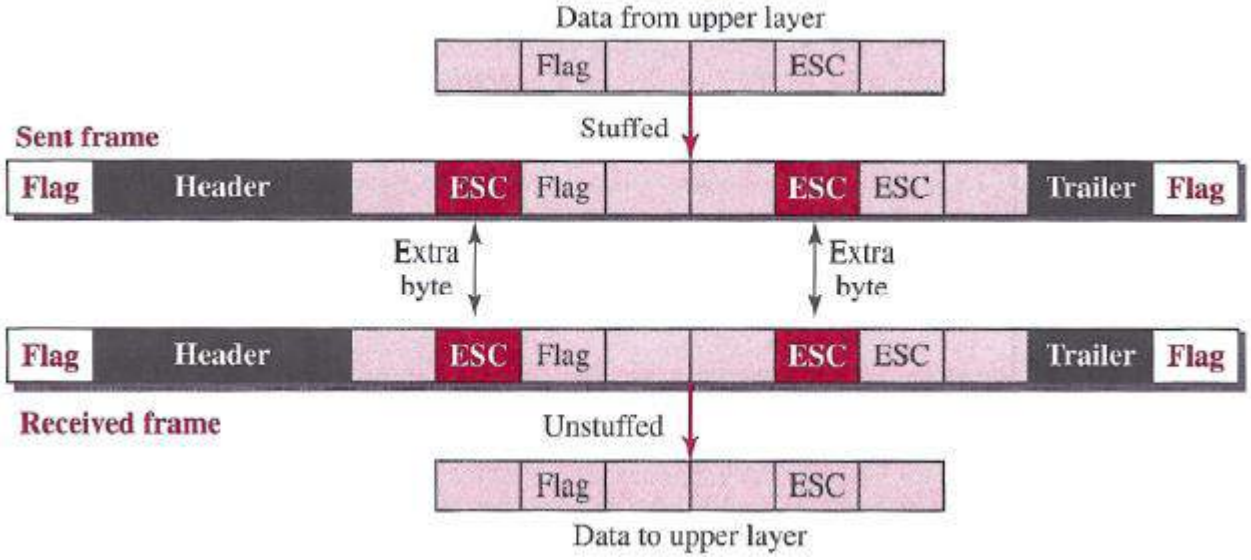
The variable-size framing prevalent in local-area networks. In variable-size framing, we need a way to define the end of one frame and the beginning of the next. Historically, two approaches were used for this purpose: a character-oriented approach and a bit-oriented approach.

Character-Oriented Framing

In character-oriented (or byte-oriented) framing, data to be carried are 8-bit characters from a coding system such as ASCII. The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection redundant bits, are also multiples of 8 bits. To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the end of a frame. The flag, composed of protocol-dependent special characters, signals the start or end of a frame. The figure below shows the format of a frame in a character-oriented protocol.



Character-oriented framing was popular when only text was exchanged by the data-link layers. The flag could be selected to be any character not used for text communication. Now, however, we send other types of information such as graphs, audio, and video; any character used for the flag could also be part of the information. If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame. To fix this problem, a byte-stuffing strategy was added to character-oriented framing. In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC) and has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not as a delimiting flag. The figure below shows the situation.



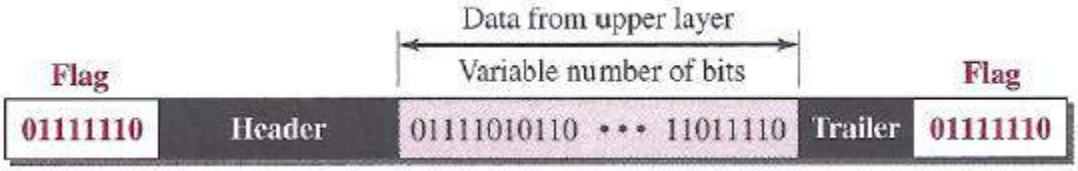
Byte stuffing by the escape character allows the presence of the flag in the data section of the frame, but it creates another problem. What happens if the text contains one or more escape characters followed by a byte with the same pattern as the flag?

The receiver removes the escape character, but keeps the next byte, which is incorrectly interpreted as the end of the frame. To solve this problem, the escape characters that are part of the text must also be marked by another escape character. In other words, if the escape character is part of the text, an extra one is added to show that the second one is part of the text.

Character-oriented protocols present another problem in data communications. The universal coding systems in use today, such as Unicode, have 16-bit and 32-bit characters that conflict with 8-bit characters. We can say that, in general, the tendency is moving toward the bit-oriented protocols that we discuss next.

Bit-Oriented Framing

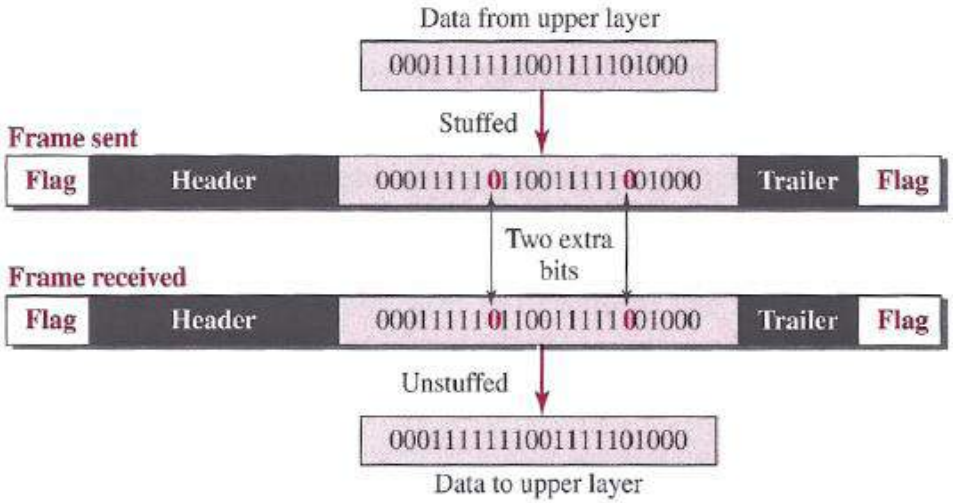
In bit-oriented framing, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on. However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other. Most protocols use a special 8-bit pattern flag, 01111110, as the delimiter to define the beginning and the end of the frame, as shown in the figure below.



This flag can create the same type of problem we saw in the character-oriented protocols. That is, if the flag pattern appears in the data, we need to somehow inform the receiver that this is not the end of the frame. We do this by stuffing 1 single bit (instead of 1 byte) to prevent the pattern from looking like a flag. The strategy is called bit stuffing. In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added. This extra stuffed bit is eventually removed from the data by the receiver. Note that the extra bit is added after one 0 followed by five 1s regardless of the value of the next bit. This guarantees that the flag field sequence does not inadvertently appear in the frame.

The figure below shows bit stuffing at the sender and bit removal at the receiver. Note that even if we have a 0 after five 1s, we still stuff a 0. The 0 will be removed by the receiver. This means that if the flag like pattern 01111110 appears in the data, it will change to 011111010 (stuffed) and is not

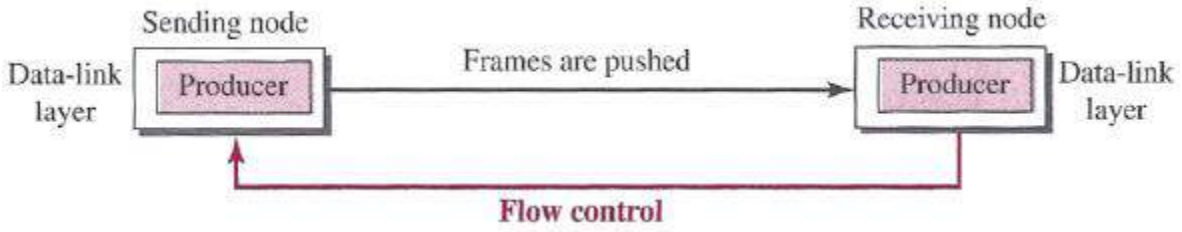
mistaken for a flag by the receiver. The real flag 01111110 is not stuffed by the sender and is recognized by the receiver.



2- Flow Control

Whenever an entity produces items and another entity consumes them, there should be a balance between production and consumption rates. If the items are produced faster than they can be consumed, the consumer can be overwhelmed and may need to discard some items. If the items are produced more slowly than they can be consumed, the consumer must wait, and the system becomes less efficient. Flow control is related to the first issue. We need to prevent losing the data items at the consumer site.

In communication at the data-link layer, we are dealing with four entities: network and data-link layers at the sending node and network and data-link layers at the receiving node. Although we can have a complex relationship with more than one producer and consumer, we ignore the relationships between networks and data-link layers and concentrate on the relationship between two data-link layers, as shown in the figure below.



The figure shows that the data-link layer at the sending node tries to push frames toward the data-link layer at the receiving node. If the receiving node cannot process and deliver the packet to its network at the same rate that the frames arrive, it becomes overwhelmed with frames. Flow control in this case can be feedback from the receiving node to the sending node to stop or slow down pushing frames.

Buffers

Although flow control can be implemented in several ways, one of the solutions is normally to use two buffers; one at the sending data-link layer and the other at the receiving data-link layer. A buffer is a set of memory locations that can hold packets at the sender and receiver. The flow control communication can occur by sending signals from the consumer to the producer. When the buffer of the receiving data-link layer is full, it informs the sending data-link layer to stop pushing frames.

3- Error Control

Since the underlying technology at the physical layer is not fully reliable, we need to implement error control at the data- link layer to prevent the receiving node from delivering corrupted packets to its network layer. Error control at the data-link layer is normally very simple and implemented using one of the following two methods. In both methods, a CRC is added to the frame header by the sender and checked by the receiver.

- In the first method, if the frame is corrupted, it is silently discarded; if it is not corrupted, the packet is delivered to the network layer. This method is used mostly in wired LANs such as Ethernet.
- In the second method, if the frame is corrupted, it is silently discarded; if it is not corrupted, an acknowledgment is sent (for the purpose of both flow and error control) to the sender.

Connectionless and Connection-Oriented

A DLC protocol can be either connectionless or connection-oriented. We will discuss this issue very briefly here, but we return to this topic in the network and transport layer.

1- Connectionless Protocol

In a connectionless protocol, frames are sent from one node to the next without any relationship between the frames; each frame is independent. Note that the term connectionless here does not mean that there is no physical connection (transmission medium) between the nodes; it means that there is no connection between frames. The frames are not numbered and there is no sense of ordering. Most of the data-link protocols for LANs are connectionless protocols.

2- Connection-Oriented Protocol

In a connection-oriented protocol, a logical connection should first be established between the two nodes (setup phase). After all frames that are somehow related to each other are transmitted (transfer phase), the logical connection is terminated (teardown phase). In this type of communication, the frames are numbered and sent in order. If they are not received in order, the receiver needs to wait until all frames belonging to the same set are received and then deliver them in order to the network layer. Connection-oriented protocols are rare in wired LANs, but we can see them in some point-to-point protocols, some wireless LANs, and some WANs.

Computer Networks

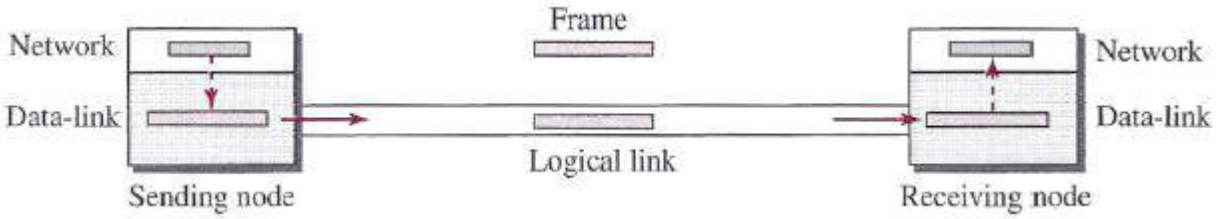
DATA-LINK LAYER PROTOCOLS

Traditionally four protocols have been defined for the data-link layer to deal with flow and error control: Simple, Stop-and-Wait, Go-Back-N, and Selective-Repeat. Although the first two protocols still are used at the data-link layer, the last two have disappeared.

We therefore briefly discuss the first two protocols in this chapter, in which we need to understand some wired and wireless LANs. We postpone the discussion of all four later, where we discuss the transport layer.

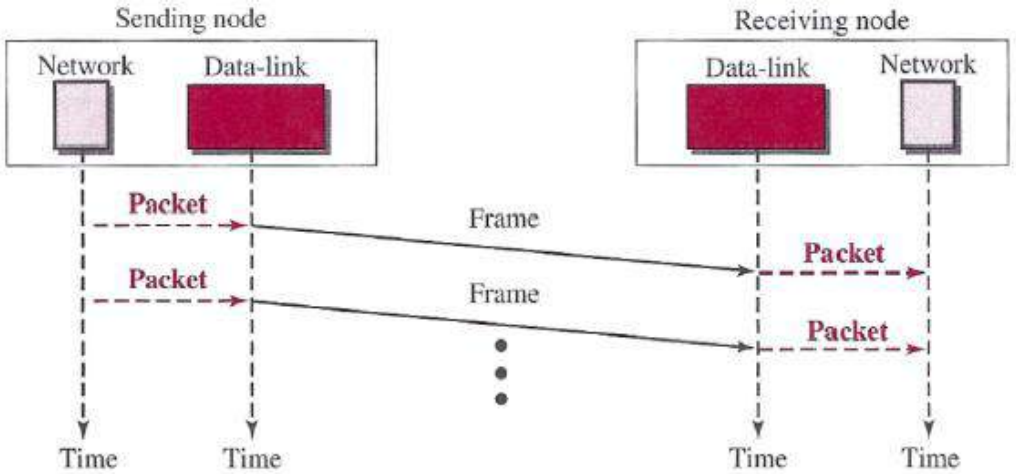
1- Simple Protocol

Our first protocol is a simple protocol with neither flow nor error control. We assume that the receiver can immediately handle any frame it receives. In other words, the receiver can never be overwhelmed with incoming frames. The figure below shows the layout for this protocol.



The data-link layer at the sender gets a packet from its network layer, makes a frame out of it, and sends the frame. The data-link layer at the receiver receives a frame from the link, extracts the packet from the frame, and delivers the packet to its network layer. The data-link layers of the sender and receiver provide transmission services for their network layers.

Example: The figure below shows an example of communication using this protocol. It is very simple. The sender sends frames one after another without even thinking about the receiver.

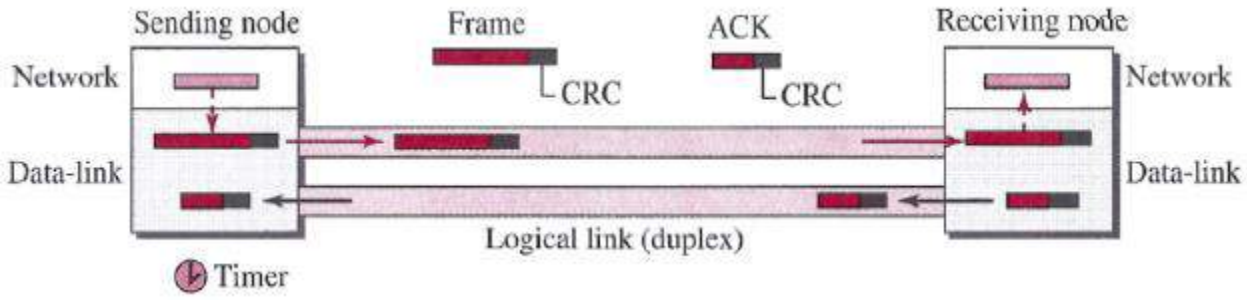


2- Stop-and- Wait Protocol

Our second protocol is called the Stop-and- Wait protocol, which uses both flow and error control. We show a primitive version of this protocol here, but we discuss the more sophisticated version later when we have learned about sliding windows.

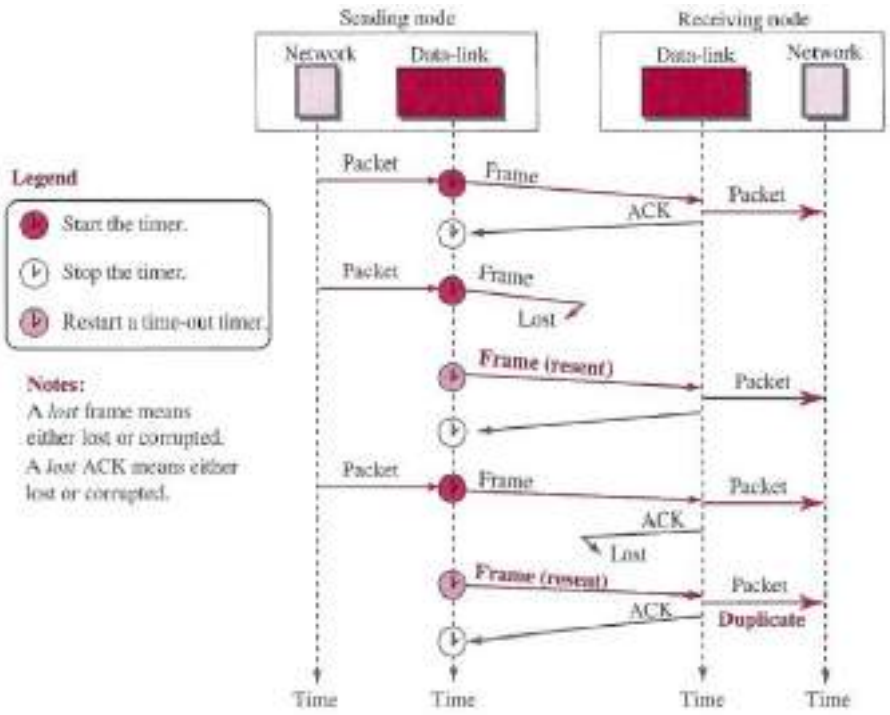
- In this protocol, the sender sends one frame at a time and waits for an acknowledgment before sending the next one.
- To detect corrupted frames, we need to add a CRC to each data frame.
- When a frame arrives at the receiver site, it is checked. If its CRC is incorrect, the frame is corrupted and silently discarded.
- The silence of the receiver is a signal for the sender that a frame was either corrupted or lost.
- Every time the sender sends a frame, it starts a timer. If an acknowledgment arrives before the timer expires, the timer is stopped and the sender sends the next frame (if it has one to send).
- If the timer expires, the sender resends the previous frame, assuming that the frame was either lost or corrupted. This means that the sender needs to keep a copy of the frame until its acknowledgment arrives.
- When the corresponding acknowledgment arrives, the sender discards the copy and sends the next frame if it is ready.

The figure below shows the outline for the Stop-and-Wait protocol. Note that only one frame and one acknowledgment can be in the channels at any time.



Example: The figure below shows an example.

The first frame is sent and acknowledged. The second frame is sent, but lost. After time-out, it is resent. The third frame is sent and acknowledged, but the acknowledgment is lost. The frame is resent. However, there is a problem with this scheme. The network layer at the receiver site receives two copies of the third packet, which is not right.



In the next section, we will see how we can correct this problem using sequence numbers and acknowledgment numbers.

Sequence and Acknowledgment Numbers

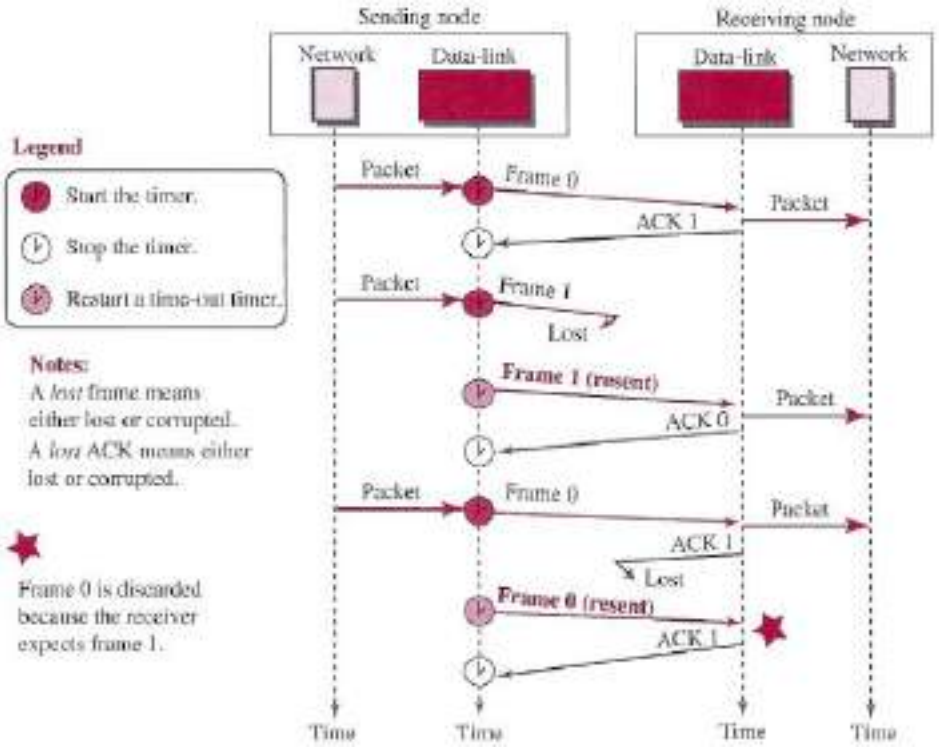
To correct this problem, we need to add sequence numbers to the data frames and acknowledgment numbers to the ACK frames. However, numbering in this case is very simple.

- Sequence numbers are $0, 1, 0, 1, 0, 1, \ldots$
- The acknowledgment numbers can also be $1, 0, 1, 0, 1, 0, \ldots$

In other words, the sequence numbers start with 0, the acknowledgment numbers start with 1. An acknowledgment number always defines the sequence number of the next frame to receive.

Example: The figure below shows how adding sequence numbers and acknowledgment numbers can prevent duplicates. The first frame is sent and acknowledged. The second frame is sent, but lost. After time-out, it is resent. The third frame is sent and acknowledged, but the acknowledgment is lost. The frame is resent.

Computer Networks



Piggybacking

The two protocols we discussed in this section are designed for unidirectional communication, in which data is flowing only in one direction although the acknowledgment may travel in the other direction. Protocols have been designed in the past to allow data to flow in both directions.

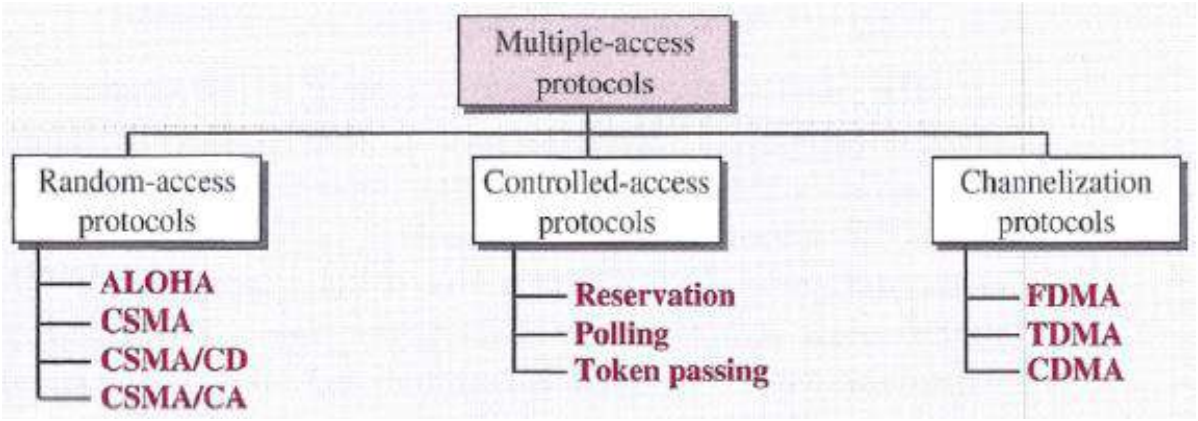
However, to make the communication more efficient, the data in one direction is piggybacked with the acknowledgment in the other direction. In other words, when node A is sending data to node B, Node A also acknowledges the data received from node B.

Because piggybacking makes communication at the datalink layer more complicated, it is not a common practice.

Computer Networks

MEDIA ACCESS CONTROL (MAC) SUBLAYER

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. The problem of controlling the access to the medium is similar to the rules of speaking in an assembly. The procedures guarantee that the right to speak is upheld and ensure that two people do not speak at the same time, do not interrupt each other, do not monopolize the discussion, and so on. Many protocols have been devised to handle access to a shared link. All of these protocols belong to a sublayer in the data-link layer called media access control (MAC). We categorize them into three groups, as shown in figure below.



1- RANDOM ACCESS

In random-access or contention methods, no station is superior to another station and none is assigned control over another. At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send. This decision depends on the state of the medium (idle or busy). In other words, each station can transmit when it desires on the condition that it follows the predefined procedure, including testing the state of the medium. 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.

In a random-access method, each station has the right to the medium without being controlled by any other station. However, if more than one station tries to send, there is an access conflict-collision-and the frames will be either destroyed or modified. To avoid access conflict or to resolve it when it happens, each station follows a procedure that answers the following questions:

- When can the station access the medium?
- What can the station do if the medium is busy?
- How can the station determine the success or failure of the transmission?
- What can the station do if there is an access conflict?

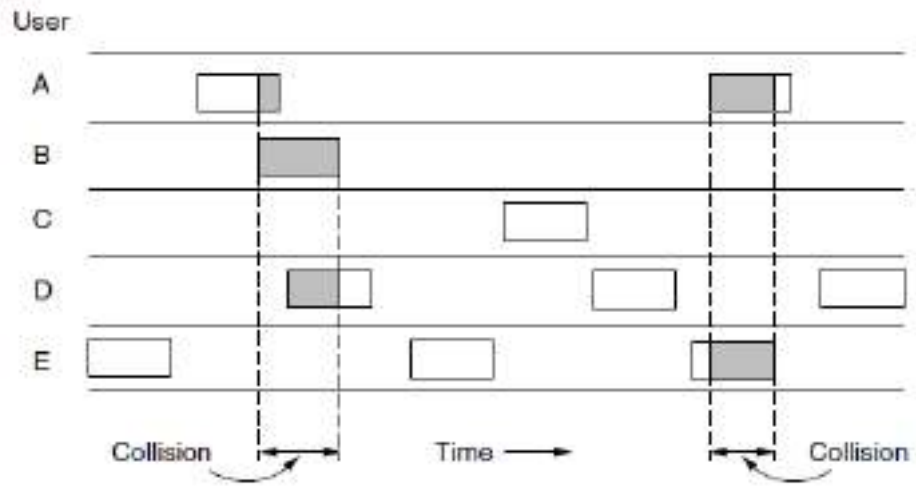
The random-access methods we study in this chapter have evolved from a very interesting protocol known as ALOHA, which used a very simple procedure called multiple access (MA). The method was improved with the addition of a procedure that forces the station to sense the medium before transmitting. This was called carrier sense multiple access (CSMA). This method later evolved into two parallel methods: carrier sense multiple access with collision detection (CSMA/CD), which tells the station what to do when a collision is detected, and carrier sense multiple access with collision avoidance (CSMA/CA), which tries to avoid the collision.

ALOHA Protocols

ALOHA, the earliest random access method, was developed at the University of Hawaii in early 1970. It was designed for a radio (wireless) LAN, but it can be used on any shared medium. It is obvious that there are potential collisions in this arrangement. The medium is shared between the stations. When a station sends data, another station may attempt to do so at the same time. The data from the two stations collide and become garbled.

A- Pure ALOHA

Pure Aloha is an unslotted, fully-decentralized protocol. It is extremely simple and trivial to implement. The ground rule is - "when you want to talk, just talk!". So, a node which wants to transmits, will go ahead and send the packet on its broadcast channel, with no consideration whatsoever as to anybody else is transmitting or not.

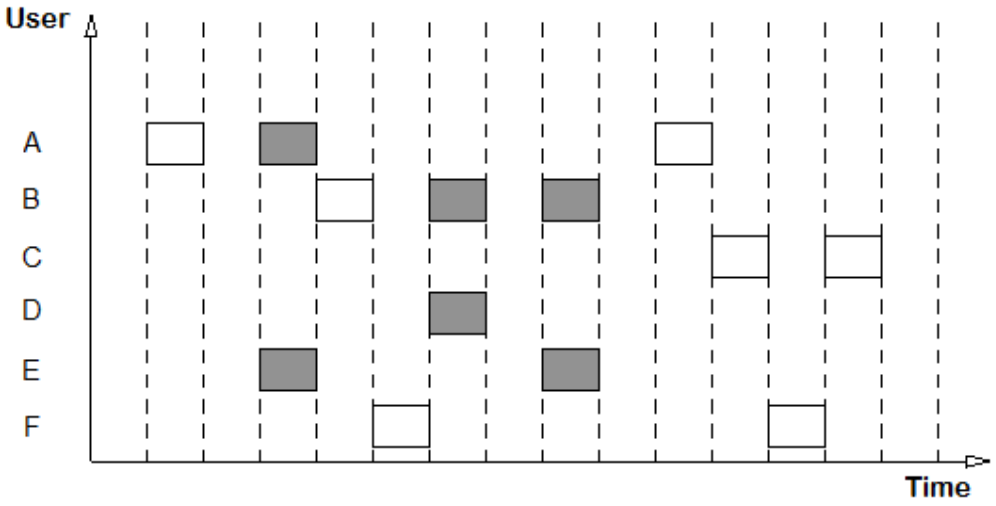


One serious drawback here is that; you don't know whether what you are sending has been received properly or not. To resolve this, in Pure Aloha, when one node finishes speaking, it expects an acknowledgement in a finite amount of time – otherwise it simply retransmits the data. This scheme works well in small networks where the load is not high. But in large, load intensive networks where many nodes may want to transmit at the same time, this scheme fails miserably. This led to the development of Slotted Aloha.

B- Slotted ALOHA

This is quite similar to Pure ALOHA, differing only in the way transmissions take place. Instead of transmitting right at demand time, the sender waits for some time. This delay is specified as follows: The timeline is divided into equal slots and then it is required that transmission should take place only at slot boundaries. To be more precise, the Slotted-Aloha makes the following assumptions:

- All frames consist of exactly L bits.
- Time is divided into slots (a slot equals the time to transmit one frame).
- Nodes start to transmit frames only at the beginnings of slots.
- The nodes are synchronized so that each node knows when the slots begin.
- If two or more frames collide in a slot, then all the nodes detect the collision event before the slot ends.

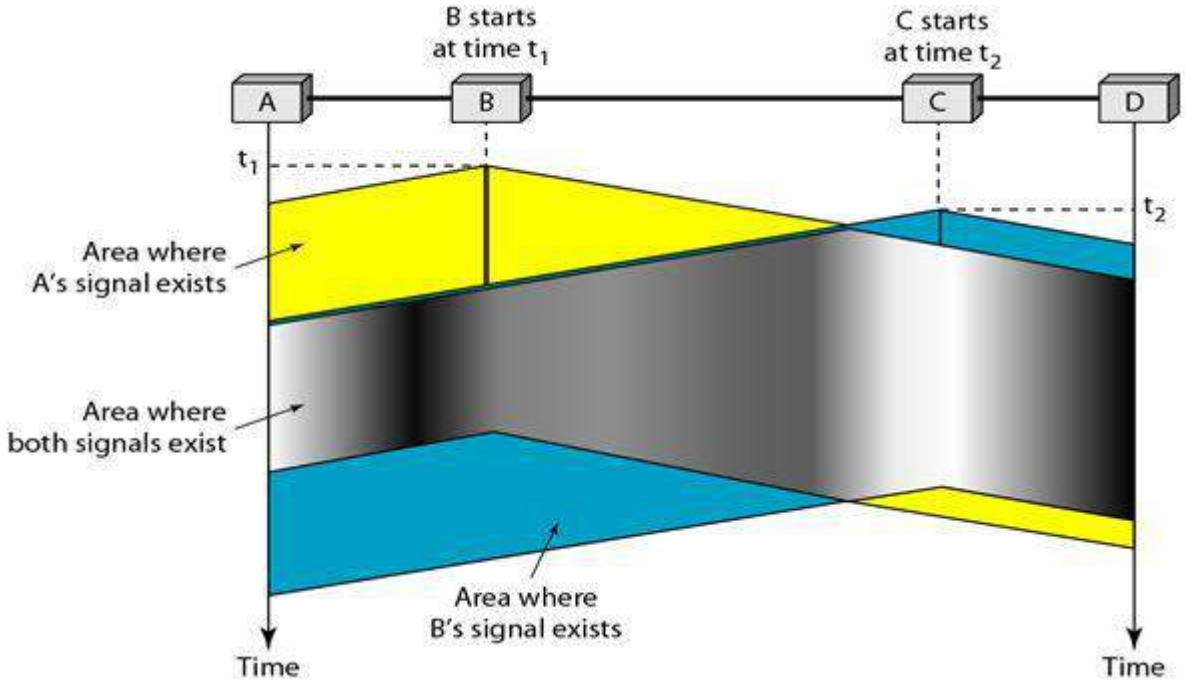


In this way, the number of collisions that can possibly take place is reduced by a huge margin. Hence, the performance become much better compared to Pure ALOHA. Collisions may only take place with nodes that are ready to speak at the same time. However, this is a substantial reduction.

CSMA Protocols

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 the figure below, a space and time model of a CSMA network. Stations are connected to a shared channel (usually a dedicated medium).

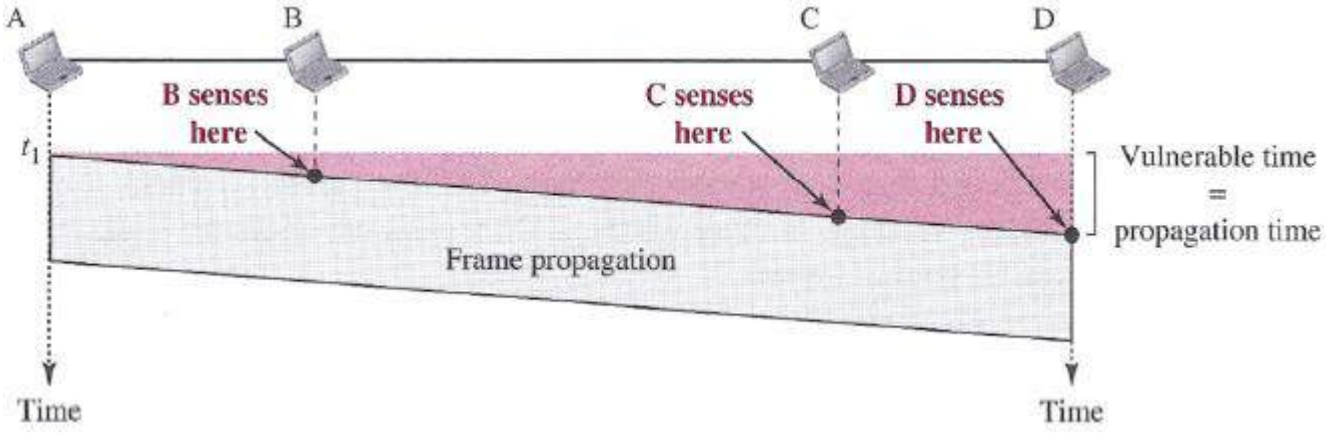


The possibility of collision still exists because of propagation delay; when a station sends a frame, it still takes time (although very short) for the first bit to reach every station and for every station to sense it. In other words, a station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.

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

Vulnerable Time

The vulnerable time is the length of time in which there is a possibility of collision. 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 frame during this time, a collision will result. But if the first bit of the frame reaches the end of the medium, every station will already have heard the bit and will refrain from sending. The figure below shows the worst case. The leftmost station, A, sends a frame at time t_1, which reaches the rightmost station, D, at time t_1 + Tp. The color area shows the vulnerable area in time and space.



Persistence Methods

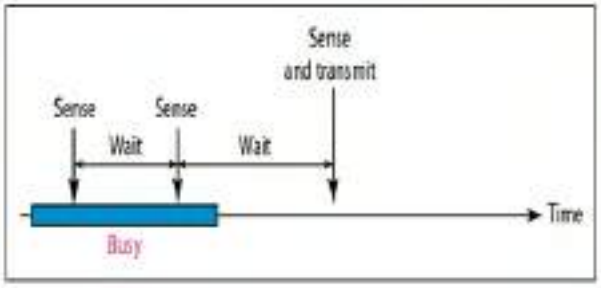
What should a station do if the channel is busy? What should a station do if the channel is idle? Three methods have been devised to answer these questions: The I-persistent method, the nonpersistent method, and the p-persistent method.

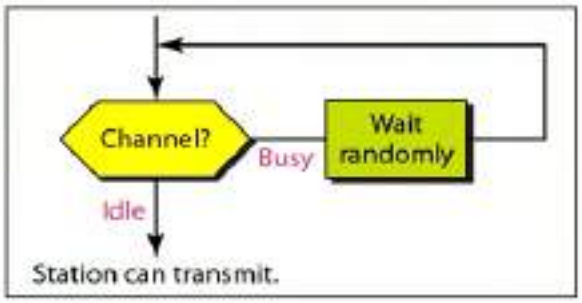
A- 1- persistent CSMA

- When the sender is ready to transmit data, it checks if the medium is busy.
- If busy, it senses the medium continually until it becomes idle.
- If channel is idle, sends the frame immediately (with probability of 1)
- Chance of collision is high.

B- Non-persistent CSMA

- If a station has a frame to send, it senses the channel.
- If channel is idle, the station sends the frame immediately.
- If channel is busy, the station waits a random period of time and then senses the channel again.
- Chance of collision is reduced because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously.
- Reduce the efficiency of the network because the medium remains idle when there may be stations with frames to send.

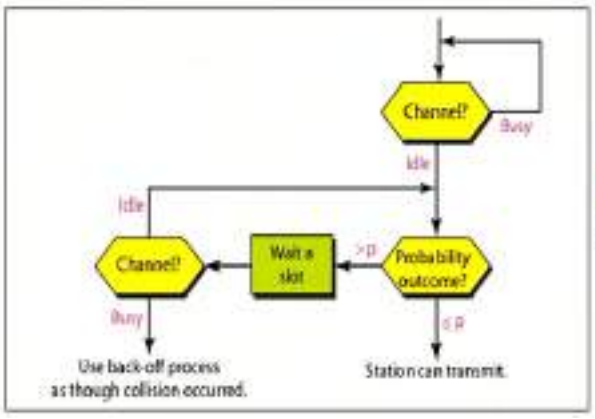


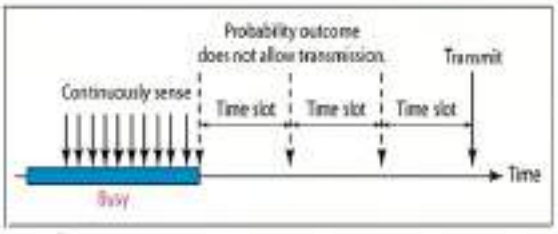


C- p-persistent CSMA

The p-persistent method 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:

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



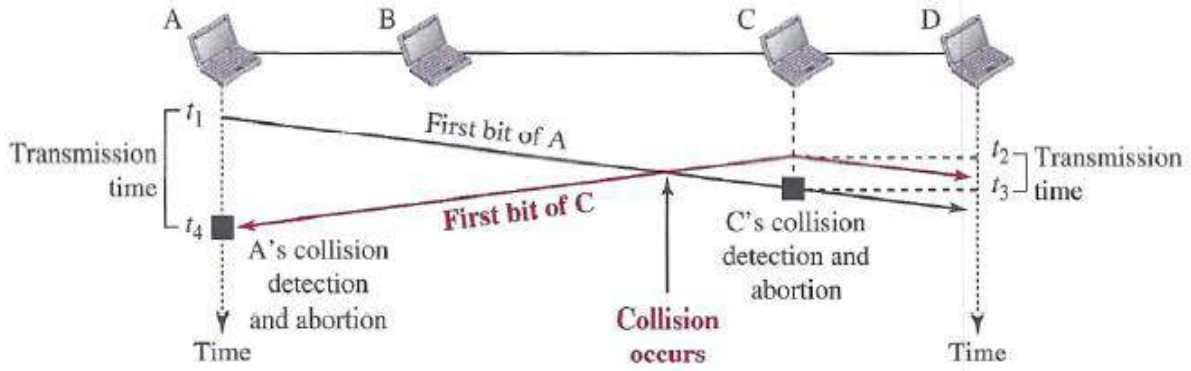


CSMA/CD Protocol

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.

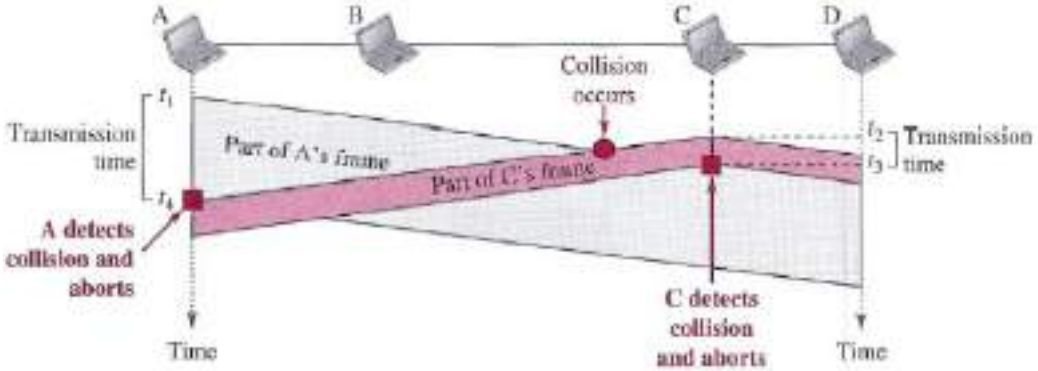
In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If, however, there is a collision, the frame is sent again.

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 the figure below, stations A and C are involved in the collision.



At time t_1, station A has executed its persistence procedure and starts sending the bits of its frame. At time t_2, station C has not yet sensed the first bit sent by A. Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right. The collision occurs sometime after time t_2. Station C detects a collision at time t_3 when it receives the first bit of A's frame. Station C immediately (or after a short time, but we assume immediately) aborts transmission. Station A detects collision at time t_4 when it receives the first bit of C's frame; it also immediately aborts transmission. Looking at the figure, we see that A transmits for the duration $(t_4 - t_1)$; C transmits for the duration $(t_3 - t_2)$.

Now that we know the time durations for the two transmissions, we can show a more complete graph in the figure below.



Minimum Frame Size

For CSMA/CD 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 (Tfr) 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 to reach 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.

Example: A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time is 25.6 μs, what is the minimum size of the frame?

Solution

The minimum frame transmission time is Tfr = 2 x Tp = 51.2 μs.

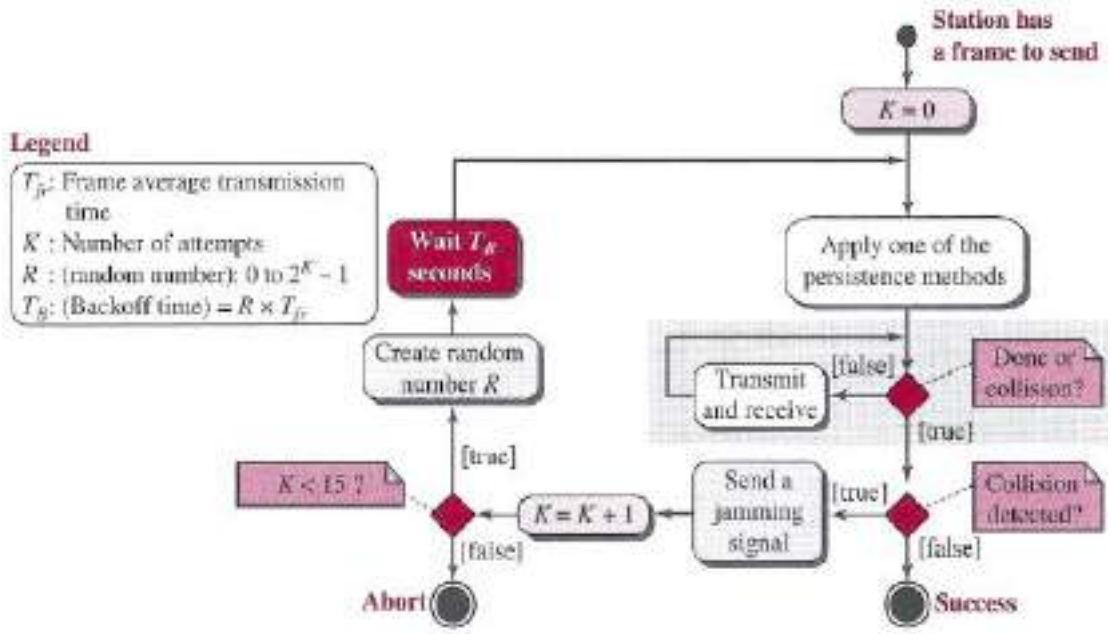
This means, in the worst case, a station needs to transmit for a period of 51.2 μs to detect the collision.

The minimum size of the frame is 10 Mbps x 51.2 μs= 512 bits or 64 bytes.

This is actually the minimum size of the frame for Standard Ethernet, as we will see later.

Procedure

Now let us look at the flow diagram for CSMA/CD in the figure below.

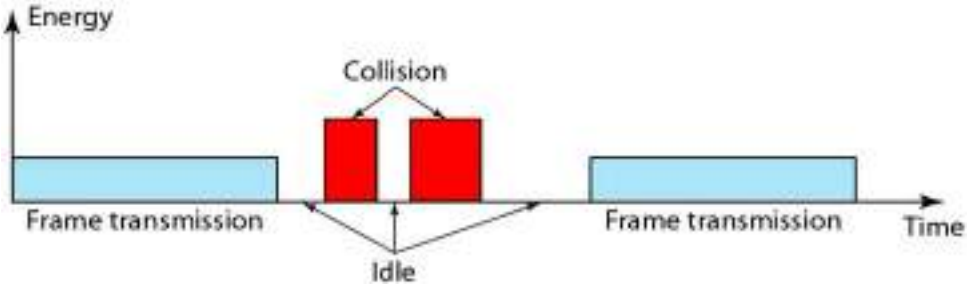


Energy Level

We can say that the level of energy in a channel can have three values: zero, normal, and abnormal.

- At the zero level, the channel is idle.
- At the normal level, a station has successfully captured the channel and is sending its frame.
- At the abnormal level, there is a collision and the level of the energy is twice the normal level.

A station that has a frame to send or is sending a frame needs to monitor the energy level to determine if the channel is idle, busy, or in collision mode. Figure below shows the situation.



2- CONTROLLED ACCESS

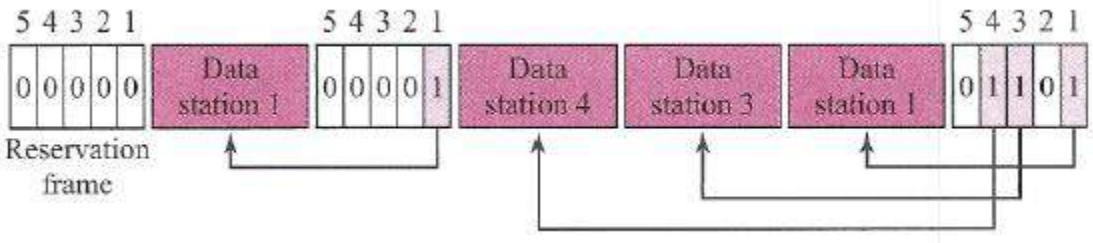
In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations. We discuss three controlled-access methods.

Reservation Protocol

In the reservation method, a station needs to make a reservation before sending data. Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.

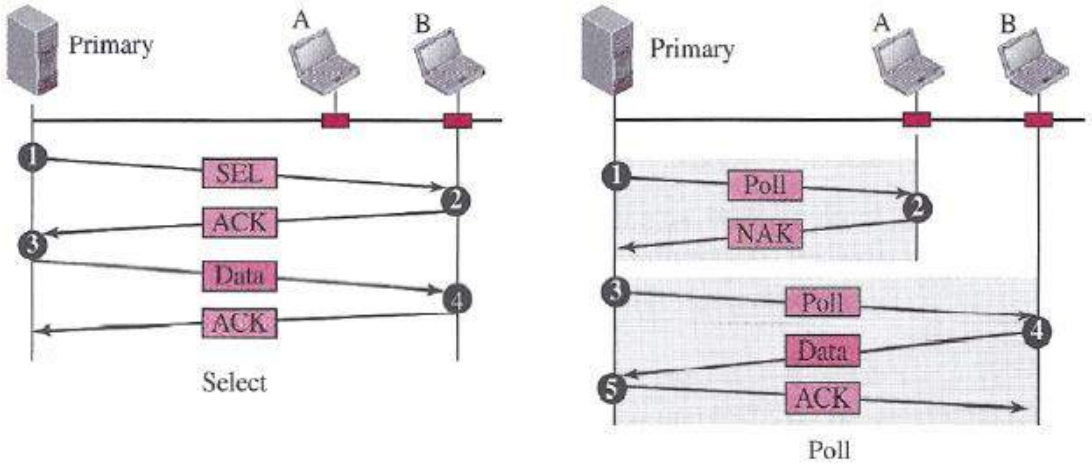
If there are N stations in the system, there are exactly N reservation minislots in the reservation frame. Each minislot belongs to a station. When a station needs to send a data frame, it makes a reservation in its own minislot. The stations that have made reservations can send their data frames after the reservation frame.

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



Polling Protocol

Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations. All data exchanges must be made through the primary device even when the ultimate destination is a secondary device. The primary device controls the link; the secondary devices follow its instructions. It is up to the primary device to determine which device is allowed to use the channel at a given time. The primary device, therefore, is always the initiator of a session (see the figure below). This method uses poll and select functions to prevent collisions. However, the drawback is if the primary station fails, the system goes down.



Select

The select function is used whenever the primary device has something to send. Remember that the primary controls the link. If the primary is neither sending nor receiving data, it knows the link is available. If it has something to send, the primary device sends it. What it does not know, however, is whether the target device is prepared to receive. So the primary must alert the secondary to the upcoming transmission and wait for an acknowledgment of the secondary's ready status. Before sending data, the primary creates and transmits a select (SEL) frame, one field of which includes the address of the intended secondary.

Poll

The poll function is used by the primary device to solicit transmissions from the secondary devices. When the primary is ready to receive data, it must ask (poll) each device in turn if it has anything to send. When the first secondary is approached, it responds either with a NAK frame if it has nothing to send or with data (in the form of a data frame) if it does. If the response is negative (a NAK frame), then the primary polls the next secondary in the same manner until it finds one with data to send. When the response is positive (a data frame), the primary reads the frame and returns an acknowledgment (ACK frame), verifying its receipt.

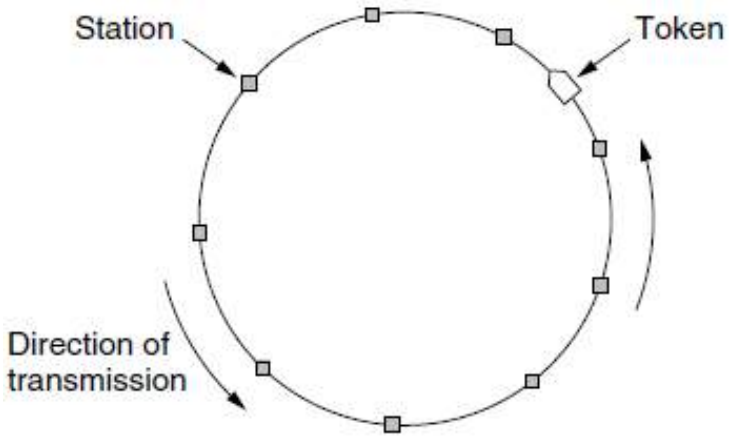
Token Passing Protocol

In the token-passing method, the stations in a network are organized in a logical ring. In other words, for each station, there is a predecessor and a successor. The predecessor is the station which is logically before the station in the ring; the successor is the station which is after the station in the ring. The current station is the one that is accessing the channel now. The right to this access has been passed from the predecessor to the current station. The right will be passed to the successor when the current station has no more data to send.

But how is the right to access the channel passed from one station to another?

In this method, a special packet called a token circulates through the ring. The possession of the token gives the station the right to access the channel and send its data. When a station has some data to send, it waits until it receives the token from its predecessor. It then holds the token and sends its data. When the station has no more data to send, it releases the token, passing it to the next logical station in the ring. The station cannot send data until it receives the token again in the next round. In this process, when a station receives the token and has no data to send, it just passes the data to the next station.

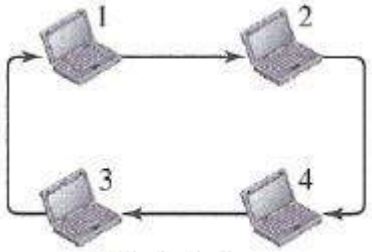
Token management is needed for this access method. Stations must be limited in the time they can have possession of the token. The token must be monitored to ensure it has not been lost or destroyed. For example, if a station that is holding the token fails, the token will disappear from the network. Another function of token management is to assign priorities to the stations and to the types of data being transmitted. Finally, token management is needed to make low-priority stations release the token to high-priority stations.



Logical Ring

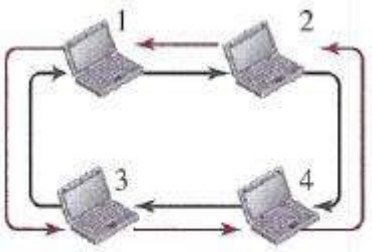
In a token-passing network, stations do not have to be physically connected in a ring; the ring can be a logical one. The figures below show four different physical topologies that can create a logical ring.

- In the physical ring topology, when a station sends the token to its successor, the token cannot be seen by other stations; the successor is the next one in line. This means that the token does not have to have the address of the next successor. The problem with this topology is that if one of the links-the medium between two adjacent stations-fails, the whole system fails.



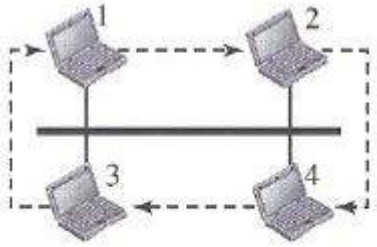
Physical ring

- The dual ring topology uses a second (auxiliary) ring which operates in the reverse direction compared with the main ring. The second ring is for emergencies only (such as a spare tire for a car). If one of the links in the main ring fails, the system automatically combines the two rings to form a temporary ring. After the failed link is restored, the auxiliary ring becomes idle again. Note that for this topology to work, each station needs to have two transmitter ports and two receiver ports. The high-speed Token Ring networks called FDDI (Fiber Distributed Data Interface] and CDDI (Copper Distributed Data Interface] use this topology.



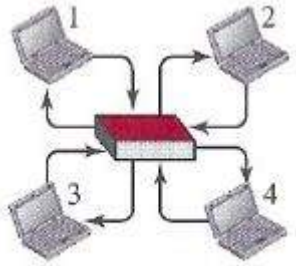
Dual ring

- In the bus ring topology, also called a token bus, the stations are connected to a single cable called a bus. They, however, make a logical ring, because each station knows the address of its successor (and also predecessor for token management purposes). When a station has finished sending its data, it releases the token and inserts the address of its successor in the token. Only the station with the address matching the destination address of the token gets the token to access the shared media. The Token Bus LAN, standardized by IEEE, uses this topology.



Bus ring

– In a star ring topology, the physical topology is a star. There is a hub, however, that acts as the connector. The wiring inside the hub makes the ring; the stations are connected to this ring through the two wire connections. This topology makes the network less prone to failure because if a link goes down, it will be bypassed by the hub and the rest of the stations can operate. Also adding and removing stations from the ring is easier. This topology is still used in the Token Ring LAN designed by IBM.



Star ring

Wired LANs: Ethernet

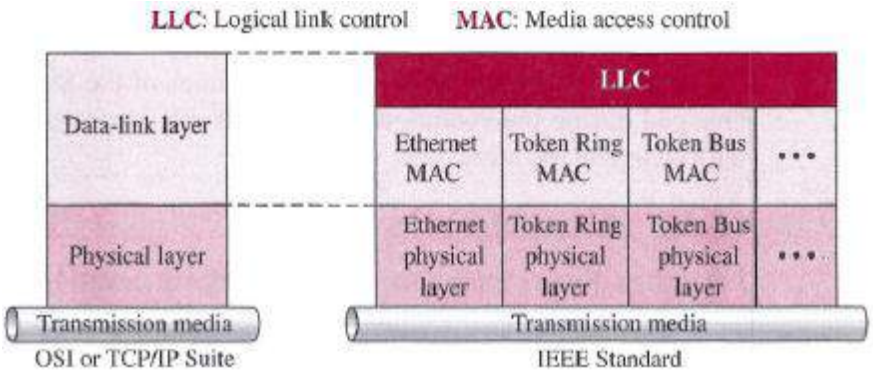
In Chapter 1, we mentioned that the TCP/IP protocol suite does not define any protocol for the data-link or the physical layer. In other words, TCP/IP accepts any protocol at these two layers that can provide services to the network layer. The data-link layer and the physical layer are actually the territory of the local and wide area networks. This means that when we discuss these two layers, we are talking about networks that are using them.

In the 1980s and 1990s several different types of LANs were used. All of these LANs used a media-access method to solve the problem of sharing the media. The Ethernet used the CSMA/CD approach. The Token Ring, Token Bus, and FDDI (Fiber Distribution Data Interface) used the token-passing approach.

Almost every LAN except Ethernet has disappeared from the marketplace because Ethernet was able to update itself to meet the needs of the time. Several reasons for this success have been mentioned in the literature, but we believe that the Ethernet protocol was designed so that it could evolve with the demand for higher transmission rates. It is natural that an organization that has used an Ethernet LAN in the past and now needs a higher data rate would update to the new generation instead of switching to another technology, which might cost more. This means that we confine our discussion of wired LANs to the discussion of Ethernet.

IEEE PROJECT 802

In February 1980, 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 does not seek to replace any part of the OSI model or TCP/IP protocol suite. Instead, it 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 TCP/IP protocol suite is shown in the figure below.



The IEEE has subdivided the data-link layer into two sublayers:

- Logical link control (LLC)
- Media access control (MAC).

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

Logical Link Control (LLC)

Earlier we discussed data link control. We said that data link control handles framing, flow control, and error control. In IEEE Project 802, flow control, error control, and part of the framing duties are collected into one sublayer called the logical link control (LLC). Framing is handled in both the LLC sublayer and the MAC sublayer. The LLC provides a single link-layer control protocol for all IEEE LANs. This means LLC protocol can provide interconnectivity between different LANs because it makes the MAC sublayer transparent.

Media Access Control (MAC)

Earlier we discussed multiple access methods including random access, controlled access, and channelization. IEEE Project 802 has created a sublayer called media access control that defines the specific access method for each LAN. For example, it defines CSMA/CD as the media access method for Ethernet LANs and defines the token-passing method for Token Ring and Token Bus LANs. As we mentioned in the previous section, part of the framing function is also handled by the MAC layer.

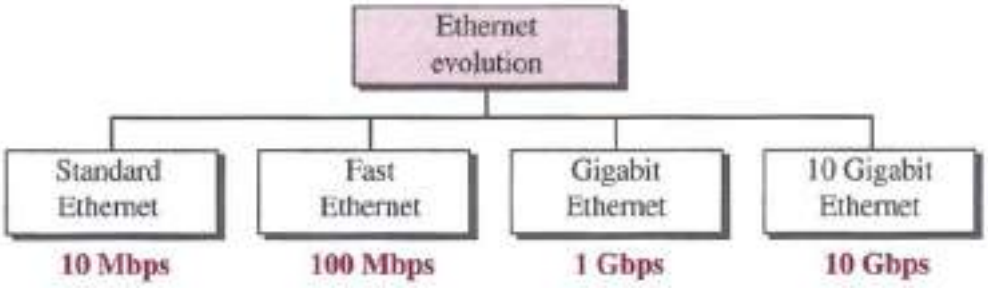
For information only, we list you below some of the IEEE 802 Networking Standards.

Standard	Topic
802.1	LAN/MAN Management (and Media Access Control Bridges)
802.2	Logical Link Control (LLC)
802.3	Ethernet (CSMA/CD)
802.4	Token Passing Bus
802.5	Token Passing Ring
802.6	Distributed Queue Dual Bus (DQDB) Metropolitan Area Network (MAN)
802.7	Broadband Local Area Networks
802.8	Fiber-Optic LANs and MANs
802.9	ISLAN or isoEthernet
802.10	LAN/MAN Security
802.11	Wireless LAN
802.12	Demand Priority Access Method
802.15	Wireless Personal Area Network
802.16	Wireless Metropolitan Area Network (also called WiMAX)
802.17	Resilient Packet Ring
802.18	LAN/MAN Standards Committee

Ethernet Evolution

The Ethernet LAN was developed in the 1970s by Robert Metcalfe and David Boggs. Since then, it has gone through four generations:

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



STANDARD ETHERNET

We refer to the original Ethernet technology with the data rate of 10 Mbps as the Standard Ethernet. Although most implementations have moved to other technologies in the Ethernet evolution, there are some features of the Standard Ethernet that have not changed during the evolution. We discuss this standard version to pave the way for understanding the other three technologies.

Characteristics

Let us first discuss some characteristics of the Standard Ethernet.

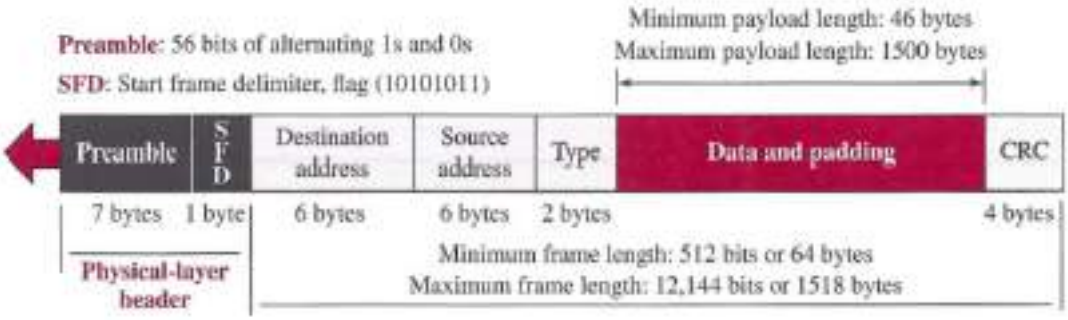
1- Connectionless and Unreliable Service

Ethernet provides a connectionless service, which means each frame sent is independent of the previous or next frame. Ethernet has no connection establishment or connection termination phases. The sender sends a frame whenever it has it; the receiver may or may not be ready for it. The sender may overwhelm the receiver with frames, which may result in dropping frames. If a frame drops, the sender will not know about it. Since IP, which is using the service of Ethernet, is also connectionless, it will not know about it either. If the transport layer is also a connectionless protocol, such as UDP, the frame is lost and salvation may only come from the application layer. However, if the transport layer is TCP, the sender TCP does not receive acknowledgment for its segment and sends it again.

Ethernet is also unreliable like IP and UDP. If a frame is corrupted during transmission and the receiver finds out about the corruption, which has a high level of probability of happening because of the CRC-32, the receiver drops the frame silently. It is the duty of high-level protocols to find out about it.

2- Frame Format

The Ethernet frame contains seven fields, as shown in the figure below.



- **Preamble:** This field contains 7 bytes (56 bits) of alternating 0s and 1s that alert the receiving system to the coming frame and enable it to synchronize its clock if it's out of synchronization. The pattern provides only an alert and a timing pulse. The 56-bit pattern allows the stations to miss some bits at the beginning of the frame. The preamble is actually added at the physical layer and is not (formally) part of the frame.

- **Start frame delimiter (SFD):** This field (1 byte: 10101011) signals the beginning of the frame. The SFD warns the station or stations that this is the last chance for synchronization. The last 2 bits are $(11)_2$ and alert the receiver that the next field is the destination address. This field is actually a flag that defines the beginning of the frame. We need to remember that an Ethernet frame is a variable-length frame. It needs a flag to define the beginning of the frame. The SFD field is also added at the physical layer.

- **Destination address (DA):** This field is six bytes (48 bits) and contains the link-layer address of the destination station or stations to receive the packet. When the receiver sees its own link-layer address, or a multicast address for a group that the receiver is a member of, or a broadcast address, it decapsulates the data from the frame and passes the data to the upper layer protocol defined by the value of the type field.

- **Source address (SA):** This field is also six bytes and contains the link-layer address of the sender of the packet.

- **Type:** This field defines the upper-layer protocol whose packet is encapsulated in the frame. This protocol can be IP, ARP, OSPF, and so on. In other words, it serves the same purpose as the protocol field in a datagram and the port number in a segment or user datagram. It is used for multiplexing and demultiplexing.

- **Data**: This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes.
  - If the data coming from the upper layer is more than 1500 bytes, it should be fragmented and encapsulated in more than one frame.
  - If it is less than 46 bytes, it needs to be padded with extra 0s. A padded data frame is delivered to the upper-layer protocol as it is (without removing the padding), which means that it is the responsibility of the upper layer to remove or, in the case of the sender, to add the padding. The upper-layer protocol needs to know the length of its data. For example, a datagram has a field that defines the length of the data.

- **CRC**: The last field contains error detection information, in this case a CRC-32. The CRC is calculated over the addresses, types, and data field. If the receiver calculates the CRC and finds that it is not zero (corruption in transmission), it discards the frame.

3- Frame Length

Ethernet has imposed restrictions on both the minimum and maximum lengths of a frame.

The minimum length restriction is required for the correct operation of CSMA/CD. 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 type, and 4 bytes of CRC), then the minimum length of data from the upper layer is 64 - 18 = 46 bytes.

If the upper-layer packet is less than 46 bytes, padding is added to make up the difference.

The standard defines the maximum length of a frame (without preamble and SFD field) as 1518 bytes. If we subtract the 18 bytes of header and trailer, the maximum length of the payload is 1500 bytes.

The maximum length restriction has two historical reasons.

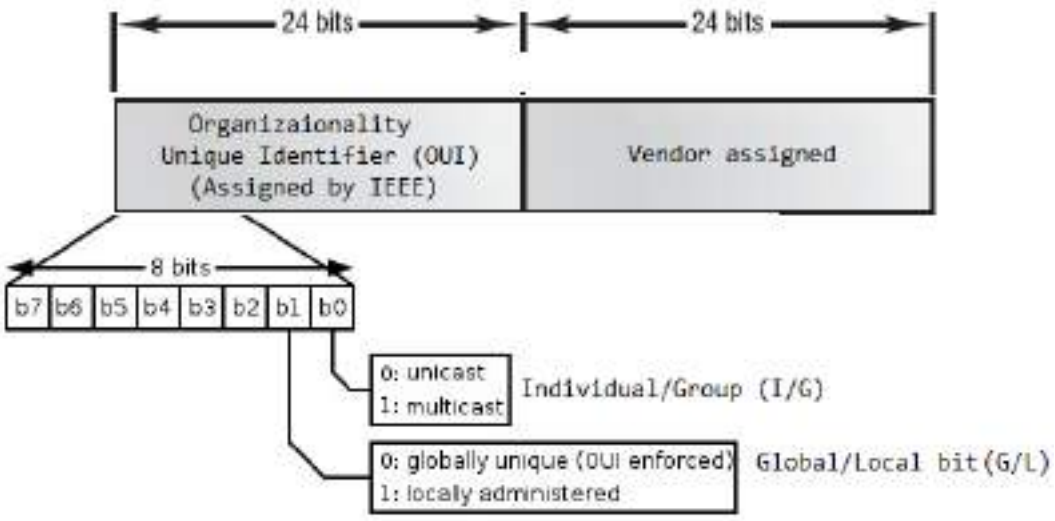
- **First**: Memory was very expensive when Ethernet was designed; a maximum length restriction helped to reduce the size of the buffer.
- **Second**: The maximum length restriction prevents one station from monopolizing the shared medium, blocking other stations that have data to send.

Minimum frame length: 64 bytes	**Maximum frame length: 1518 bytes**
Minimum data length: 46 bytes	**Maximum data length: 1500 bytes**

Computer Networks

Ethernet Addressing

Each station on an Ethernet network (such as a PC, workstation, or printer) has its own network interface card (NIC). The NIC fits inside the station and provides the station with a link-layer address. The Ethernet address is 6 bytes (48 bits), normally written in **hexadecimal notation**, with a colon between the bytes. The figure below shows the 48-bit MAC addresses and how the bits are divided.



- The organizationally unique identifier (OUI) is assigned by the Institute of Electrical and Electronics Engineers (IEEE) to an organization. It's composed of 24 bits, or 3 bytes. The organization, in turn, assigns a globally administered address (24 bits, or 3 bytes) that is unique (supposedly—no guarantees) to each and every adapter it manufactures. Look closely at the figure.
  - The first bit of last byte is the Individual/Group (I/G) bit. When it has a value of 0, we can assume that the address is the MAC address of a device and may well appear in the source portion of the MAC header. When it is a 1, we can assume that the address represents either a broadcast or multicast address in Ethernet or a broadcast.
  - The next bit is the Global/Local bit (G/L, also known as U/L, where U means universal). When set to 0, this bit represents a globally administered address (as standardized by the IEEE). When the bit is a 1, it represents a locally governed and administered address.
- The low-order 24 bits of an Ethernet address represent a locally administered or manufacturer assigned code. This portion commonly starts with 24 0s for the first card made and continues in order until there are 24 1s for the last (16,777,216th) card made. You'll find that many manufacturers use these same six hex digits as the last six characters of their serial number on the same card.

Transmission of Address Bits

The way the addresses are sent out online is different from the way they are written in hexadecimal notation. The transmission is left to right, byte by byte; however, for each byte, the least significant bit is sent first and the most significant bit is sent last. This means that the bit that defines an address as unicast or multicast arrives first at the receiver. This helps the receiver to immediately know if the packet is unicast or multicast.

Example: Show how the address 47:20:IB:2E:08:EE is sent out online.

Solution: The address is sent left to right, byte by byte; for each byte, it is sent right to left, bit by bit, as shown below:

Hexadecimal:	47	20	1B	2E	08	EE
Binary:	01000111	00100000	00011011	00101110	00000100	11101110
Transmitted: ←	11100010	00000100	11011000	01110100	00010000	01110111

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.

If the least significant bit of the first byte in a destination address is 0, the address is unicast; otherwise, it is multicast. Note that with the way the bits are transmitted, the unicast/multicast bit is the first bit which is transmitted or received. The broadcast address is a special case of the multicast address: the recipients are all the stations on the LAN. A broadcast destination address is forty-eight Is.

Example: Define the type of the following destination addresses:

4A:30:10:21:10:1A 47:20:IB:2E:08:EE FF:FF:FF:FF:FF:FF

Solution: To find the type of the address, we need to look at the second hexadecimal digit from the left. If it is even, the address is unicast. If it is odd, the address is multicast. If all digits are Fs, the address is broadcast. Therefore, we have the following:

4A:30:10:21:10:1A is a **unicast** address because A in binary is 1010 (even).

47:20:IB:2E:08:EE is a **multicast** address because 7 in binary is 0111 (odd).

FF:FF:FF:FF:FF:FF is a **broadcast** address because all digits are Fs in hexadecimal.

Efficiency of Standard Ethernet

The efficiency of the Ethernet is defined as the ratio of the time used by a station to send data to the time the medium is occupied by this station. The practical efficiency of standard Ethernet has been measured to be:

$$\text{Efficiency} = \frac{1}{(1 + 6.4 \text{ x a})}$$

- The parameter "a" is the number of frames that can fit on the medium.
- It can be calculated as a = $\frac{\text{Propagation delay}}{\text{Transmission delay}}$ because the transmission delay is the time it takes a frame of average size to be sent out and the propagation delay is the time it takes to reach the end of the medium.

Note that as the value of parameter a decreases, the efficiency increases. This means that if the length of the media is shorter or the frame size longer, the efficiency increases. In the ideal case, a = 0 and the efficiency is 1.

Example: In the Standard Ethernet with the transmission rate of 10 Mbps, we assume that the length of the medium is 2500 m and the size of the frame is 512 bits. The propagation speed of a signal in a cable is normally 2 x 10^8 m/s.

Propagation delay = $2500/(2 \text{ x } 10^8) = 12.5$ μs
Transmission delay = $512/(10^7) = 51.2$ μs
a = $12.5/51.2 = 0.24$
Efficiency = 39%

The example shows that a = 0.24, which means only 0.24 of a frame occupies the whole medium in this case. The efficiency is 39 percent, which is considered moderate; it means that only 61 percent of the time the medium is occupied but not used by a station.

Implementation

The Standard Ethernet defined several implementations, but only four of them became popular during the 1980s. The table below shows a summary of Standard Ethernet

Implementation	Medium	Medium Length
10Base5	Thick coax	500m
10Base2	Thin coax	185m
10Base-T	2 UTP	100m
10Base-F	2 Fiber	2000m

In the nomenclature 10BaseX, the number defines the data rate (10 Mbps), the term Base means baseband (digital) signal, and X approximately defines either the maximum size of the cable in 100 meters (for example 5 for 500 or 2 for 185 meters) or the type of cable, T for unshielded twisted pair cable (UTP) and F for fiber-optic. The standard Ethernet uses a baseband signal, which means that the bits are changed to a digital signal and directly sent on the line.

FAST ETHERNET (100 Mbps)

In the 1990s, some LAN technologies with transmission rates higher than 10 Mbps, such as FDDI and Fiber Channel, appeared on the market. If the Standard Ethernet wanted to survive, it had to compete with these technologies. Ethernet made a big jump by increasing the transmission rate to 100 Mbps, and the new generation was called the Fast Ethernet. The designers of the Fast Ethernet needed to make it compatible with the Standard Ethernet. The MAC sublayer was left unchanged, which meant the frame format and the maximum and minimum size could also remain unchanged. By increasing the transmission rate, features of the Standard Ethernet that depend on the transmission rate, access method, and implementation had to be reconsidered. The goals of Fast Ethernet can be summarized as follows:

1- Upgrade the data rate to 100 Mbps.

2- Make it compatible with Standard Ethernet.

3- Keep the same 48-bit address.

4- Keep the same frame format.

Implementation

Fast Ethernet implementation at the physical layer can be categorized as either two-wire or four-wire. The two-wire implementation can be either twisted pair (STP, UTP), which is called 100Base-TX, or fiber-optic cable, which is called 100Base-FX. The four-wire implementation is designed only for unshielded twisted pair (UTP), which is called 100Base-T4. The table below is a summary of the Fast Ethernet implementations.

Implementation	Medium	Medium Length	Wires
100Base-TX	UTP or STP	100m	2
100Base-FX	Fiber	2000m	2
10Base-T4	UTP	100m	4

GIGABIT ETHERNET

The need for an even higher data rate resulted in the design of the Gigabit Ethernet Protocol (1000 Mbps). The IEEE committee calls it the Standard 802.3z. The goals of the Gigabit Ethernet were to upgrade the data rate to 1 Gbps, but keep the address length, the frame format, and the maximum and minimum frame length the same. The goals of the Gigabit Ethernet design can be summarized as follows:

1- Upgrade the data rate to 1 Gbps.

2- Make it compatible with Standard or Fast Ethernet.

3- Use the same 48-bit address.

4- Use the same frame format.

5- Keep the same minimum and maximum frame lengths.

6- Support auto-negotiation as defined in Fast Ethernet.

Implementation

The table below is a summary of the Gigabit Ethernet implementations. S-Wand L-W mean short-wave and long-wave respectively.

Implementation	Medium	Medium Length	Wires
1000Base-SX	Fiber S-W	550m	2
1000Base-LX	Fiber L-W	2000m	2
1000Base-CX	STP	25m	2
1000Base-T4	UTP	100m	4

10 GIGABIT ETHERNET

In recent years, there has been another look into the Ethernet for use in metropolitan areas. The idea is to extend the technology, the data rate, and the coverage distance so that the Ethernet can be used as LAN and MAN (metropolitan area network). The IEEE committee created 10 Gigabit Ethernet and called it Standard 802.3ae. The goals of the 10 Gigabit Ethernet design can be summarized as follows:

1- Upgrading the data rate to 10 Gbps,

2- Keeping the same frame size and format

3-Allowing the interconnection of LAN s, MANs, and WAN possible.

This data rate is possible only with fiber-optic technology at this time.

Implementation

10 Gigabit Ethernet operates only in full-duplex mode, which means there is no need for contention; CSMA/CD is not used in 10 Gigabit Ethernet. Four implementations are the most common: 10GBase-SR, 10GBase-LR, 10GBase-EW, and 10GBase-X4. The table below shows a summary of the 10 Gigabit Ethernet implementations:

Implementation	Medium	Medium Length	Wires
10GBase-SR	Fiber 850nm laser	300m	2
10GBase-LR	Fiber 1310nm laser	10km	2
10GBase-EW	Fiber 1550nm laser	40km	2
10GBase-X4	Fiber 1310nm laser	300m to 10Km	2

Computer Networks

WIRELESS LANS: Wi-Fi

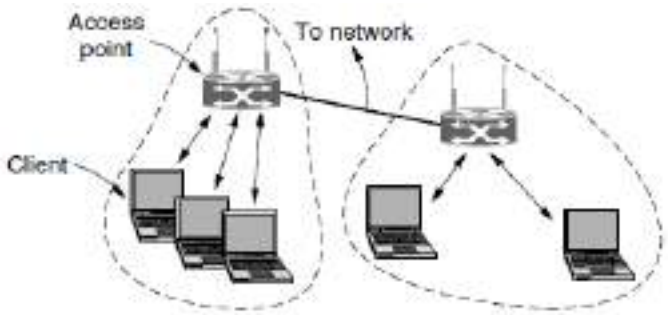
Wireless LANs are increasingly popular, and homes, offices, cafes, libraries, airports, zoos, and other public places are being outfitted with them to connect computers, PDAs, and smart phones to the Internet. Wireless LANs can also be used to let two or more nearby computers communicate without using the Internet. The main wireless LAN standard is 802.11 (Wi-Fi), which covers the physical and data-link layers. It is sometimes called wireless Ethernet. In some countries, including the United States, the public uses the term Wi-Fi (short for wireless fidelity) as a synonym for wireless LAN. Wi-Fi, however, is a wireless LAN that is certified by the Wi-Fi Alliance, a global, nonprofit industry association of more than 300 member companies devoted to promoting the growth of wireless LANs.

The 802.11 Architecture

802.11 networks can be used in two modes:

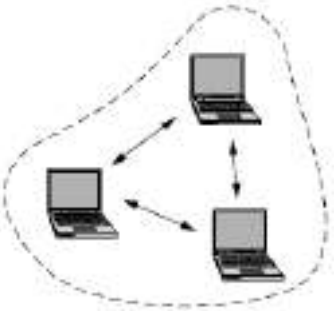
A- Infrastructure mode

The most popular mode is to connect clients, such as laptops and smart phones, to another network, such as a company intranet or the Internet. In infrastructure mode, each client is associated with an AP (Access Point) that is in turn connected to the other network. The client sends and receives its packets via the AP. This mode is shown in the figure below.



B- Ad hoc network

This mode is a collection of computers that are associated so that they can directly send frames to each other. There is no access point. Since Internet access is the main application for wireless, ad hoc networks are not very popular. (See the figure below)



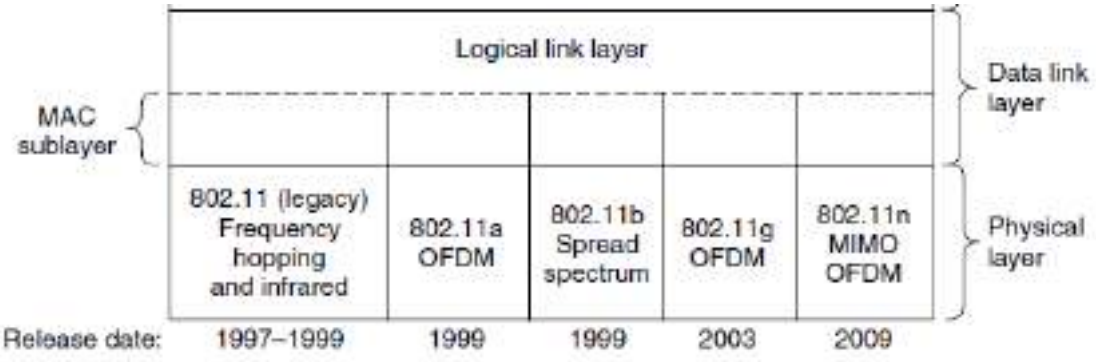
Protocol Stack

The protocol stack, shown in figure below, is the same for clients and APs.

1- **Physical layer** corresponds fairly well to the TCP/IP physical layer.

2- **The data link layer** in all the 802 protocols is split into two or more sublayers.

- **The MAC (Medium Access Control) sublayer** determines how the channel is allocated, that is, who gets to transmit next.
- **The LLC (Logical Link Control) sublayer**, whose job it is to hide the differences between the different 802 variants and make them indistinguishable as far as the network layer is concerned.



1- The 802.11 Physical Layer

All of the 802.11 techniques use short-range radios to transmit signals in either the 2.4-GHz or the 5-GHz ISM (Industrial, Scientific, and Medical) frequency bands, which defines three unlicensed bands in the three ranges 902–928 MHz, 2.400–4.835 GHz, and 5.725–5.850 GHz. These bands have the advantage of being unlicensed and hence freely available to any transmitter willing to meet some restrictions, such as radiated power of at most 1 W (though 50 mW is more typical for wireless LAN radios). Unfortunately, this fact is also known to the manufacturers of garage door openers, cordless phones, microwave ovens, and countless other devices, all of which compete with laptops for the same spectrum. The 2.4-GHz band tends to be more crowded than the 5-GHz band, so 5 GHz can be better for some applications even though it has shorter range due to the higher frequency.

802.11 physical layer techniques

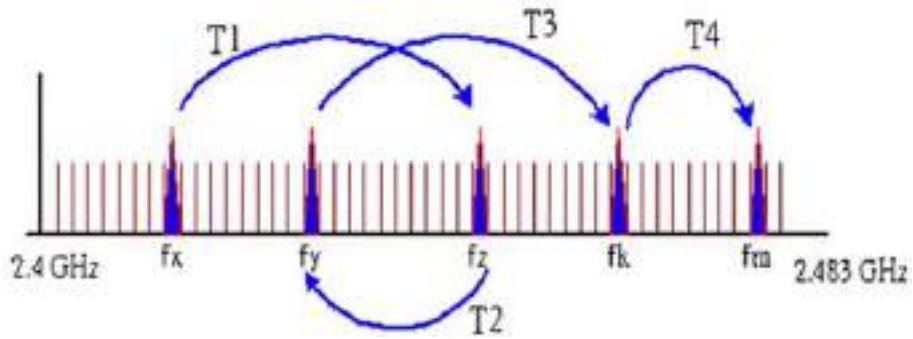
There are four techniques are use in 802.11 standard:

1- Frequency- hopping spread spectrum (FHSS)

- The 802.11 (legacy) method is based on FHSS.
- Operating in 2.4 GHz ISM band.
- Signal is broadband over random series of radio frequencies.
- Signal hops from frequency to frequency at fixed intervals.
- At receiver, hopping between frequencies in synchronization with transmitter, picks up message.

Advantages:

- Lower cost.
- Lower Power consumption.
- Most tolerant to signal interference.



2- Infrared (IR)

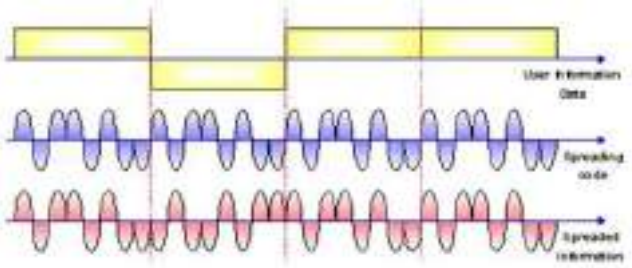
- Data Rate between 1 to 2 Mbps.
- Lowes range compared to spread spectrum (10 to 20 meter).
- Doesn't work outdoor.
- Lowest cost.

3- Direct-sequence spread spectrum (DSSS)

- The 802.11b method is based on DSSS.
- Operating in 2.4 GHz ISM band.
- Supports rates of 1, 2, 5.5, and 11 Mbps, though in practice the operating rate is nearly always 11 Mbps.
- Each bit in original signal is represented by multiple bits in the transmitted signal.
- Spreading code spreads signal across a wider frequency band.

Advantage

- Support higher data rate.
- More range than FHSS.

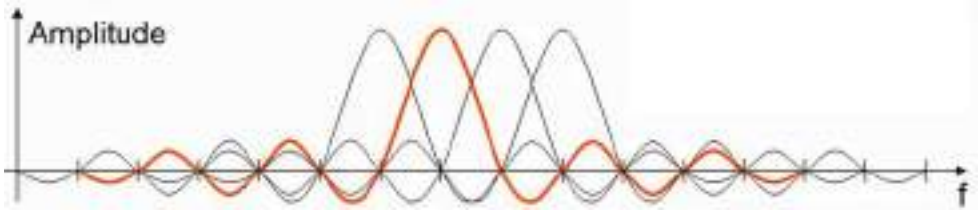


4- Orthogonal Frequency Division Multiplexing (OFDM)

- The 802.11a and 802.11g methods are based on OFDM.
- Both 802.11a and 802.11g can run at eight different rates, ranging from 6 to 54 Mbps.
- 802.11a operates in 5 GHz ISM band.
- 802.11g operates in 2.4 GHz ISM band.
- Parallel data transmission on several orthogonal subcarriers with lower rate.
- Maximum of one subcarrier frequency appears exactly at frequency where all other subcarriers equal zero.

Advantage

- No expensive filter.
- Better spectral efficiency than DSSS.



5- MIMO OFDM (802.11n)

- The IEEE committee began work on a high-throughput physical layer called 802.11n.
- It was ratified in 2009.
- The goal for 802.11n was throughput of at least 100 Mbps after all the wireless overheads were removed.
- 802.11n uses up to four antennas to transmit up to four streams of information at the same time.
- The signals of the streams interfere at the receiver, but they can be separated using MIMO (Multiple Input Multiple Output) communications techniques.

The 802.11 MAC Sublayer Protocol

The 802.11 MAC sublayer protocol is quite different from that of Ethernet, due to two factors that are fundamental to wireless communication:

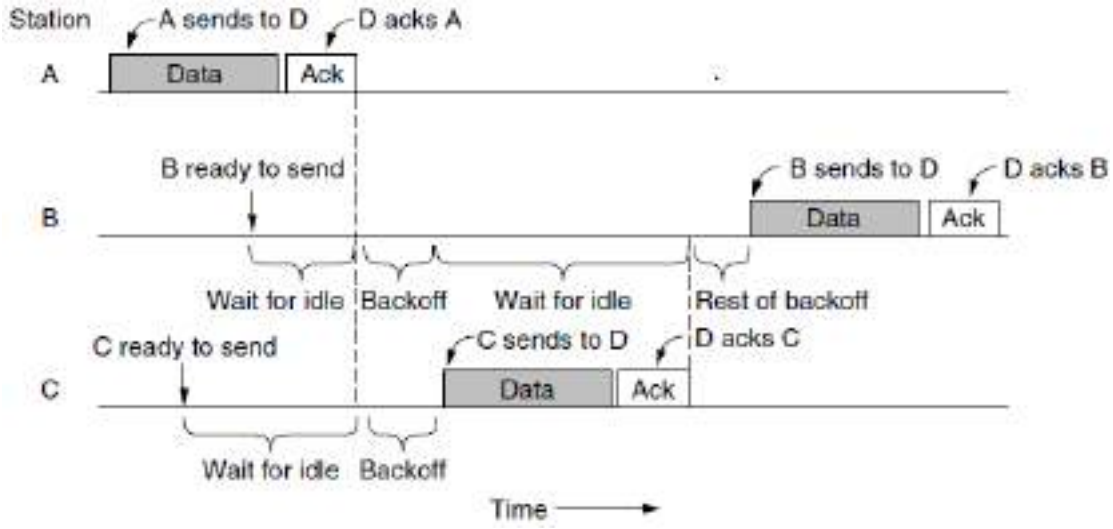
1- Radios are nearly always half duplex, meaning that they cannot transmit and listen for noise bursts at the same time on a single frequency. With Ethernet, a station just waits until the either goes silent and then starts transmitting. If it does not receive a noise burst back while transmitting the first 64 bytes, the frame has almost assuredly been delivered correctly. With wireless, this collision detection mechanism does not work.

Instead, 802.11 tries to avoid collisions with a protocol called **CSMA/CA (CSMA with Collision Avoidance).** This protocol is conceptually similar to Ethernet's CSMA/CD, with channel sensing before sending and exponential back off after collisions.

However, a station that has a frame to send starts with a random back-off. The number of slots to back-off is chosen in the range 0 to, say, 15 in the case of the OFDM physical layer. The station waits until the channel is idle, by sensing that there is no signal for a short period of time (called the **distributed inter-frame space DIFS**), and counts down idle slots, pausing when frames are sent. It sends its frame when the counter reaches 0. If the frame gets through, the destination immediately sends a short acknowledgement. Lack of an acknowledgement is inferred to indicate an error, whether a collision or otherwise. In this case, the sender doubles the back-off period and tries again, continuing with exponential back-off as in Ethernet until the frame has been successfully transmitted or the maximum number of retransmissions has been reached.

An example timeline is shown in figure below:

Station A is the first to send a frame. While A is sending, stations B and C become ready to send. They see that the channel is busy and wait for it to become idle. Shortly after A receives an acknowledgement, the channel goes idle. However, rather than sending a frame right away and colliding, B and C both perform a back-off. C picks a short back-off, and thus sends first. B pauses its countdown while it senses that C is using the channel, and resumes after C has received an acknowledgement. B soon completes its back-off and sends its frame.



Compared to Ethernet, there are two main differences:

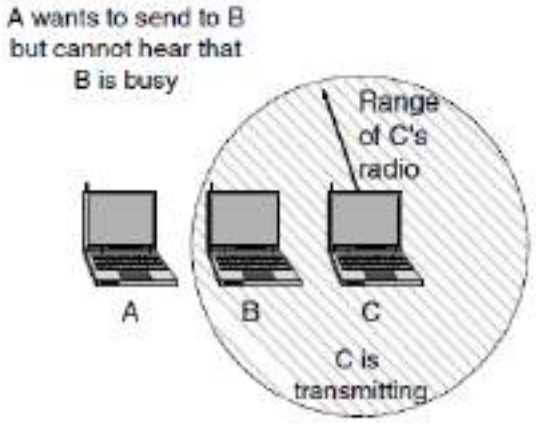
First, starting back-offs early helps to avoid collisions. This avoidance is worthwhile because collisions are expensive, as the entire frame is transmitted even if one occurs.

Second, acknowledgements are used to infer collisions because collisions cannot be detected.

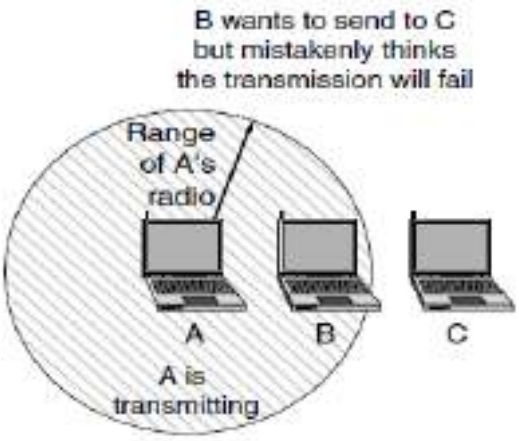
This mode of operation is called **DCF (Distributed Coordination Function)** because each station acts independently, without any kind of central control. The standard also includes an optional mode of operation called **PCF (Point Coordination Function)** in which the access point controls all activity in its cell, just like a cellular base station. However, PCF is not used in practice because there is normally no way to prevent stations in another nearby network from transmitting competing traffic.

2- **The transmission ranges of different stations may be different.** With a wire, the system is engineered so that all stations can hear each other. With the complexities of RF propagation this situation does not hold for wireless stations.

Consequently, situations such as **the hidden terminal problem** illustrated in figure below can arise. Since not all stations are within radio range of each other, transmissions going on in one part of a cell may not be received elsewhere in the same cell. In this example, station C is transmitting to station B. If A senses the channel, it will not hear anything and will falsely conclude that it may now start transmitting to B. This decision leads to a collision.



The inverse situation is **the exposed terminal problem**, illustrated in figure below. Here, B wants to send to C, so it listens to the channel. When it hears a transmission, it falsely concludes that it may not send to C, even though A may in fact be transmitting to D (not shown). This decision wastes a transmission opportunity.

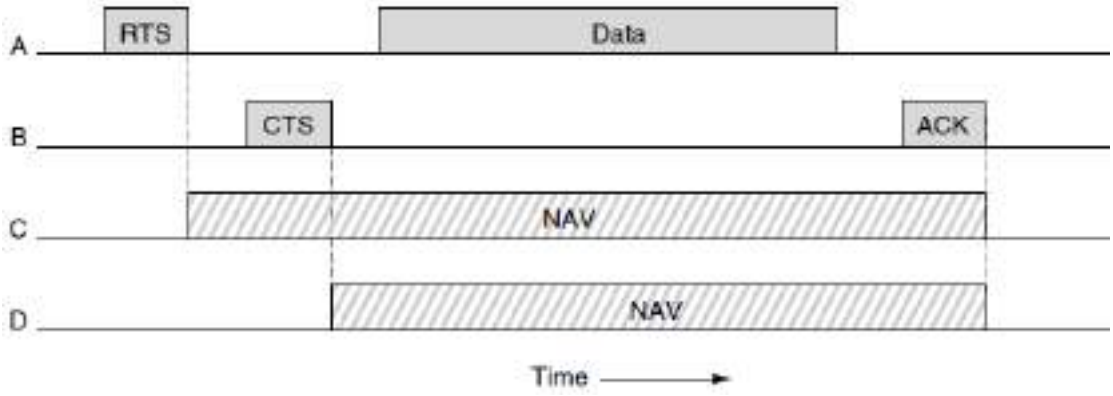


To reduce ambiguities about which station is sending, 802.11 defines channel sensing to consist of both **physical sensing and virtual sensing.** Physical sensing simply checks the medium to see if there is a valid signal. With virtual sensing, each station keeps a logical record of when the channel is

in use by tracking the **NAV (Network Allocation Vector)**. Each frame carries a NAV field that says how long the frame will take to complete.

Stations that overhear this frame know that the channel will be busy for the period indicated by the NAV, regardless of whether they can sense a physical signal. For example, the NAV of a data frame includes the time needed to send an acknowledgement. All stations that hear the data frame will defer during the acknowledgement period, whether or not they can hear the acknowledgement.

An optional **RTS/CTS** (Request To Send/Clear To Send) mechanism uses the NAV to prevent terminals from sending frames at the same time as hidden terminals. It is shown in figure below.



In this example, A wants to send to B. C is a station within range of A (and possibly within range of B, but that does not matter). D is a station within range of B but not within range of A. The protocol starts when A decides it wants to send data to B. A begins by sending an RTS frame to B to request permission to send it a frame. If B receives this request, it answers with a CTS frame to indicate that the channel is clear to send. Upon receipt of the CTS, A sends its frame and starts an ACK timer. Upon correct receipt of the data frame, B responds with an ACK frame, completing the exchange. If A's ACK timer expires before the ACK gets back to it, it is treated as a collision and the whole protocol is run again after a back-off.

Now let us consider this exchange from the viewpoints of C and D. C is within range of A, so it may receive the RTS frame. If it does, it realizes that someone is going to send data soon. From the information provided in the RTS request, it can estimate how long the sequence will take, including the final ACK. So, for the good of all, it desists from transmitting anything until the exchange is completed.

It does so by updating its record of the NAV to indicate that the channel is busy, as shown in figure above. D does not hear the RTS, but it does hear the CTS, so it also updates its NAV. Note that the NAV signals are not transmitted; they are just internal reminders to keep quiet for a certain period of time.

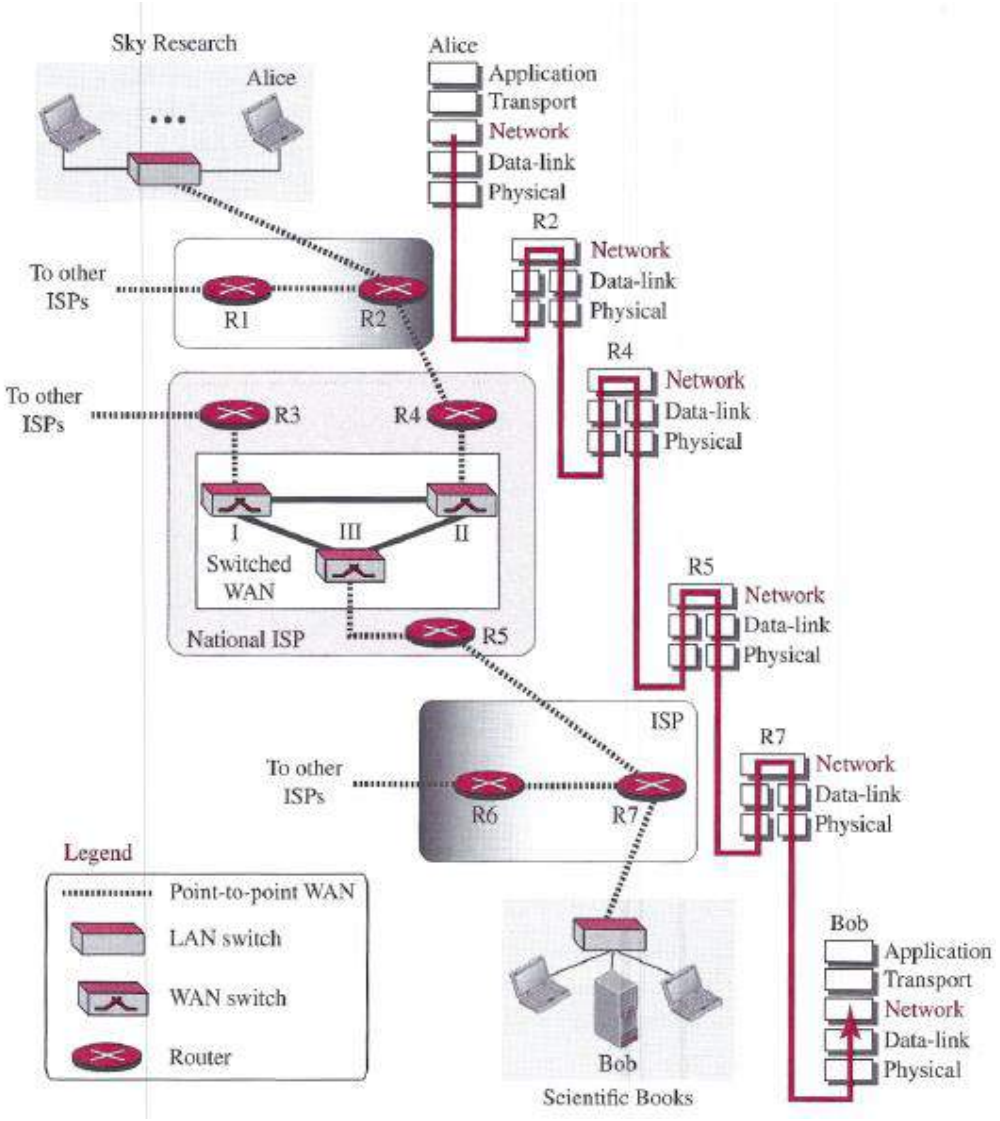
Computer Networks

THE NETWORK LAYER

The network layer is concerned with getting packets from the source all the way to the destination. Getting to the destination may require making many hops at intermediate routers along the way. This function clearly contrasts with that of the data link layer, which has the more modest goal of just moving frames from one end of a wire to the other. Thus, the network layer is the lowest layer that deals with end-to-end transmission. It provides services to the transport layer and receives services from the data-link layer.

NETWORK LAYER SERVICES

The figure below shows the communication between Alice and Bob at the network layer. This is the same scenario we used in previously to show the communication at the physical and the data-link layers.



As the figure shows, the network layer is involved at the source host, destination host, and all routers in the path (R2, R4, RS, and R7). At the source host (Alice), the network layer accepts a packet from a transport layer, encapsulates the packet in a datagram, and delivers the packet to the data-link layer. At the destination host (Bob), the datagram is decapsulated, and the packet is extracted and delivered to the corresponding transport layer. Although the source and destination hosts are involved in all five layers of the TCP/IP suite, the routers use three layers if they are routing packets only; however, they may need the transport and application layers for control purposes. A router in the path is normally shown with two data-link layers and two physical layers, because it receives a packet from one network and delivers it to another network.

1- Packetizing

The first duty of the network layer is definitely **packetizing**: encapsulating the payload (data received from upper layer) in a network-layer packet at the source and decapsulating the payload from the network-layer packet at the destination. In other words, one duty of the network layer is to carry a payload from the source to the destination without changing it or using it. The network layer is doing the service of a carrier such as the postal office, which is responsible for delivery of packages from a sender to a receiver without changing or using the contents.

The source host receives the payload from an upper-layer protocol, adds a header that contains the source and destination addresses and some other information that is required by the network-layer protocol (as discussed later) and delivers the packet to the data-link layer. The source is not allowed to change the content of the payload unless it is too large for delivery and needs to be fragmented.

The destination host receives the network-layer packet from its data-link layer, decapsulates the packet, and delivers the payload to the corresponding upper-layer protocol. If the packet is fragmented at the source or at routers along the path, the network layer is responsible for waiting until all fragments arrive, reassembling them, and delivering them to the upper-layer protocol.

The routers in the path are not allowed to decapsulate the packets they received unless the packets need to be fragmented. The routers are not allowed to change source and destination addresses either. They just inspect the addresses for the purpose of forwarding the packet to the next network on the path. However, if a packet is fragmented, the header needs to be copied to all fragments and some changes are needed.

2- Routing and Forwarding

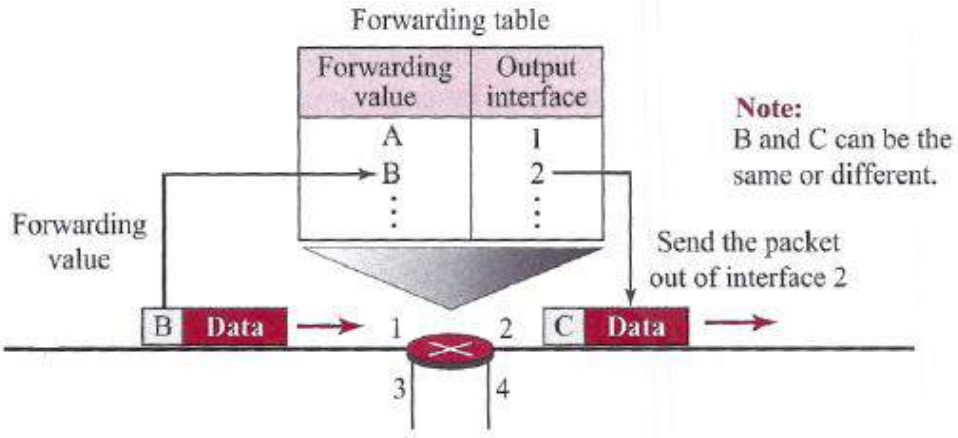
Other duties of the network layer, which are as important as the first, are routing and forwarding, which are directly related to each other.

Routing

The network layer is responsible for routing the packet from its source to the destination. A physical network is a combination of networks (LANs and WANs) and routers that connect them. This means that there is more than one route from the source to the destination. The network layer is responsible for finding the best one among these possible routes. The network layer needs to have some specific strategies for defining the best route. In the Internet today, this is done by running some routing protocols to help the routers coordinate their knowledge about the neighborhood and to come up with consistent tables to be used when a packet arrives. The routing protocols should be run before any communication occurs.

Forwarding

If routing is applying strategies and running some routing protocols to create the decision-making tables for each router, forwarding can be defined as the action applied by each router when a packet arrives at one of its interfaces. The decision-making table that a router normally uses for applying this action is sometimes called the forwarding table and sometimes the routing table. When a router receives a packet from one of its attached networks, it needs to forward the packet to another attached network (in unicast routing) or to some attached networks (in multicast routing). To make this decision, the router uses a piece of information in the packet header, which can be the destination address or a label, to find the corresponding output interface number in the forwarding table. The figure below shows the idea of the forwarding process in a router.



3- Congestion Control

Another issue in a network layer protocol is congestion control. Congestion in the network layer is a situation in which too many datagrams are present in an area of the Internet. Congestion may occur if the number of datagrams sent by source computers is beyond the capacity of the network or routers. In this situation, some routers may drop some of the datagrams. However, as more datagrams are dropped, the situation may become worse because, due to the error control mechanism at the upper layers, the sender may send duplicates of the lost packets. If the congestion continues, sometimes a situation may reach a point where the system collapses and no datagrams are delivered. We discuss congestion control at the network layer later in the chapter although it is not implemented in the Internet.

PACKET SWITCHING

From the discussion of routing and forwarding in the previous section, we infer that a kind of switching occurs at the network layer. A router, in fact, is a switch that creates a connection between an input port and an output port (or a set of output ports), just as an electrical switch connects the input to the output to let electricity flow.

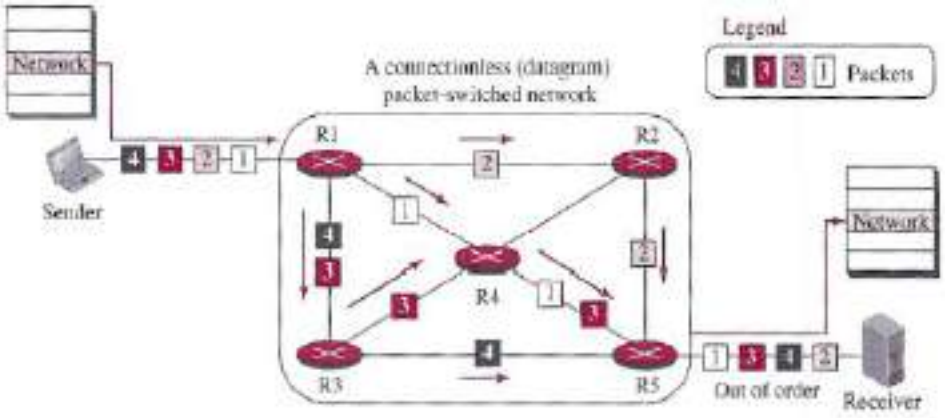
Although in data communication switching techniques are divided into two broad categories, circuit switching and packet switching, only packet switching is used at the network layer because the unit of data at this layer is a packet. Circuit switching is mostly used at the physical layer.

At the network layer, a message from the upper layer is divided into manageable packets and each packet is sent through the network. The source of the message sends the packets one by one; the destination of the message receives the packets one by one. The destination waits for all packets belonging to the same message to arrive before delivering the message to the upper layer. The connecting devices in a packet-switched network still need to decide how to route the packets to the final destination. Today, a packet-switched network can use two different approaches to route the packets: the datagram approach and the virtual circuit approach.

1- Datagram Approach: Connectionless Service

When the Internet started, to make it simple, the network layer was designed to provide a connectionless service in which the network layer protocol treats each packet independently, with each packet having no relationship to any other packet. The idea was that the network layer is only

responsible for delivery of packets from the source to the destination. In this approach, the packets in a message may or may not travel the same path to their destination. The figure below shows the idea.



When the network layer provides a connectionless service, each packet traveling in the Internet is an independent entity; there is no relationship between packets belonging to the same message. A packet belonging to a message may be followed by a packet belonging to the same message or to a different message. A packet may be followed by a packet coming from the same or from a different source.

Each packet is routed based on the information contained in its header: source and destination addresses. The destination address defines where it should go; the source address defines where it comes from. The router in this case routes the packet based only on the destination address. The source address may be used to send an error message to the source if the packet is discarded.

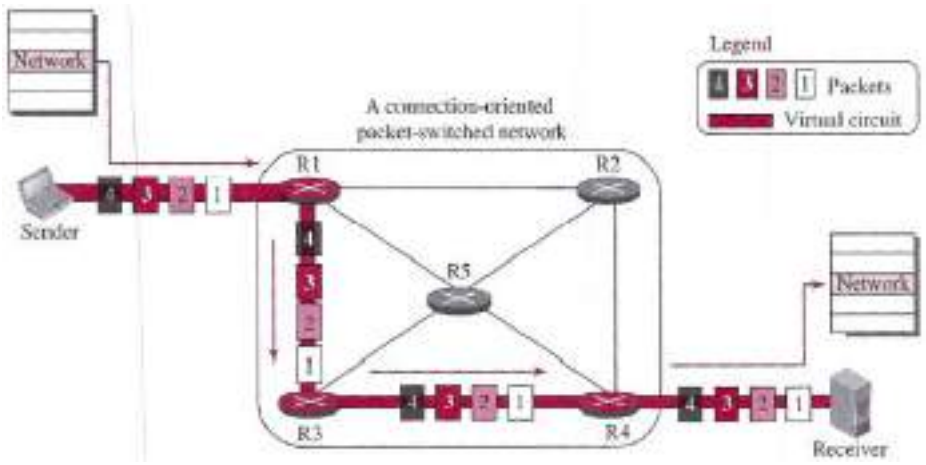
2- Virtual-Circuit Approach: Connection-Oriented Service

In a connection-oriented service (also called virtual-circuit approach), there is a relationship between all packets belonging to a message. Before all datagrams in a message can be sent, a virtual connection should be set up to define the path for the datagrams. After connection setup, the datagrams can all follow the same path. In this type of service, not only must the packet contain the source and destination addresses, it must also contain a flow label, a virtual circuit identifier that defines the virtual path the packet should follow.

Although it looks as though the use of the label may make the source and destination addresses unnecessary during the data transfer phase, parts of the Internet at the network layer still keep these addresses.

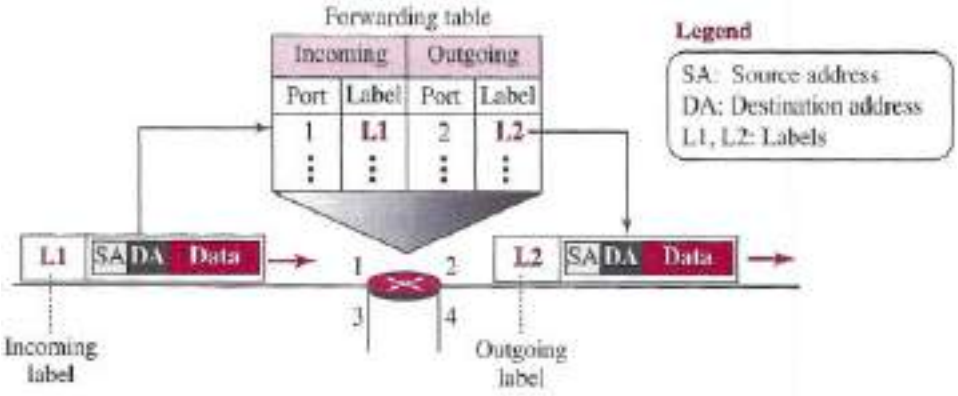
- One reason is that part of the packet path may still be using the connectionless service.
- Another reason is that the protocol at the network layer is designed with these addresses, and it may take a while before they can be changed.

The figure below shows the concept of connection-oriented service.



Each packet is forwarded based on the label in the packet. To follow the idea of connection-oriented design to be used in the Internet, we assume that the packet has a label when it reaches the router. The figure below shows the idea. In this case, the forwarding decision is based on the value of the label, or virtual circuit identifier, as it is sometimes called. To create a connection-oriented service, a three-phase process is used: setup, data transfer, and teardown.

- In the setup phase, the source and destination addresses of the sender and receiver are used to make table entries for the connection-oriented service.
- In the teardown phase, the source and destination inform the router to delete the corresponding entries.
- Data transfer occurs between these two phases.

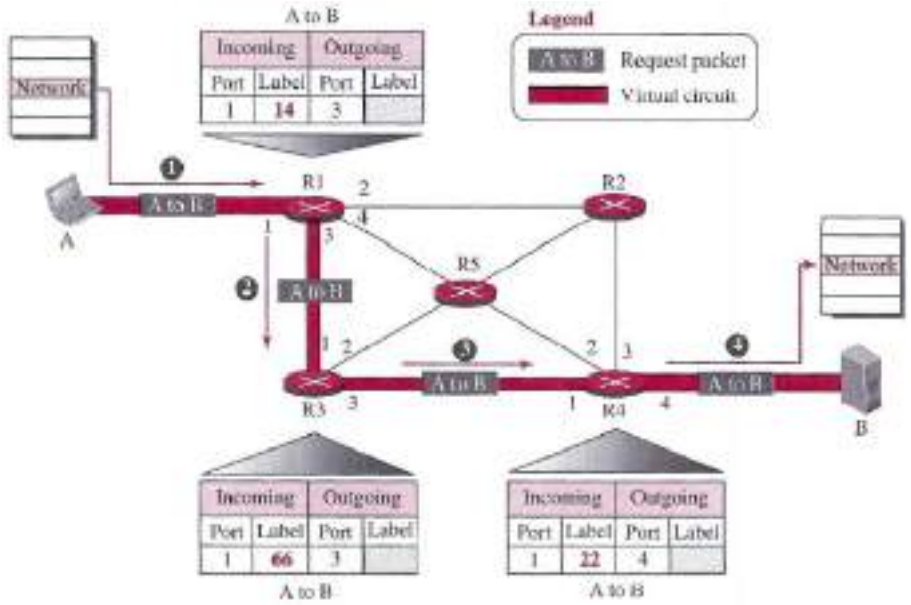


1- Setup Phase

In the setup phase, a router creates an entry for a virtual circuit. For example, suppose source A needs to create a virtual circuit to destination B. Two auxiliary packets need to be exchanged between the sender and the receiver: the request packet and the acknowledgment packet.

Request packet

A request packet is sent from the source to the destination. This auxiliary packet carries the source and destination addresses. The figure below shows the process.



1- Source A sends a request packet to router R1.

2- Router R1 receives the request packet. It knows that a packet going from A to B goes out through port 3. The router creates an entry in its table for this virtual circuit, but it is only able to fill three of the four columns. The router assigns the incoming port (1) and chooses an available incoming label (14) and the outgoing port (3). It does not yet know the outgoing label, which will be found during the acknowledgment step. The router then forwards the packet through port 3 to router R3.

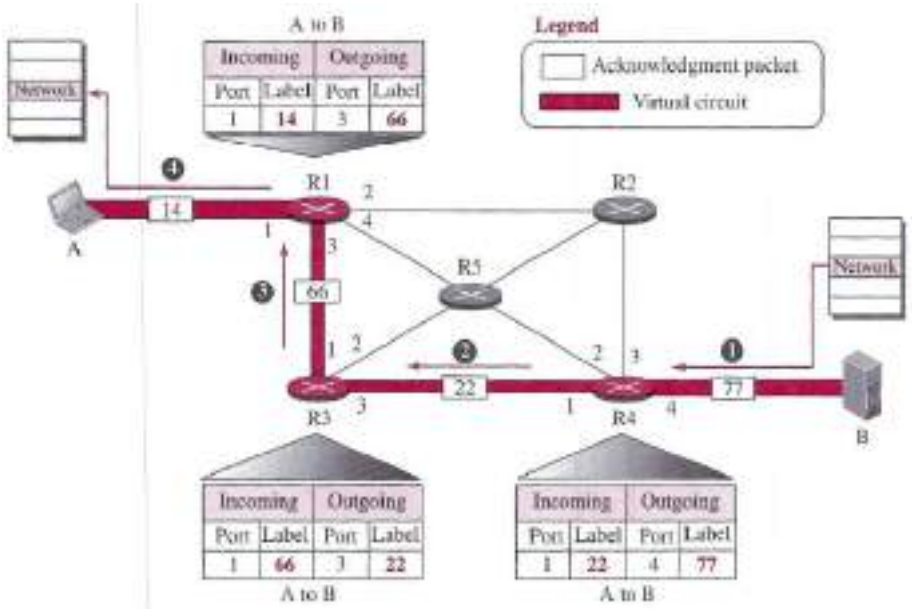
3- Router R3 receives the setup request packet. The same events happen here as at router R1; three columns of the table are completed: in this case, incoming port (1), incoming label (66), and outgoing port (3).

4- Router R4 receives the setup request packet. Again, three columns are completed: incoming port (1), incoming label (22), and outgoing port (4).

5- Destination B receives the setup packet, and if it is ready to receive packets from A, it assigns a label to the incoming packets that come from A, in this case 77. This label lets the destination know that the packets come from A, and not from other sources.

Acknowledgment Packet

A special Packet, called the acknowledgment packet, completes the entries in the switching tables. The figure below shows the process.



1- The destination sends an acknowledgment to router R4. The acknowledgment carries the global source and destination addresses so the router knows which entry in the table is to be completed. The packet also carries label 77, chosen by the destination as the incoming label for packets from A. Router R4 uses this label to complete the outgoing label column for this entry. Note that 77 is the incoming label for destination B, but the outgoing label for router R4.

2- Router R4 sends an acknowledgment to router R3 that contains its incoming label in the table, chosen in the setup phase. Router R3 uses this as the outgoing label in the table.

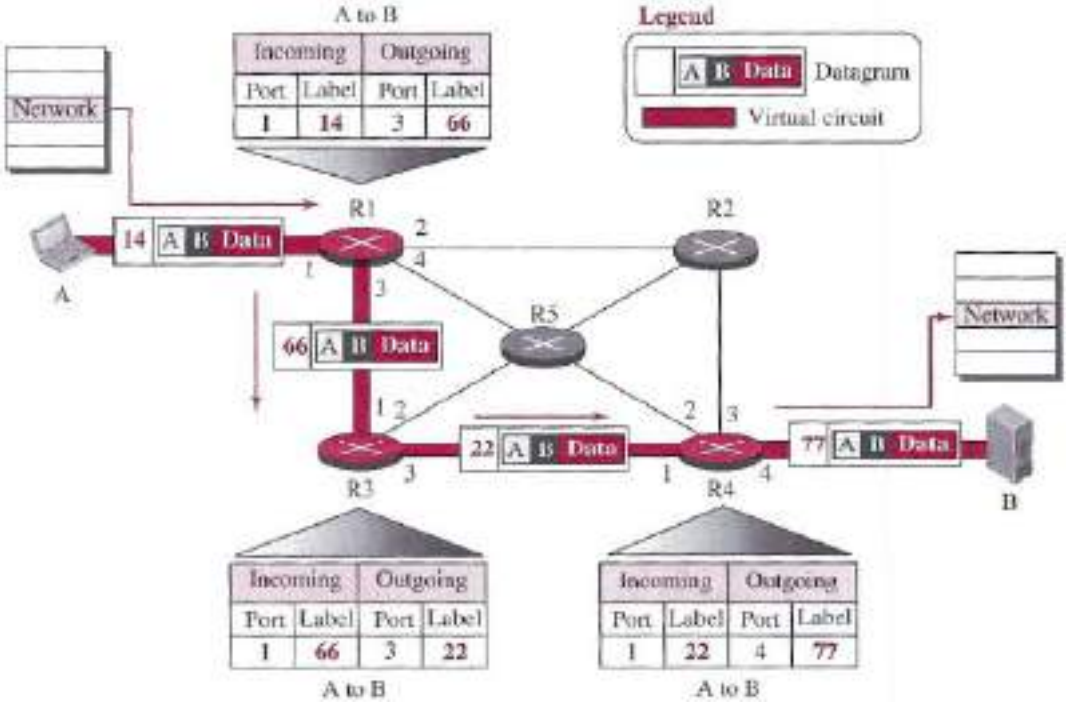
3- Router R3 sends an acknowledgment to router R1 that contains its incoming label in the table, chosen in the setup phase. Router RI uses this as the outgoing label in the table.

4- Finally, router R1 sends an acknowledgment to source A that contains its incoming label in the table, chosen in the setup phase.

5- The source uses this as the outgoing label for the data packets to be sent to destination B.

2- Data Transfer Phase

The second phase is called the data-transfer phase. After all routers have created their forwarding table for a specific virtual circuit, then the network-layer packets belonging to one message can be sent one after another. In the figure below, we show the flow of a single packet, but the process is the same for 1, 2, or 100 packets. The source computer uses the label 14, which it has received from router R1 in the setup phase. Router R1 forwards the packet to router R3, but changes the label to 66. Router R3 forwards the packet to router R4, but changes the label to 22. Finally, router R4 delivers the packet to its final destination with the label 77. All the packets in the message follow the same sequence of labels, and the packets arrive in order at the destination.



3- Teardown Phase

In the teardown phase, source A, after sending all packets to B, sends a special packet called a teardown packet. Destination B responds with a confirmation packet. All routers delete the corresponding entries from their tables.

CONGESTION CONTROL

Congestion control refers to techniques and mechanisms that can either prevent congestion before it happens or remove congestion after it has happened. In general, we can divide congestion control mechanisms into two broad categories: open-loop congestion control (prevention) and closed-loop congestion control (removal).

1- Open-Loop Congestion Control

In open-loop congestion control, policies are applied to prevent congestion before it happens. In these mechanisms, congestion control is handled by either the source or the destination. We give a brief list of policies that can prevent congestion.

Retransmission Policy

Retransmission is sometimes unavoidable. If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted. Retransmission in general may increase congestion in the network. However, a good retransmission policy can prevent congestion. The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion.

Window Policy

The type of window at the sender may also affect congestion. The Selective Repeat window is better than the Go-Back-N window for congestion control. In the Go-Back-N window, when the timer for a packet times out, several packets may be resent, although some may have arrived safe and sound at the receiver. This duplication may make the congestion worse. The Selective Repeat window, on the other hand, tries to send the specific packets that have been lost or corrupted.

Acknowledgment Policy

The acknowledgment policy imposed by the receiver may also affect congestion. If the receiver does not acknowledge every packet it receives, it may slow down the sender and help prevent congestion. Several approaches are used in this case. A receiver may send an acknowledgment only if it has a packet to be sent or a special timer expires. A receiver may decide to acknowledge only N packets at a time. We need to know that the acknowledgments are also part of the load in a network. Sending fewer acknowledgments means imposing less load on the network.

Discarding Policy

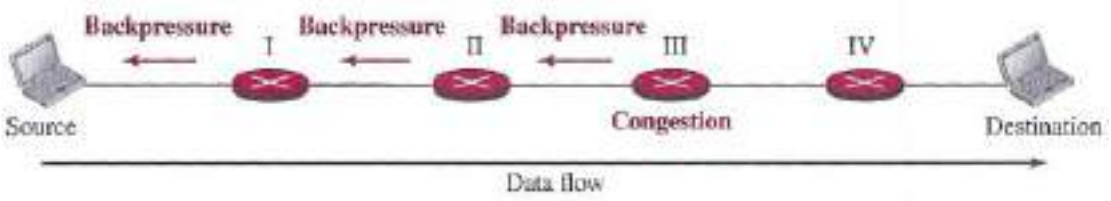
A good discarding policy by the routers may prevent congestion and at the same time may not harm the integrity of the transmission. For example, in audio transmission, if the policy is to discard less sensitive packets when congestion is likely to happen, the quality of sound is still preserved and congestion is prevented or alleviated.

2- Closed-Loop Congestion Control

Closed-loop congestion control mechanisms try to alleviate congestion after it happens. Several mechanisms have been used by different protocols. We describe a few of them here.

Backpressure

The technique of backpressure refers to a congestion control mechanism in which a congested node stops receiving data from the immediate upstream node or nodes. This may cause the upstream node or nodes to become congested, and they, in turn, reject data from their upstream node or nodes, and so on. Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source. The backpressure technique can be applied only to virtual circuit networks, in which each node knows the upstream node from which a flow of data is coming. The figure below shows the idea of backpressure.



Node III in the figure has more input data than it can handle. It drops some packets in its input buffer and informs node II to slow down. Node II, in turn, may be congested because it is slowing down the output flow of data. If node II is congested, it informs node I to slow down, which in turn may create congestion. If so, node I informs the source of data to slow down. This, in time, alleviates the congestion. Note that the pressure on node III is moved backward to the source to remove the congestion.

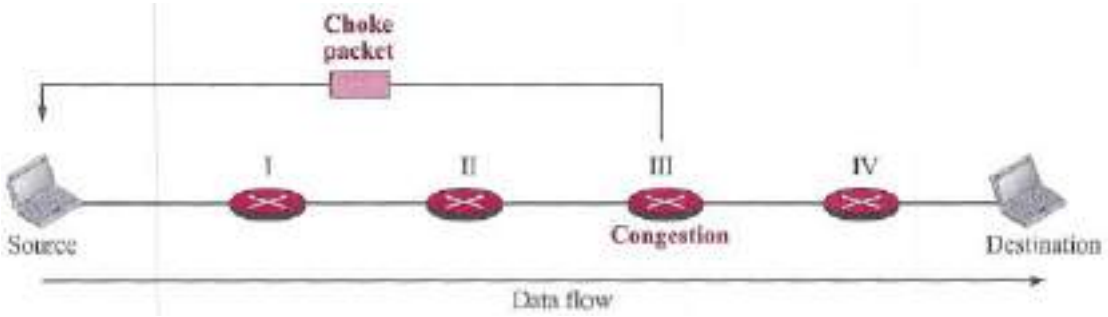
It is important to stress that this type of congestion control can only be implemented in virtual-circuit. The technique cannot be implemented in a datagram network, in which a node (router) does not have the slightest knowledge of the upstream router.

Choke Packet

A choke packet is a packet sent by a node to the source to inform it of congestion. Note the difference between the backpressure and choke-packet methods.

- In backpressure, the warning is from one node to its upstream node, although the warning may eventually reach the source station.
- In the choke-packet method, the warning is from the router, which has encountered congestion, directly to the source station. The intermediate nodes through which the packet has traveled are not warned.

The figure below shows the idea of a choke packet.



Implicit Signaling

In implicit signaling, there is no communication between the congested node or nodes and the source. The source guesses that there is congestion somewhere in the network from other symptoms. For example, when a source sends several packets and there is no acknowledgment for a while, one assumption is that the network is congested. The delay in receiving an acknowledgment is interpreted as congestion in the network; the source should slow down.

Explicit Signaling

The node that experiences congestion can explicitly send a signal to the source or destination. The explicit-signaling method, however, is different from the choke-packet method.

- In the choke-packet method, a separate packet is used for this purpose
- In the explicit-signaling method, the signal is included in the packets that carry data.

IPv4 ADDRESSES

- The identifier used in the network layer of the TCP/IP protocol suite to identify the connection of each device to the Internet is called the Internet address or IP address.
- An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a host or a router to the Internet.
- The IP address is the address of the connection, not the host or the router, because if the device is moved to another network, the IP address may be changed.
- IPv4 addresses are unique in the sense that each address defines one, and only one, connection to the Internet.
- If a device has two connections to the Internet, via two networks, it has two IPv4 addresses.
- IPv4 addresses are universal in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet.

Address Space

- A protocol like IPv4 that defines addresses has an address space.
- An address space is the total number of addresses used by the protocol. If a protocol uses b bits to define an address, the address space is 2^b because each bit can have two different values (0 or 1)
- IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (more than four billion).
- If there were no restrictions, more than 4 billion devices could be connected to the Internet.

Notation

There are three common notations to show an IPv4 address:

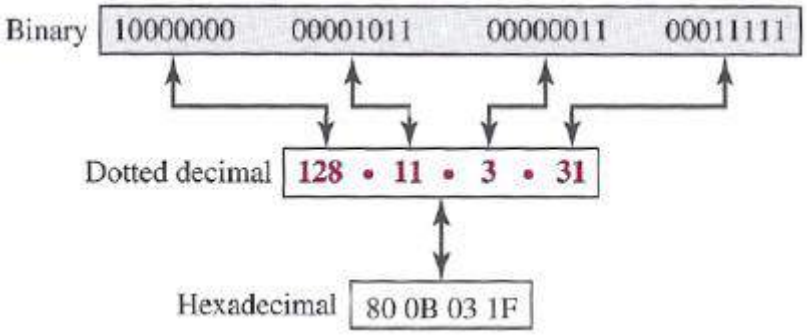
1- Binary notation (base 2): In binary notation, an IPv4 address is displayed as 32 bits. To make the address more readable, one or more spaces are usually inserted between each octet (8 bits). Each octet is often referred to as a byte.

2- Dotted-decimal notation (base 256): To make the IPv4 address more compact and easier to read, it is usually written in decimal form with a decimal point (dot) separating the bytes. This format is referred to as dotted-decimal notation. Note that because each byte (octet) is only 8 bits, each number in the dotted-decimal notation is between 0 and 255.

3- Hexadecimal notation (base 16)

We sometimes see an IPv4 address in hexadecimal notation. Each hexadecimal digit is equivalent to four bits. This means that a 32-bit address has 8 hexadecimal digits. This notation is often used in network programming.

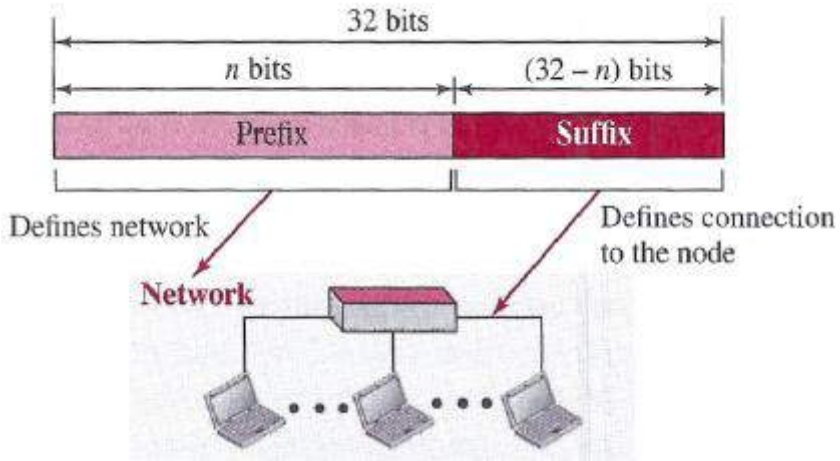
The figure below shows an IP address in the three discussed notations.



Hierarchy in Addressing

- In any communication network that involves delivery, such as a telephone network or a postal network, the addressing system is hierarchical.
- In a postal network, the postal address (mailing address) includes the country, state, city, street, house number, and the name of the mail recipient. Similarly, a telephone number is divided into the country code, area code, local exchange, and the connection.
- A 32-bit IPv4 address is also hierarchical, but divided only into two parts. The first part of the address, called the prefix, defines the network; the second part of the address, called the suffix, defines the node (connection of a device to the Internet).

The figure below shows the prefix and suffix of a 32-bit IPv4 address. The prefix length is n bits and the suffix length is (32 − n) bits.

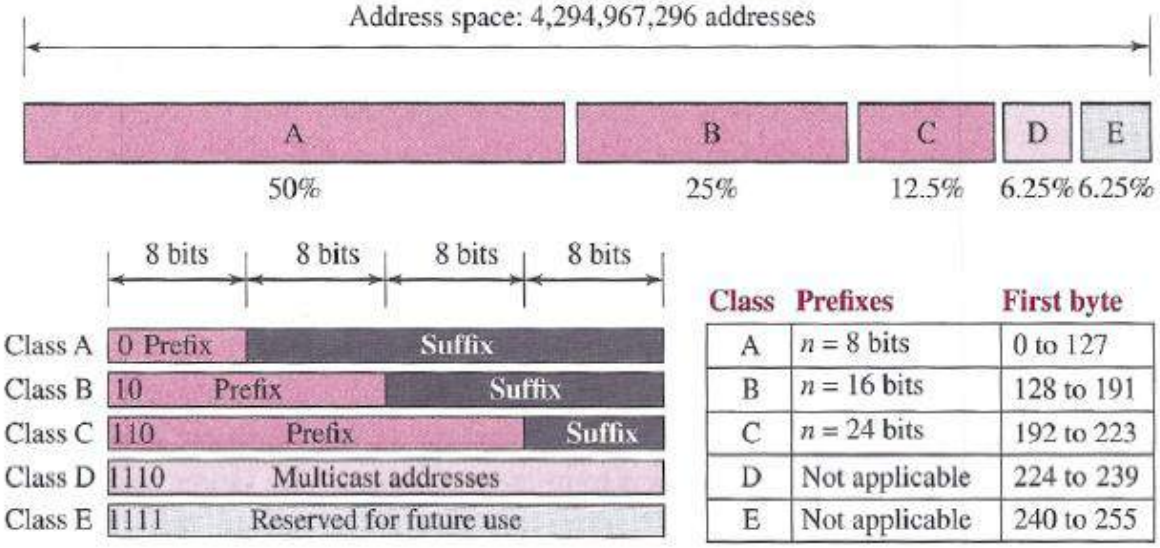


A prefix can be fixed length or variable length. The network identifier in the IPv4 was first designed as a fixed-length prefix. This scheme, which is now obsolete, is referred to as classful addressing.

The new scheme, which is referred to as classless addressing, uses a variable-length network prefix.

1- Classful Addressing

- When the Internet started, an IPv4 address was designed with a fixed-length prefix, but to accommodate both small and large networks, three fixed-length prefixes were designed instead of one (n = 8, n = 16, and n = 24).
- The whole address space was divided into five classes (class A, B, C, D, and E), as shown in the figure below.



Class	Prefixes	First byte
A	$n = 8$ bits	0 to 127
B	$n = 16$ bits	128 to 191
C	$n = 24$ bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

This scheme is referred to as classful addressing. Although classful addressing belongs to the past, it helps us to understand classless addressing

- In class A, the network length is 8 bits, but since the first bit, which is 0, defines the class, we can have only seven bits as the network identifier. This means there are only $2^7 = 128$ networks in the world that can have a class A address.
- In class B, the network length is 16 bits, but since the first two bits, which are $(10)_2$, define the class, we can have only 14 bits as the network identifier. This means there are only $2^{14} = 16{,}384$ networks in the world that can have a class B address.
- All addresses that start with $(110)_2$, belong to class C. In class C, the network length is 24 bits, but since three bits define the class, we can have only 21 bits as the network identifier, this means there are $2^{21} = 2{,}097{,}152$ networks in the world that can have a class c address.

- Class D is not divided into prefix and suffix. It is used for multicast addresses.
- All addresses that start with 1111 in binary belong to class E. As in Class D, Class E is not divided into prefix and suffix and is used as reserve.

Address Depletion

The reason that classful addressing has become obsolete is address depletion. Since the addresses were not distributed properly, the Internet was faced with the problem of the addresses being rapidly used up, resulting in no more addresses available for organizations and individuals that needed to be connected to the Internet. To understand the problem:

- let us think about class A. This class can be assigned to only 128 organizations in the world, but each organization needs to have a single network (seen by the rest of the world) with $16{,}777{,}216$ nodes (computers in this single network). Since there may be only a few organizations that are this large, most of the addresses in this class were wasted (unused).
- Class B addresses were designed for midsize organizations, but many of the addresses in this class also remained unused.
- Class C addresses have a completely different flaw in design. The number of addresses that can be used in each network (256) was so small that most companies were not comfortable using a block in this address class.
- Class E addresses were almost never used, wasting the whole class.

Advantage of Classful Addressing

Although classful addressing had several problems and became obsolete, it had one advantage: Given an address, we can easily find the class of the address and, since the prefix length for each class is fixed, we can find the prefix length immediately. In other words, the prefix length in classful addressing is inherent in the address; no extra information is needed to extract the prefix and the suffix.

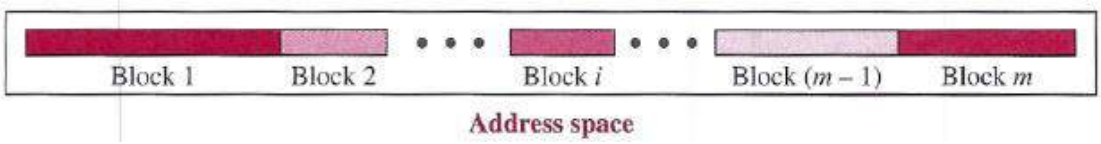
2- Classless Addressing

- With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution.
- The larger address space, however, requires that the length of IP addresses also be increased, which means the format of the IP packets needs to be changed. Although the long-range solution has already been devised and is called IPv6.
- A short-term solution was also devised to use the same address space but to change the distribution of addresses to provide a fair share to each organization.
- The short-term solution still uses IPv4 addresses, but it is called classless addressing. In other words, the class privilege was removed from the distribution to compensate for the address depletion.

There was another motivation for classless addressing:

- During the 1990s, Internet Service Providers (ISPs) came into prominence. An ISP is an organization that provides Internet access for individuals, small businesses, and midsize organizations that do not want to create an Internet site and become involved in providing Internet services (such as electronic mail) for their employees. An ISP can provide these services. An ISP is granted a large range of addresses and then subdivides the addresses (in groups of 1, 2, 4, 8, 16, and so on), giving a range of addresses to a household or a small business. The customers are connected via a dial-up modem, DSL, or cable modem to the ISP. However, each customer needs some IPv4 addresses.
- In 1996, the Internet authorities announced a new architecture called classless addressing.
- In classless addressing, variable-length blocks are used that belong to no classes.
- We can have a block of 1 address, 2 addresses, 4 addresses, 128 addresses, and so on.
- In classless addressing, the whole address space is divided into variable length blocks.
- The prefix in an address defines the block (network); the suffix defines the node (device).
- Theoretically, we can have a block of $2^0, 2^1, 2^2, \ldots, 2^{32}$ addresses.
- One of the restriction is that the number of addresses in a block needs to be a power of 2. An organization can be granted one block of addresses.

The figure below shows the division of the whole address space into non-overlapping blocks

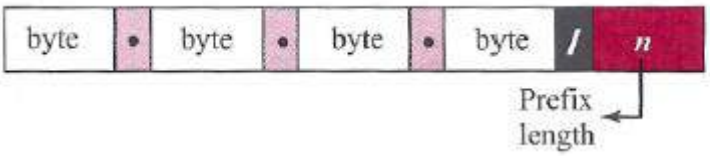


- Unlike classful addressing, the prefix length in classless addressing is variable.
- We can have a prefix length that ranges from 0 to 32.
- The size of the network is inversely proportional to the length of the prefix. A small prefix means a larger network; a large prefix means a smaller network.
- We need to emphasize that the idea of classless addressing can be easily applied to classful addressing. An address in class A can be thought of as a classless address in which the prefix length is 8. An address in class B can be thought of as a classless address in which the prefix is 16, and so on. In other words, classful addressing is a special case of classless addressing.

Prefix Length: Slash Notation

- The first question that we need to answer in classless addressing is how to find the prefix length if an address is given.
- Since the prefix length is not inherent in the address, we need to separately give the length of the prefix.
- In this case, the prefix length, n, is added to the address, separated by a slash.
- The notation is informally referred to as slash notation and formally as classless inter-domain routing or CIDR (pronounced cider) strategy.
- In other words, an address in classless addressing does not define the block or network to which the address belongs; we need to give the prefix length also.

An address in classless addressing can then be represented as shown in the figure below.



Examples:
12.24.76.8/**8**
23.14.67.92/**12**
220.8.24.255/**25**

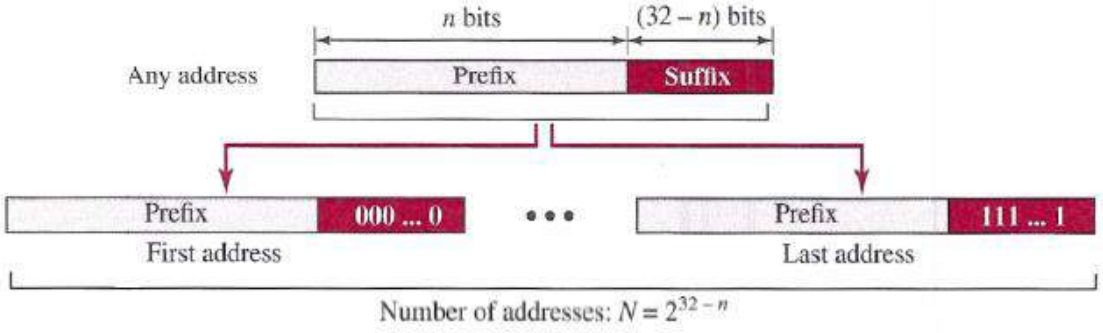
Extracting Information from an Address

Given any address in the block, we normally like to know three pieces of information about the block to which the address belongs: the number of addresses, the first address in the block, and the last address. Since the value of prefix length, n, is given, we can easily find these three pieces of information, as shown in figure below.

1- The number of addresses in the block is found as $N = 2^{32-n}$

2- To find the first address, we keep the n leftmost bits and set the $(32 - n)$ rightmost bits all to 0s.

3- To find the last address, we keep the n leftmost bits and set the $(32 - n)$ rightmost bits all to 1s.



Example: A classless address is given as 167.199.170.82/27. We can find the above three pieces of information as follows:

167.199.170.82/27

10100111 11000111 10101010 01010010

1- The number of addresses in the network is $2^{32-27} = 2^5 = 32$ addresses.

2- The first address can be found by keeping the first 27 bits and changing the rest of the bits to 0s.

10100111 11000111 10101010 01000000

167.199.170.64/27

3- The last address can be found by keeping the first 27 bits and changing the rest of the bits to 1s.

10100111 11000111 10101010 01011111

167.199.170.95/27

Address (Subnet) Mask

- Another way to find the first and last addresses in the block is to use the address mask.
- The address mask is a 32-bit number in which the n leftmost bits are set to 1s and the rest of the bits (32 − n) are set to 0s.
- A computer can easily find the address mask because it is the complement $(2^{32-n}-1)$.
- The reason for defining a mask in this way is that it can be used by a computer program to extract the information in a block, using the three bit-wise operations NOT, AND, and OR.

1- The number of addresses in the block N = NOT (mask) + 1.

2- The first address in the block = (Any address in the block) AND (mask).

3- The last address in the block = (Any address in the block) OR [(NOT (mask)].

Example: We repeat previous example using the mask. The mask in dotted-decimal notation is 255.255.255.224. The AND, OR, and NOT operations can be applied to individual bytes using calculators.

1- Number of addresses in the block: N = NOT (mask) + 1= 0.0.0.31 + 1 = 32 addresses

2- First address: First = (address) AND (mask) = 167.199.170.64

3- Last address: Last = (address) OR (NOT mask) = 167.199.170.95

Example: The address 230.8.24.56 can belong to many blocks. Some of them are shown below with the value of the prefix associated with that block:

Prefix length: 16	→	Block:	230.8.0.0	to	230.8.255.255
Prefix length: 20	→	Block:	230.8.16.0	to	230.8.31.255
Prefix length: 26	→	Block:	230.8.24.0	to	230.8.24.63
Prefix length: 27	→	Block:	230.8.24.32	to	230.8.24.63
Prefix length: 29	→	Block:	230.8.24.56	to	230.8.24.63
Prefix length: 31	→	Block:	230.8.24.56	to	230.8.24.57

Network and Broadcast Addresses

On each IP network, two host addresses are reserved for special use:

- The network (or subnet) address
- The broadcast address

Neither of these addresses can be assigned to an actual host.

The network address

- The network address is used to identify the network itself.
- A routing table contains a list of known networks, and each network is identified by its network address.
- Network addresses contain all 0 bits in the host portion of the address.

Example: $192.168.1.0/24$ is a network address. This can be determined by looking at the address and subnet mask in binary:

IP Address: $11000000.10101000.00000001.00000000$

Subnet Mask: $11111111.11111111.11111111.00000000$

Note that all host bits in the address are set to 0.

The broadcast address

The broadcast address identifies all hosts on a particular network.

- A packet sent to the broadcast address will be received and processed by every host on that network.
- Broadcast addresses contain all 1 bits in the host portion of the address.

Example: $192.168.1.255/24$ is a broadcast address:

IP Address: $11000000.10101000.00000001.11111111$

Subnet Mask: $11111111.11111111.11111111.00000000$

Note that all host in the address bits are set to 1

Example: For 192.168.0.5/24 IP addresses, Find:

1- Network address → 192.168.0.0

2- First host IP address → 192.168.0.1

3- Last host IP address → 192.168.0.254

4- Broadcast address → 192.168.0.255

(H.W) Example: For following IP addresses:

10.223.220.15/8, 10.223.220.15/16, 192.15.32.16/20, 80.90.161.50/11 Find:

1- Network address 2- First host IP address

3- Last host IP address 4- Broadcast address

Address Classes vs. Subnet Mask

Remember the following three rules:

- The first octet on an address dictates the class of that address.
- The subnet mask determines what part of an address identifies the network, and what part identifies the host.
- Each class has a default subnet mask. A network using its default subnet mask is referred to as a classful network.

For example, 10.1.1.1 is a Class A address, and its default subnet mask is 255.0.0.0 (/8 in CIDR). It is entirely possible to use subnet masks other than the default. For example, a Class B subnet mask can be applied to a Class A address: 10.1.1.1 /16

However, this does not change the class of the above address. It remains a Class A address, which has been subnetted using a Class B mask.

- **Remember**, the only thing that determines the class of an IP address is the first octet of that address.
- Likewise, the subnet mask is the only thing that determines what part of an address identifies the network, and what part identifies the host.

- Make a note to self that the /8 through /15 can only be used with Class A network addresses. /16 through /23 can be used by Class A and B network addresses. /24 through /30 can be used by Class A, B, and C network addresses. This is a big reason why most companies use Class A network addresses. By being allowed the use all subnet masks, they gain the valuable benefit of maximum flexibility for their network design.

Private vs. Public IPv4 Addresses

The rapid growth of the Internet resulted in a shortage of available IPv4 addresses. In response, a specific subset of the IPv4 address space was designated as private, to temporarily alleviate this problem.

A public address can be routed on the Internet. Thus, hosts that must be Internet-accessible must be configured with (or reachable by) public addresses. Allocation of public addresses is governed by the Internet Assigned Numbers Authority (IANA).

A private address is intended for internal use within a home or organization, and can be freely used by anyone. However, private addresses can never be routed on the Internet. In fact, Internet routers are configured to immediately drop traffic with private addresses.

Three private address ranges were defined in RFC 1918, one for each IPv4 class:

- Class A – 10.x.x.x /8
- Class B – 172.16.x.x /12
- Class C – 192.168.x.x /24

- It is possible to translate between private and public addresses, using Network Address Translation (NAT). NAT allows a host configured with a private address to be stamped with a public address, thus allowing that host to communicate across the Internet.
- It is also possible to translate multiple privately-addressed hosts to a single public address, which conserves the public address space.
- NAT provides an additional benefit – hiding the specific addresses and addressing structure of the internal (or private) network.

Note: NAT is not restricted to private-to-public address translation, though that is the most common application. NAT can also perform public-to-public address translation, as well as private-to-private address translation.

NAT is only a temporarily solution to the address shortage problem. IPv4 will eventually be replaced with IPv6, which supports a vast address space.

Block Allocation

The next issue in classless addressing is block allocation. How are the blocks allocated?

The ultimate responsibility of block allocation is given to a global authority called the Internet Corporation for Assigned Names and Numbers (ICANN). However, ICANN does not normally allocate addresses to individual Internet users. It assigns a large block of addresses to an ISP (or a larger organization that is considered an ISP in this case). For the proper operation of the CIDR, a restriction need to be applied to the allocated block.

- The number of requested addresses, N, needs to be a power of 2.

The reason is that:

$$N = 2^{32-n} \text{ or } n = 32 - \log_2 N.$$

If N is not a power of 2, we cannot have an integer value for n.

Example: An ISP has requested a block of 1000 addresses. Since 1000 is not a power of 2, 1024 addresses are granted.

The prefix length is calculated as $n = 32 - \log_2 1024 = 22$. An available block, 18.14.12.0/22, is granted to the ISP.

SUBNETTING

Subnetting is the process of creating new networks (or subnets) by stealing bits from the host portion of a subnet mask. There is one caveat: stealing bits from hosts creates more networks but fewer hosts per network.

Consider the following Class C network: $192.168.254.0$

The default subnet mask for this network is $255.255.255.0$.

This single network can be segmented, or subnetted, into multiple networks.

For example, assume a minimum of 10 new networks are required.

Resolving this is possible using the following magical formula: $\mathbf{2^b}$

The exponent (b) identifies the number of bits to steal from the host portion of the subnet mask.

The default Class C mask $(255.255.255.0)$ looks as follows in binary:

$$11111111.1111111.1111111.00000000$$

- There are a total of 24 bits set to 1, which are used to identify the network.
- There are a total of 8 bits set to 0, which are used to identify the host, and these host bits can be stolen.

Stealing bits essentially involves changing host bits (set to 0 or off) in the subnet mask to network bits (set to 1 or on).

Remember, network bits in a subnet mask must always be contiguous (skipping bits is not allowed).

– Consider the result if three bits are stolen. Using the above formula:

$$2^b = 2^3 = 8 \rightarrow 8 \text{ new networks created}$$

However, a total of 8 new networks does not meet the original requirement of at least 10 networks.

– Consider the result if four bits are stolen:

$$2^b = 2^4 = 16 \rightarrow 16 \text{ new networks created}$$

A total of 16 new networks does meet the original requirement. Stealing four host bits results in the following new subnet mask:

$$11111111.11111111.11111111.11110000$$

$$255.255.255.\underline{240}$$

In the previous example, a Class C network was subnetted to create 16 new networks, using a subnet mask of $255.255.255.240$ (or $/28$ in CIDR). Four bits were stolen in the subnet mask, leaving only four bits for hosts.

- To determine the number of hosts (usable hosts) for each of the new 16 networks, a slightly modified formula is required: $\mathbf{2^{32-n} - 2}$
- $32 - \mathrm{n}$ = number of bits available for hosts = $32 - 28 = 4$

$$\mathbf{2^4 - 2 = 16 - 2 = 14\ usable\ hosts\ per\ network}$$

Thus, subnetting a Class C network with a $/28$ mask creates 16 new networks, with 14 usable hosts per network.

Why is the formula for calculating usable hosts $2^{32-n} - 2$?

Because it is never possible to assign the bits which belongs to host portion in any block of addresses with all 0 bits (first address) or all 1 (last address). These are reserved for the network (subnet) and broadcast addresses.

Determining the Range of Subnetted Networks

Determining the range of the newly created networks can be accomplished using two methods:

1- The long method

- Consider the example $192.168.254.0/24$ network again, which was subnetted using a $255.255.255.240$ mask:

$192.168.254.0 \rightarrow 11000000.10101000.11111110.00000000$

$255.255.255.240 \rightarrow 11111111.11111111.11111111.11110000$

Subnetting stole four bits in the fourth octet, creating a total of 16 new networks. Looking at only the fourth octet, the first newly created network is 0000. The second new network is 0001. Calculating all possible permutations of the four stolen bits:

Binary	Network Address	First host	Last host	broadcast Address
.0000 xxxx	.0	.1	.14	.15
.0001 xxxx	.16	.17	.30	.31
.0010 xxxx	.32	.33	.46	.47
.0011 xxxx	.48	.49	.62	.63
.0100 xxxx	.64	.65	.78	.79
.0101 xxxx	.80	.81	.94	.95
.0110 xxxx	.96	.97	.110	.111
.0111 xxxx	.112	.113	.126	.127
.1000 xxxx	.128	.129	.142	.143
.1001 xxxx	.144	.145	.158	.159
.1010 xxxx	.160	.161	.174	.175
.1011 xxxx	.176	.177	.190	.191
.1100 xxxx	.192	.193	.206	.207
.1101 xxxx	.208	.209	.222	.223
.1110 xxxx	.224	.225	.238	.239
.1111 xxxx	.240	.241	.254	.255

2- Fast method

- Calculating the ranges of subnetted networks can **quickly** become tedious when using the long binary method.
- The shortcut method involves taking the subnet mask (255.255.255.240 from the previous example), and subtracting the subnetted octet (240) from 256.

256 – 240 = 16

- The first network will begin at 0. Then, simply continue adding 16 to identify the first address of each new network:

0 16 32 48 64 80 96 112 128 144 160 176 192 208 224 240

- Knowing the first address of each new network makes it simple to determine the last address of each network:

First address of network:	0	16	32	48	64	80	96	112	128	144
Last address of network:	15	31	47	63	79	95	111	127	143	159

- Only the first 10 networks were calculated, for brevity.
- The first address of each network becomes the subnet address for that network.
- The last address of each network becomes the broadcast address for that network.
- Once the first and last address of each network is known, determining the usable range for hosts is straightforward:

Network Address	0	16	32	48	64	80	96	112	128	144
	1	17	33	49	65	81	97	113	129	145
Usable Addresses	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
	14	30	46	46	78	94	110	126	142	158
Broadcast Address	15	31	47	47	79	95	111	127	143	159

- Hosts on the same network (such as $192.168.254.2$ and $192.168.254.14$) can communicate freely.
- Hosts on different networks (such as $192.168.254.61$ and $192.168.254.66$) require a router to communicate

Example: Consider the following subnetted Class A network: $10.0.0.0/8$ which was subnetted using a $255.255.248.0$ mask. Now consider the following questions:

1- How many new networks were created?

2- How many usable hosts are there per network?

3- What is the full range of the first three networks?

Solution:

1- By default, the $10.0.0.0$ network has a subnet mask of $255.0.0.0$.

To determine the number of bits stolen:

$255.0.0.0 \rightarrow 11111111.\underline{00000000.00000}000.00000000$

$255.255.248.0 \rightarrow 11111111.\underline{11111111.11111}000.00000000$

Clearly, **13 bits** have been stolen to create the new subnet mask. To calculate the total number of new networks:

$$\mathbf{2^b = 2^{13} = 8192 \text{ new network created}}$$

2- There are clearly 11 bits remaining in the host portion of the mask:

$$\mathbf{2^{32-n} - 2 = 2^{11} - 2 = 2048 - 2 = 2046 \text{ usable hosts per network}}$$

3- Calculating the ranges is a bit tricky. Using the shortcut method, subtract the third octet (248) of the subnet mask $(255.255.248.0)$ from 256.

$$256 - 248 = 8$$

The first network will begin at 0, again. However, the ranges are spread across multiple octets.

The ranges of the first three networks look as follows:

Network Address	10.0.0.0	10.0.8.0	10.0.16.0	10.0.24.0
Usable Addresses	10.0.0.1 ↕	10.0.8.1 ↕	10.0.16.1 ↕	10.0.24.1 ↕
	10.0.7.254	10.0.15.254	10.0.23.254	10.0.31.254
Broadcast Address	10.0.7.255	10.0.15.255	10.0.23.255	10.0.31.255

Example: If you have these information:

192.168.10.33 = Host address

255.255.255.224 = Subnet mask

Determine the subnet and broadcast address of this IP address?

Solution:

- Block size or increment number = 256 − 224 = 32.
- Start counting at zero in blocks of 32 until you reach the subnet mask value, and these are your subnets. 0, 32, 64, 96,128, 160, 192,224.
- The address of 33 falls between the two subnets of 32 and 64 and must be part of the 192.168.10.32 subnet.
- The next subnet is 64, so the broadcast address of the 32 subnet is 63. (Remember that the broadcast address of a subnet is always the number right before the next subnet.)
- The valid host range is 33–62 (the numbers between the subnet and broadcast address).

H.W1: An organization is granted the block 130.56.0.0/16. The administrator wants to create 1024 subnets.

1- Find the subnet mask.

2- Find the number of addresses in each subnet.

3- Find the first and last addresses in subnet 1.

H.W2: In a block of addresses, we know the IP address of one host is 182.44.82.16/26. What are the first address (network address) and the last address in this block?

H.W3: Find the range of addresses in the following blocks:

1- 123.56.77.32/29

2- 200.17.21.128/27

VLSM (Variable Length Subnet Mask)

Internet Service Providers may face a situation where they need to allocate IP subnets of different sizes as per the requirement of customer. One customer may ask Class C subnet of 3 IP addresses and another may ask for 10 IPs. For an ISP, it is not feasible to divide the IP addresses into fixed size subnets, rather he may want to subnet the subnets in such a way which results in minimum wastage of IP addresses.

For example, an administrator has $192.168.1.0/24$ network. the administrator has four different departments with different number of hosts. Sales department has 100 computers, Purchase department has 50 computers, Accounts has 25 computers and Management has 5 computers.

In CIDR, the subnets are of fixed size. Using the same methodology, the administrator cannot fulfill all the requirements of the network.

Designing subnets

The subnetworks in a network should be carefully designed to enable the routing of packets.

Assume: The total number of addresses granted to the organization is N

The prefix length is n

The assigned number of addresses to each subnetwork is N_{sub}

The prefix length for each subnetwork is n_{sub}

Then the following steps need to be carefully followed to guarantee the proper operation of the subnetworks.

- The number of addresses in each subnetwork should be a power of 2.
- The prefix length for each subnetwork should be found using the following formula:

$$n_{sub} = 32 - \log_2 N_{sub}$$

Example: An organization is granted a block of addresses with the beginning address $14.24.74.0/24$. The organization needs to have 3 subblocks of addresses to use in its three subnets: one subblock of 10 addresses, one subblock of 60 addresses, and one subblock of 120 addresses. Design the subblocks.

Solution:

There are $2^{32-24} = 256$ addresses in this block:

- The first address is $14.24.74.0/24$
- The last address is $14.24.74.255/24$.

To satisfy the third requirement, we assign addresses to subblocks, starting with the largest and ending with the smallest one.

1- The number of addresses in the largest subblock, which requires 120 addresses, is not a power of 2. We allocate 128 addresses.

The sub net mask for this subnet can be found as $n_1 = 32 - 10g_2128 = 25$.

The first address in this block is $14.24.74.0/25$; the last address is $14.24.74.127/25$.

2- The number of addresses in the second largest subblock, which requires 60 addresses, is not a power of 2 either. We allocate 64 addresses.

The subnet mask for this subnet can be found as $n_2 = 32 - \log_2 64 = 26$.

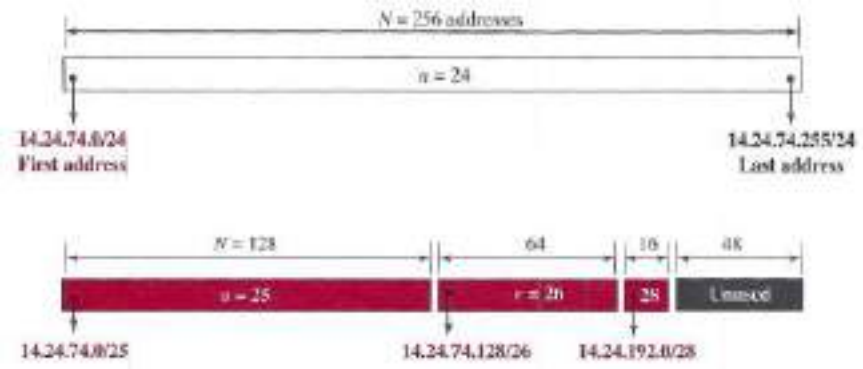
The first address in this block is $14.24.74.128/26$; the last address is $14.24.74.191/26$.

3- The number of addresses in the smallest subblock, which requires 10 addresses, is not a power of 2 either. We allocate 16 addresses.

The subnet mask for this subnet can be found as $n_3 = 32 - \log_2 16 = 28$.

The first address in this block is $14.24.74.192/28$; the last address is $14.24.74.207/28$.

If we add all addresses in the previous subblocks, the result is 208 addresses, which means 48 addresses are left in reserve. The first address in this range is $14.24.74.208$. The last address is $14.24.74.255$. We don't know about the prefix length yet. The figure below shows the configuration of blocks. We have shown the first address in each block.



Special Addresses

Before finishing the topic of addresses in IPv4, we need to mention five special addresses that are used for special purposes: this-host address, limited-broadcast address, loopback address, and multicast addresses.

1- This-host Address

The only address in the block $0.0.0.0/32$ is called the this-host address. It is used whenever a host needs to send an IP datagram but it does not know its own address to use as the source address.

2- Limited-broadcast Address

The only address in the block $255.255.255.255/32$ is called the limited-broadcast address. It is used whenever a router or a host needs to send a datagram to all devices in a network. The routers in the network, however, block the packet having this address as the destination; the packet cannot travel outside the network.

3- Loopback Address

The block $127.0.0.0/8$ is called the loopback address. A packet with one of the addresses in this block as the destination address never leaves the host; it will remain in the host. Any address in the block is used to test a piece of software in the machine.

4- Multicast Addresses

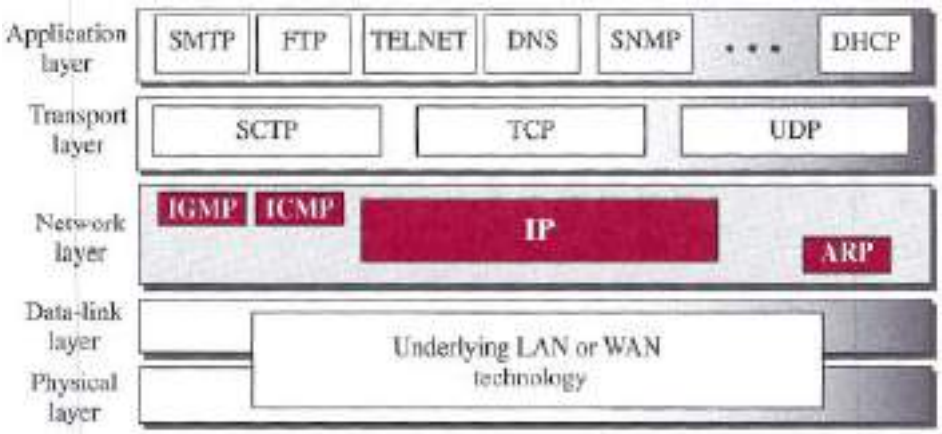
The block $224.0.0.0/4$ is reserved for multicast addresses.

INTERNET PROTOCOL (IP)

The network layer in version 4 can be thought of as one main protocol and three auxiliary ones.

- The main protocol, **Internet Protocol version 4 (IPv4),** is responsible for packetizing, forwarding, and delivery of a packet at the network layer.
- **The Internet Control Message Protocol version 4 (ICMPv4)** helps IPv4 to handle some errors that may occur in the network-layer delivery.
- **The Internet Group Management Protocol (IGMP)** is used to help IPv4 in multicasting.
- **The Address Resolution Protocol (ARP)** is used to glue the network and data-link layers in mapping network-layer addresses to link-layer addresses.

The figure below shows the positions of these four protocols in the TCP/IP protocol suite.



We will now discuss IPv4 and later ICMPv4

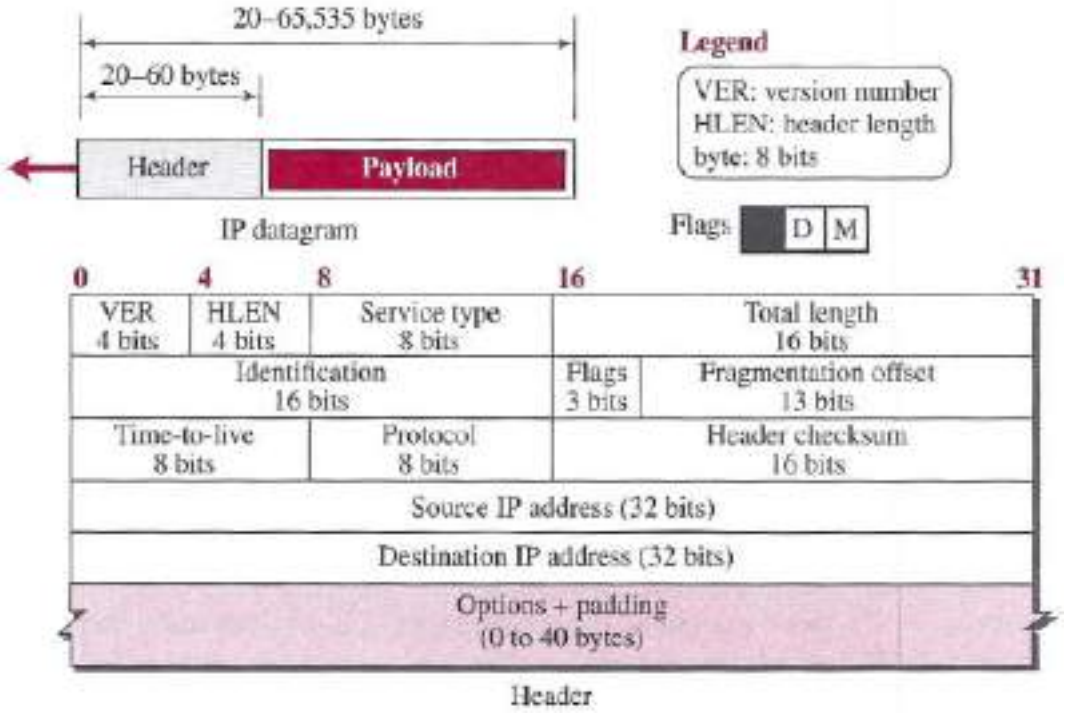
IPv4 is an unreliable datagram protocol, a best-effort delivery service.

The term **best-effort** means that IPv4 packets can be corrupted, be lost, arrive out of order, or be delayed, and may create congestion for the network. If reliability is important, IPv4 must be paired with a reliable transport-layer protocol such as TCP. An example of a more commonly understood best-effort delivery service is the post office. The post office does its best to deliver the regular mail but does not always succeed. If an unregistered letter is lost or damaged, it is up to the sender or would be recipient to discover this. The post office itself does not keep track of every letter and cannot notify a sender of loss or damage of one.

IPv4 is also a connectionless protocol that uses the datagram approach. This means that each datagram is handled independently, and each datagram can follow a different route to the destination. This implies that datagrams sent by the same source to the same destination could arrive out of order. Again, IPv4 relies on a higher level protocol to take care of all these problems.

Datagram Format

We begin by discussing the first service provided by IPv4, packetizing. We show how IPv4 defines the format of a packet in which the data coming from the upper layer or other protocols are encapsulated. Packets used by the IP are called datagrams. The figure below shows the IPv4 datagram format. A datagram is a variable-length packet consisting of two parts: header and payload (data). The header is 20 to 60 bytes in length and contains essential information to routing and delivery.



Discussing the meaning and rationale for the existence of each field is essential to understanding the operation of IPv4; a brief description of each field is in order.

- **Version Number**: The 4-bit version number (VER) field defines the version of the IPv4 protocol, which, obviously, has the value of 4.

- **Header Length**: The 4-bit header length (HLEN) field defines the total length of the datagram header in 4-byte words. The IPv4 datagram has a variable-length header. When a device receives a datagram, it needs to know when the header stops and the data, which is encapsulated in the packet, starts. However, to make the value of the header length (number of bytes) fit in a 4-bit header length, the total length of the header is calculated as 4-byte words. The total length is divided by 4 and the value is inserted in the field. The receiver needs to multiply the value of this field by 4 to find the total length.

- **The Differentiated services (Type of service):** It is one of the few fields that has changed its meaning (slightly) over the years. Originally, it was called the Type of service field. it defines the quality of service.

  Bits $0-2$: Precedence.

  Bit 3: 0 = Normal Delay, 1 = Low Delay.

  Bit 4: 0 = Normal Throughput, 1 = High Throughput.

  Bit 5: 0 = Normal Reliability, 1 = High Reliability.

  Bit $6-7$: Reserved for Future Use.

- **Total Length:** This 16-bit field defines the total length (header plus data) of the IP datagram in bytes. A 16-bit number can define a total length of up to $65,535$ (when all bits are 1s). However, the size of the datagram is normally much less than this. This field helps the receiving device to know when the packet has completely arrived. To find the length of the data coming from the upper layer, subtract the header length from the total length. The header length can be found by multiplying the value in the HLEN field by 4.

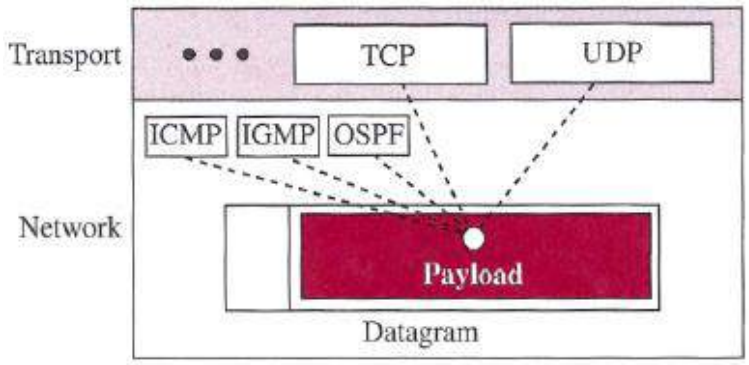
  Length of data = total length − (HLEN) X 4

  One may ask why we need this field anyway. When a machine (router or host) receives a frame, it drops the header and the trailer, leaving the datagram. Why include an extra field that is not needed? The answer is that in many cases we really do not need the value in this field. However, there are occasions in which the datagram is not the only thing encapsulated in a frame; it may be that padding has been added. For example, the Ethernet protocol has a minimum and maximum restriction on the size of data that can be encapsulated in a frame (46 to 1500 bytes). If the size of an IPv4 datagram is less than 46 bytes, some padding will be added to meet the requirement. In this case, when a machine decapsulates the datagram, it needs to check the total length field to determine how much is really data and how much is padding.

- **Identification, Flags, and Fragmentation Offset:** These three fields are related to the fragmentation of the IP datagram when the size of the datagram is larger than the underlying network can carry. We discuss the contents and importance of these fields when we talk about fragmentation in the next section.

- **Time-to-live (TTL):** Due to some malfunctioning of routing protocols, a datagram may be circulating in the Internet, visiting some networks over and over without reaching the destination. This may create extra traffic in the Internet. The time-to-live (TTL) field is used to control the maximum number of hops (routers) visited by the datagram. When a source host sends the datagram, it stores a number in this field. This value is approximately two times the maximum number of routers between any two hosts. Each router that processes the datagram decrements this number by one. If this value, after being decremented, is zero, the router discards the datagram.

- **Protocols:** In TCP/IP, the data section of a packet, called the payload, carries the whole packet from another protocol. A datagram, for example, can carry a packet belonging to any transport-layer protocol such as UDP or TCP. A datagram can also carry a packet from other protocols that directly use the service of the IP, such as some routing protocols or some auxiliary protocols. The Internet authority has given any protocol that uses the service of IP a unique 8-bit number which is inserted in the protocol field. When the payload is encapsulated in a datagram at the source IP, the corresponding protocol number is inserted in this field; when the datagram arrives at the destination, the value of this field helps to define to which protocol the payload should be delivered. In other words, this field provides multiplexing at the source and de-multiplexing at the destination, as shown in the figure below. Note that the protocol fields at the network layer play the same role as the port numbers at the transport layer. However, we need two port numbers in a transport-layer packet because the port numbers at the source and destination are different, but we need only one protocol field because this value is the same for each protocol no matter whether it is located at the source or the destination.



Some protocol values

ICMP	01
IGMP	02
TCP	06
UDP	17
OSPF	89

- **Header checksum:** IP is not a reliable protocol; it does not check whether the payload carried by a datagram is corrupted during the transmission. IP puts the burden of error checking of the payload on the protocol that owns the payload, such as UDP or TCP. The datagram header, however, is added by IP, and its error-checking is the responsibility of IP. Errors in the IP header can be a disaster. For example, if the destination IP address is corrupted, the packet can be delivered to the wrong host. If the protocol field is corrupted, the payload may be delivered to the wrong protocol. If the fields related to the fragmentation are corrupted, the datagram cannot be reassembled correctly at the destination, and so on. For these reasons, IP adds a header checksum field to check the header, but not the payload. We need to remember that, since the value of some fields, such as TTL, which are related to fragmentation and options, may change from router to router, the checksum needs to be recalculated at each router.

- **Source and Destination Addresses:** These 32-bit source and destination address fields define the IP address of the source and destination respectively. The source host should know its IP address. The destination IP address is either known by the protocol that uses the service of IP or is provided by the DNS as described later. Note that the value of these fields must remain unchanged during the time the IP datagram travels from the source host to the destination host.

- **Options:** A datagram header can have up to 40 bytes of options. Options can be used for network testing and debugging. The options field was created in order to allow features to be added into IP as time passes and requirements change. The existence of options in a header creates some burden on the datagram handling; some options can be changed by routers, which forces each router to recalculate the header checksum.

- **Payload:** Payload, or data, is the main reason for creating a datagram. Payload is the packet coming from other protocols that use the service of IP. Comparing a datagram to a postal package, payload is the content of the package; the header is only the information written on the package.

Example: An IPv4 packet has arrived with the first 8 bits as $(01000010)2$. The receiver discards the packet. Why?

Solution: There is an error in this packet.

The 4 leftmost bits $(0100)2$ show the version, which is correct.

The next 4 bits $(0010)2$ show an invalid header length $(2 \times 4 = 8)$. The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

Example: In an IPv4 packet, the value of HLEN is (1000). How many bytes of options are being carried by this packet?

Solution: The HLEN value is 8, which means the total number of bytes in the header is 8 x 4, or 32 bytes. The first 20 bytes are the base header, the next 12 bytes are the options.

Example: In an IPv4 packet, the value of HLEN is 5, and the value of the total length field is $(0028)_{16}$. How many bytes of data are being carried by this packet?

Solution: The HLEN value is 5, which means the total number of bytes in the header is 5 x 4, or 20 bytes (no options). The total length is $(0028)16$ or 40 bytes, which means the packet is carrying 20 bytes of data $(40 - 20)$.

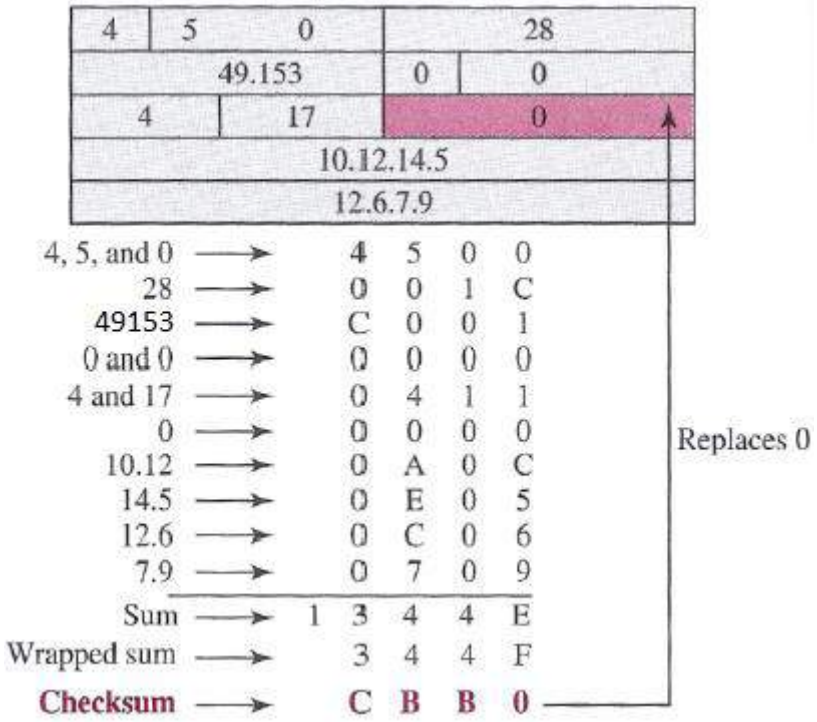
Example: An IPv4 packet has arrived with the first few hexadecimal digits as shown.

$$(45000028000100000102 \ldots)_{16}$$

How many hops can this packet travel before being dropped? The data belong to what upper-layer protocol?

Solution: To find the time-to-live field, we skip 8 bytes (16 hexadecimal digits). The time-to-live field is the ninth byte, which is $(01)_{16}$. This means the packet can travel only one hop. The protocol field is the next byte $(02)16$, which means that the upper-layer protocol is IGMP.

Example: The figure below shows an example of a checksum calculation for an IPv4 header without options. The header is divided into 16-bit sections. All the sections are added and the sum is complemented after wrapping the leftmost digit. The result is inserted in the checksum field.



Note that the calculation of wrapped sum and checksum can also be done as follows in hexadecimal:

Wrapped Sum = Sum mod FFFF

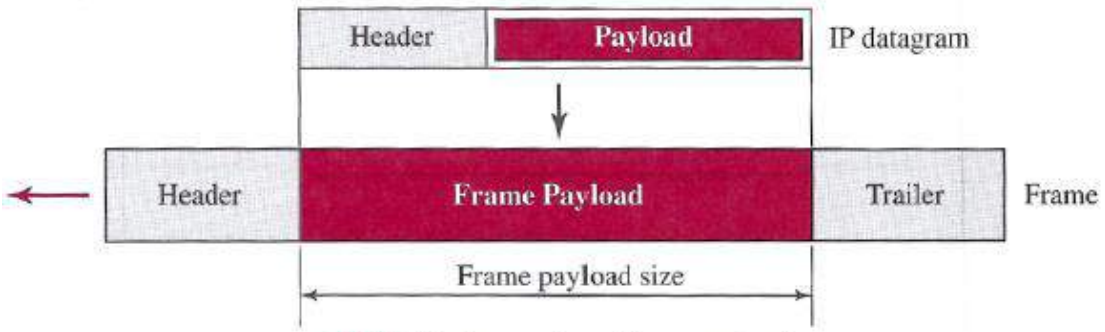
Checksum = FFFF − Wrapped Sum

FRAGMENTATION

- A datagram can travel through different networks.
- Each router decapsulates the IP datagram from the frame it receives, processes it, and then encapsulates it in another frame.
- The format and size of the received frame depend on the protocol used by the physical network through which the frame has just traveled.
- The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel.
- For example, if a router connects a LAN to a WAN, it receives a frame in the LAN format and sends a frame in the WAN format.

Maximum Transfer Unit (MTU)

- Each link-layer protocol has its own frame format.
- One of the features of each format is the maximum size of the payload that can be encapsulated.
- In other words, when a datagram is encapsulated in a frame, the total size of the datagram must be less than this maximum size, which is defined by the restrictions imposed by the hardware and software used in the network (see the figure below).



MTU: Maximum size of frame payload

- The value of the MTU differs from one physical network protocol to another. For example, the value for a LAN is normally 1500 bytes, but for a WAN it can be larger or smaller.
- In order to make the IP protocol independent of the physical network, the designers decided to make the maximum length of the IP datagram equal to 65,535 bytes.
- This makes transmission more efficient if one day we use a link-layer protocol with an MTU of this size.

- However, for other physical networks, we must divide the datagram to make it possible for it to pass through these networks. This is called fragmentation.
- When a datagram is fragmented, each fragment has its own header with most of the fields repeated, but some have been changed.
- A fragmented datagram may itself be fragmented if it encounters a network with an even smaller MTU.
- In other words, a datagram may be fragmented several times before it reaches the final destination.
- A datagram can be fragmented by the source host or any router in the path.
- The reassembly of the datagram, however, is done only by the destination host, because each fragment becomes an independent datagram. Whereas the fragmented datagram can travel through different routes, and we can never control or guarantee which route a fragmented datagram may take, all of the fragments belonging to the same datagram should finally arrive at the destination host. So it is logical to do the reassembly at the final destination.
- When we talk about fragmentation, we mean that the payload of the IP datagram is fragmented.
- However, most parts of the header, with the exception of some options, must be copied by all fragments.
- The host or router that fragments a datagram must change the values of three fields: flags, fragmentation offset, and total length.
- The rest of the fields must be copied. Of course, the value of the checksum must be recalculated regardless of fragmentation.

Fields Related to Fragmentation

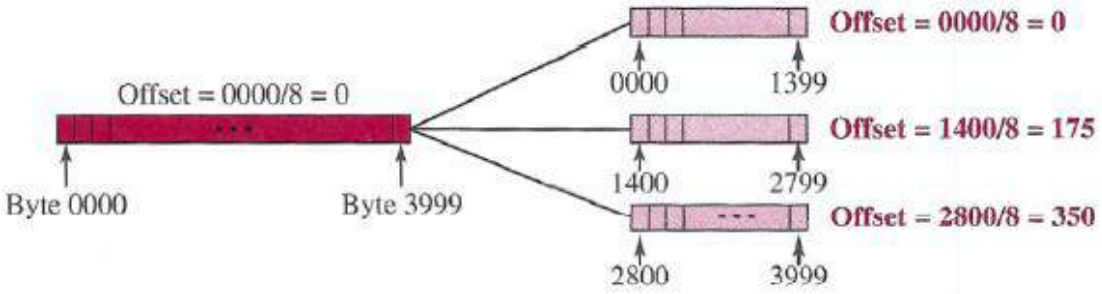
We mentioned before that three fields in an IP datagram are related to fragmentation: identification, flags, and fragmentation offset. Let us explain these fields now.

- **The 16-bit identification field** identifies a datagram originating from the source host. The combination of the identification and source IP address must uniquely define a datagram as it leaves the source host. When a datagram is fragmented, the value in the identification field is copied into all fragments. In other words, all fragments have the same identification number, which is also the same as the original datagram. The identification number helps the destination in reassembling the

datagram. It knows that all fragments having the same identification value should be assembled into one datagram.

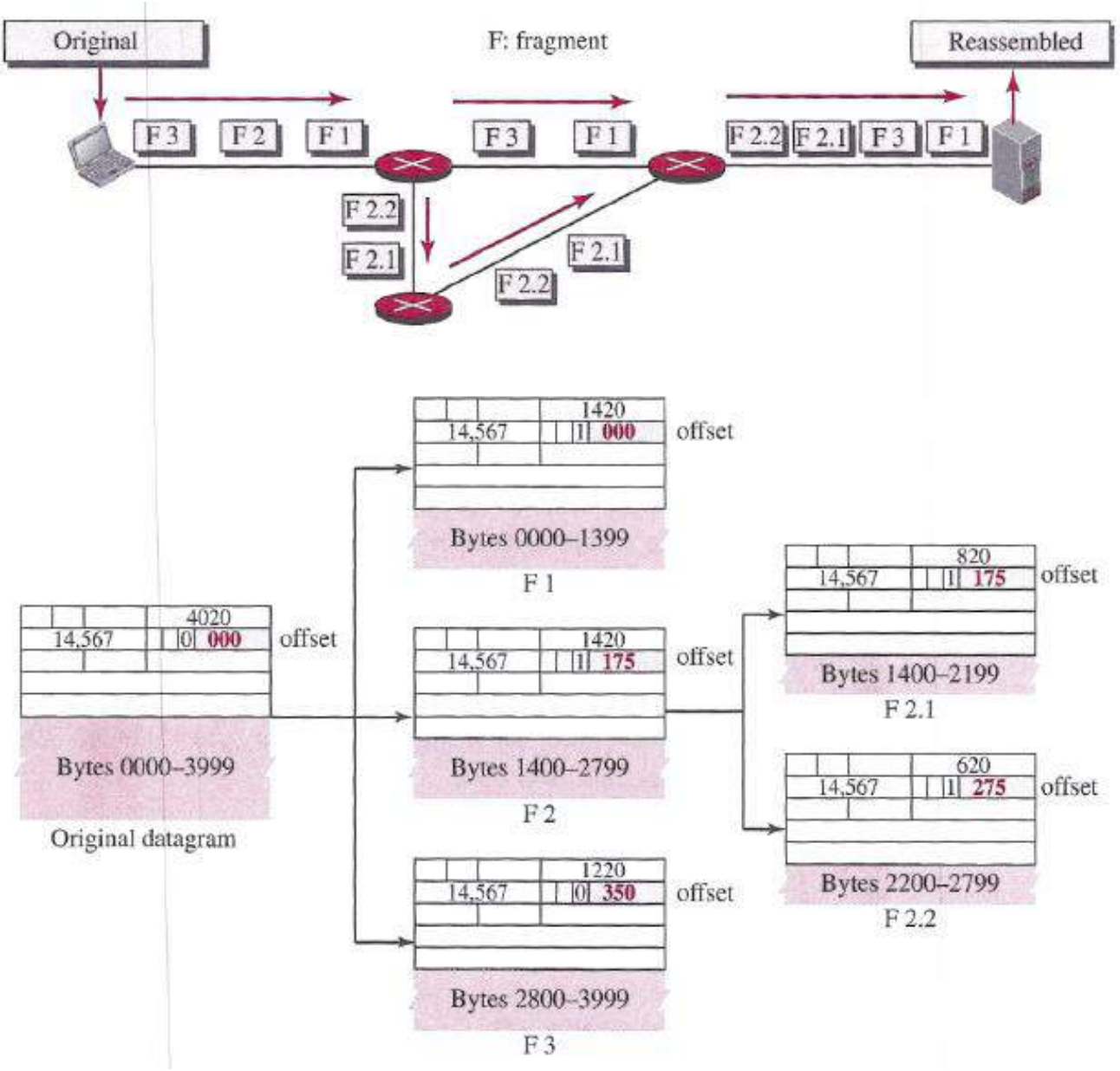
- The 3-bit flags field defines three flags.
  - **The leftmost bit** is reserved (not used).
  - **The second bit (D bit)** is called the do not fragment bit. If its value is 1, the machine must not fragment the datagram. If it cannot pass the datagram through any available physical network, it discards the datagram and sends an ICMP error message to the source host. If its value is 0, the datagram can be fragmented if necessary.
  - **The third bit (M bit)** is called the more fragment bit. If its value is I, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment.

- **The 13-bit fragmentation offset field** shows the relative position of this fragment with respect to the whole datagram. It is the offset of the data in the original datagram measured in units of 8 bytes. The figure below shows a datagram with a data size of 4000 bytes fragmented into three fragments. The bytes in the original datagram are numbered 0 to 3999. The first fragment carries bytes 0 to 1399. The offset for this datagram is 0/8 = 0. The second fragment carries bytes 1400 to 2799; the offset value for this fragment is 1400/8 = 175. Finally, the third fragment carries bytes 2800 to 3999. The offset value for this fragment is 2800/8 = 350.



Remember that the value of the offset is measured in units of 8 bytes. This is done because the length of the offset field is only 13 bits long and cannot represent a sequence of bytes greater than 8191. This forces hosts or routers that fragment datagrams to choose the size of each fragment so that the first byte number is divisible by 8.

The figure below shows an expanded view of the fragments in the previous figure. The original packet starts at the client; the fragments are reassembled at the server. The value of the identification field is the same in all fragments, as is the value of the flags field with the more bit set for all fragments except the last. Also, the value of the offset field for each fragment is shown. Note that although the fragments arrived out of order at the destination, they can be correctly reassembled.



The figure also shows what happens if a fragment itself is fragmented. In this case the value of the offset field is always relative to the original datagram. For example, in the figure1 the second fragment

is itself fragmented later into two fragments of 800 bytes and 600 bytes, but the offset shows the relative position of the fragments to the original data.

It is obvious that even if each fragment follows a different path and arrives out of order, the final destination host can reassemble the original datagram from the fragments received (if none of them is lost) using the following strategy:

1- The first fragment has an offset field value of zero.

2- Divide the length of the first fragment by 8. The second fragment has an offset value equal to that result.

3- Divide the total length of the first and second fragment by 8. The third fragment has an offset value equal to that result.

4- Continue the process. The last fragment has its M bit set to 0.

5- Continue the process. The last fragment has a more bit value of 0.

Example: A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution: If the M bit is 0, it means that there are no more fragments; the fragment is the last one. However, we cannot say if the original packet was fragmented or not. A non-fragmented packet is considered the last fragment.

Example: A packet has arrived with an M bit value of 1. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution: If the M bit is 1, it means that there is at least one more fragment. This fragment can be the first one or a middle one, but not the last one. We don't know if it is the first one or a middle one; we need more information (the value of the fragmentation offset).

Example: A packet has arrived with an M bit value of 1 and a fragmentation offset value of 0. Is this the first fragment, the last fragment, or a middle fragment?

Solution: Because the M bit is I, it is either the first fragment or a middle one. Because the offset value is 0, it is the first fragment.

Example: A packet has arrived in which the offset value is 100. What is the number of the first byte? Do we know the number of the last byte?

Solution: To find the number of the first byte, we multiply the offset value by 8. This means that the first byte number is 800. We cannot determine the number of the last byte unless we know the length of the data.

Example: A packet has arrived in which the offset value is 100, the value of HLEN is 5, and the value of the total length field is 100. What are the numbers of the first byte and the last byte?

Solution: The first byte number is 100 x 8 = 800. The total length is 100 bytes, and the header length is 20 bytes (5 x 4), which means that there are 80 bytes in this datagram. If the first byte number is 800, the last byte number must be 879.

The Internet Control Message Protocol version 4 (ICMPv4)

The IPv4 has no error-reporting or error-correcting mechanism. What happens if something goes wrong? What happens if a router must discard a datagram because it cannot find a route to the final destination, or because the time-to-live field has a zero value? What happens if the final destination host must discard the received fragments of a datagram because it has not received all fragments within a predetermined time limit?

These are examples of situations where an error has occurred and the IP protocol has no built-in mechanism to notify the original host.

The IP protocol also lacks a mechanism for host and management queries. A host sometimes needs to determine if a router or another host is alive. Sometimes a network manager needs information from another host or router.

The Internet Control Message Protocol version 4 (ICMPv4) has been designed to compensate for the above two deficiencies. It is a companion to the IP protocol. ICMP itself is a network-layer protocol. However, its messages are not passed directly to the data-link layer as would be expected. Instead, the messages are first encapsulated inside IP datagrams before going to the lower layer. When an IP datagram encapsulates an ICMP message, the value of the protocol field in the IP datagram is set to 1 to indicate that the IP payroll is an ICMP message.

Messages

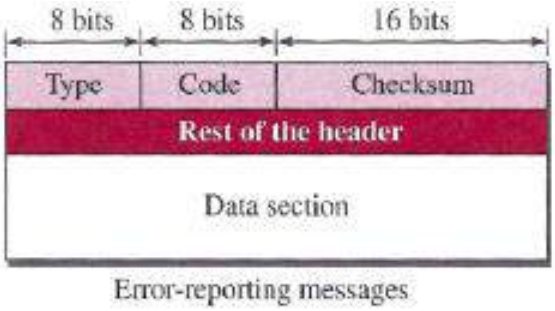
ICMP messages are divided into two broad categories: error-reporting messages and query messages.

- The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet.
- The query messages help a host or a network manager get specific information from a router or another host. For example, nodes can discover their neighbors. Also, hosts can discover and learn about routers on their network and routers can help a node redirect its messages.

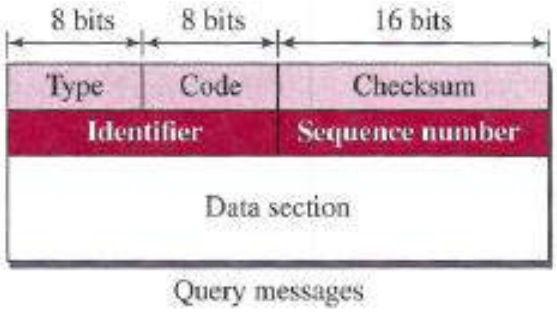
An ICMP message has an 8-byte header and a variable-size data section. Although the general format of the header is different for each message type, the first 4 bytes are common to all. As the figure below shows:

- The first field, ICMP type, defines the type of the message.

- The code field specifies the reason for the particular message type.
- The last common field is the checksum field.
- The rest of the header is specific for each message type.



Error-reporting messages



Query messages

– The data section in error messages carries information for finding the original packet that had the error. In query messages, the data section carries extra information based on the type of query.

1- Error Reporting Messages

Since IP is an unreliable protocol, one of the main responsibilities of ICMP is to report some errors that may occur during the processing of the IP datagram. ICMP does not correct errors, it simply reports them. Error correction is left to the higher-level protocols.

Error messages are always sent to the original source because the only information available in the datagram about the route is the source and destination IP addresses.

ICMP uses the source IP address to send the error message to the source (originator) of the datagram.

To make the error-reporting process simple, ICMP follows some rules in reporting messages.

First, no error message will be generated for a datagram having a multicast address or special address (such as this host or loopback).

Second, no ICMP error message will be generated in response to a datagram carrying an ICMP error message.

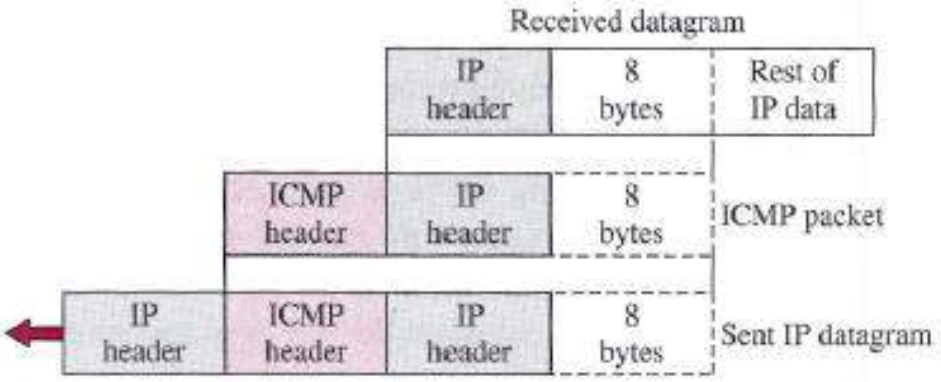
Third, no ICMP error message will be generated for a fragmented datagram that is not the first fragment.

Note that all error messages contain a data section that includes the IP header of the original datagram plus the first 8 bytes of data in that datagram.

- The original datagram header is added to give the original source, which receives the error message, information about the datagram itself.

- The 8 bytes of data are included because the first 8 bytes provide information about the port numbers (UDP and TCP) and sequence number (TCP). This information is needed so the source can inform the protocols (TCP or UDP) about the error.

ICMP forms an error packet, which is then encapsulated in an IP datagram (see the figure below).



Destination Unreachable (Type 3, Code 0 to 15)

The most widely used error message is the destination unreachable (type 3). This message uses different codes (0 to 15) to define the type of error message and the reason why a datagram has not reached its final destination.

For example, code 1 tells the source that a host is unreachable. This may happen, for example, when we use the HTTP protocol to access a web page, but the server is down. The message "destination host is not reachable" is created and sent back to the source.

Source Quench (Type 4, Code 0)

Another error message is called the source quench (type 4) message, which informs the sender that the network has encountered congestion and the datagram has been dropped; the source needs to slow down sending more datagrams. In other words, ICMP adds a kind of congestion control mechanism to the IP protocol by using this type of message.

Redirection Message (Type 5, Codes 0 to 3)

The redirection message (type 5) is used when the source uses a wrong router to send out its message. The router redirects the message to the appropriate router, but informs the source that it needs to change its default router in the future. The IP address of the default router is sent in the message.

Time exceeded (Type 11, Codes 0 and 1)

We discussed the purpose of the time-to-live (TTL) field in the IP datagram and explained that it prevents a datagram from being aimlessly circulated in the Internet. When the TTL value becomes 0, the datagram is dropped by the visiting router and a time exceeded message (type 11) with code 0 is sent to the source to inform it about the situation. The time-exceeded message (with code 1) can also be sent when not all fragments of a datagram arrive within a predefined period of time.

Parameter Problem (Type 12, Codes 0 and l)

A parameter problem message (type 12) can be sent when either there is a problem in the header of a datagram (code 0) or some options are missing or cannot be interpreted (code 1).

2- Query Messages

Interestingly, query messages in ICMP can be used independently without relation to an IP datagram. Of course, a query message needs to be encapsulated in a datagram, as a carrier. Query messages are used to probe or test the liveliness of hosts or routers in the Internet, find the one-way or the round-trip time for an IP datagram between two devices, or even find out whether the clocks in two devices are synchronized. Naturally, query messages come in pairs: request and reply.

Echo request and reply (Type 8 and 0, Code 0)

The echo request (type 8) and the echo reply (type 0) pair of messages are used by a host or a router to test the liveliness of another host or router. A host or router sends an echo request message to another host or router; if the latter is alive, it responds with an echo reply message. The applications of this pair in two debugging tools: ping and traceroute.

Timestamp request and reply (Type 13 and 14, Code 0)

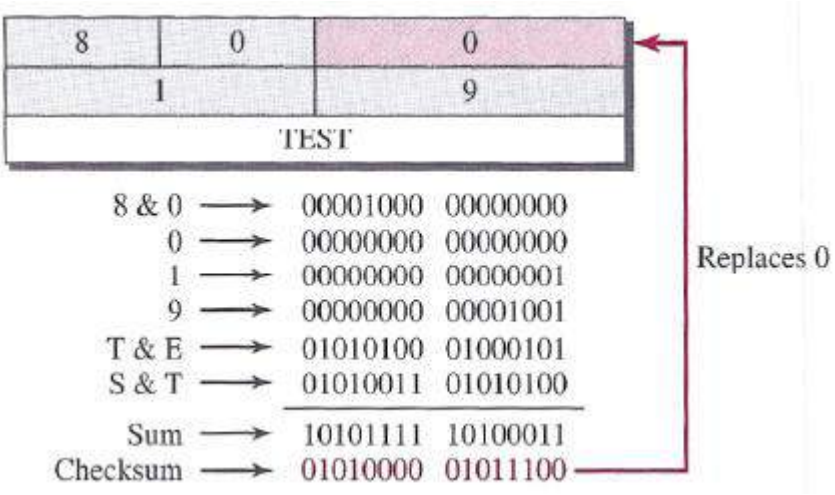
The timestamp request (type 13) and the timestamp reply (type 14) pair of messages are used to find the round-trip time between two devices or to check whether the clocks in two devices are synchronized. The timestamp request message sends a 32-bit number, which defines the time the message is sent. The timestamp reply resends that number, but also includes two new 32-bit numbers

representing the time the request was received and the time the response was sent. If all timestamps represent Universal time, the sender can calculate the one-way and round-trip time.

ICMP Checksum

In ICMP the checksum is calculated over the entire message (header and data).

Example: The figure below shows an example of checksum calculation for a simple echo-request message. We randomly chose the identifier to be 1 and the sequence number to be 9. The message is divided into 16-bit (2-byte) words. The words are added and the sum is complemented. Now the sender can put this value in the checksum field.



ICMP (Notable control messages)

Type	Code	Status	Description
0 – Echo Reply	0		Echo reply (used to ping)
1 and 2		Unassigned	Reserved
3 – Destination Unreachable	0		Destination network unreachable
	1		Destination host unreachable
	2		Destination protocol unreachable
	3		Destination port unreachable
	4		Fragmentation required, and DF flag set
	5		Source route failed
	6		Destination network unknown
	7		Destination host unknown

	8		Source host isolated
	9		Network administratively prohibited
	10		Host administratively prohibited
	11		Network unreachable for ToS
	12		Host unreachable for ToS
	13		Communication administratively prohibited
	14		Host Precedence Violation
	15		Precedence cutoff in effect
4 – Source Quench	0	Deprecated	Source quench (congestion control)
5 – Redirect Message	0		Redirect Datagram for the Network
	1		Redirect Datagram for the Host
	2		Redirect Datagram for the ToS & network
	3		Redirect Datagram for the ToS & host
6		Deprecated	Alternate Host Address
7		Unassigned	Reserved
8 – Echo Request	0		Echo request (used to ping)
9 – Router Advertisement	0		Router Advertisement
10 – Router Solicitation	0		Router discovery/selection/solicitation
11 – Time Exceeded	0		TTL expired in transit
	1		Fragment reassembly time exceeded
12 – Parameter Problem: Bad IP header	0		Pointer indicates the error
	1		Missing a required option
	2		Bad length
13 – Timestamp	0		Timestamp
14 – Timestamp Reply	0		Timestamp reply
15 – Information Request	0	Deprecated	Information Request
16 – Information Reply	0	Deprecated	Information Reply

17 – Address Mask Request	0	Deprecated	Address Mask Request
18 – Address Mask Reply	0	Deprecated	Address Mask Reply
19		Reserved	Reserved for security
20 through 29		Reserved	Reserved for robustness experiment
30 – Traceroute	0	Deprecated	Information Request
31		Deprecated	Datagram Conversion Error
32		Deprecated	Mobile Host Redirect
33		Deprecated	Where-Are-You (originally meant for IPv6)
34		Deprecated	Here-I-Am (originally meant for IPv6)
35		Deprecated	Mobile Registration Request
36		Deprecated	Mobile Registration Reply
37		Deprecated	Domain Name Request
38		Deprecated	Domain Name Reply
39		Deprecated	SKIP Algorithm Discovery Protocol, Simple Key-Management for Internet Protocol
40			Photuris, Security failures
41		Experimental	ICMP for experimental mobility protocols such as Seamoby [RFC4065]
42 through 252		Unassigned	Reserved
253		Experimental	RFC3692-style Experiment 1 (RFC 4727)
254		Experimental	RFC3692-style Experiment 2 (RFC 4727)
255		Reserved	Reserved

NEXT GENERATION IP (IPv6)

The address depletion of IPv4 and other shortcomings of this protocol prompted a new version of IP in the early 1990s. The new version, which is called Internet Protocol version 6 (IPv6) or IP new generation (IPng) was a proposal to augment the address space of IPv4 and at the same time redesign the format of the IP packet and revise some auxiliary protocols such as ICMP. It is interesting to know that IPv5 was a proposal, based on the OSI model, that never materialized.

The following lists the main changes in the IPv6 protocol: larger address space, better header format, new options, allowance for extension, support for resource allocation, and support for more security. The implementation of these changes made it necessary to create a new version of the ICMP protocol, ICMPv6.

Address Space

The main reason for migration from IPv4 to IPv6 is the small size of the address space in IPv4. An IPv6 address is 128 bits or 16 bytes (octets) long, four times the address length in IPv4.

The address space of IPv6 contains 2^{128} addresses. This address space is 296 times the IPv4 address–definitely no address depletion–as shown, the size of the space is:

$$340,282,366,920,938,463,374,607,431,768,211,456.$$

To give some idea about the number of addresses, we assume that only 1/64 (almost 2 percent) of the addresses in the space can be assigned to the people on planet Earth and the rest are reserved for special purposes. We also assume that the number of people on the earth is soon to be 2^{34} (more than 16 billion). Each person can have 2^{88} addresses to use. Address depletion in this version is impossible.

Representation

A computer normally stores the address in binary, but it is clear that 128 bits cannot easily be handled by humans. Several notations have been proposed to represent IPv6 addresses when they are handled by humans. The following shows two of these notations: binary and colon hexadecimal.

Binary (128 bits): 1111111011110110 1111111100000000

Colon Hexadecimal: FEF6: BA98: 7654: 3210: ADEF: BBFF: 2922: FF00

Binary notation is used when the addresses are stored in a computer. The colon hexadecimal notation (or colon hex for short) divides the address into eight sections, each made of four hexadecimal digits separated by colons.

Abbreviation

Although an IPv6 address, even in hexadecimal format, is very long, many of the digits are zeros. In this case, we can abbreviate the address. The leading zeros of a section can be omitted.

Using this form of abbreviation, 0074 can be written as 74, 000Fas F, and 0000 as 0. Note that 3210 cannot be abbreviated.

Further abbreviation, often called zero compression, can be applied to colon hex notation if there are consecutive sections consisting of zeros only. We can remove all the zeros and replace them with a double semicolon.

FDEC:0:0:0:0:BBFF:0:FFFF → FDEC: :BBFF:0:FFFF

Note that this type of abbreviation is allowed only once per address. If there is more than one run of zero sections, only one of them can be compressed.

Three Address Types

In IPv6, a destination address can belong to one of three categories: unicast, anycast, and multicast.

Unicast Address

A unicast address defines a single interface (computer or router). The packet sent to a unicast address will be routed to the intended recipient.

Anycast Address

An anycast address defines a group of computers that all share a single address. A packet with an anycast address is delivered to only one member of the group, the most reachable one. An anycast communication is used, for example, when there are several servers that can respond to an inquiry. The request is sent to the one that is most reachable. The hardware and software generate only one copy of the request; the copy reaches only one of the servers. IPv6 does not designate a block for anycasting; the addresses are assigned from the unicast block.

Multicast Address

A multicast address also defines a group of computers. However, there is a difference between anycasting and multicasting. In anycasting, only one copy of the packet is sent to one of the members of the group; in multicasting each member of the group receives a copy. IPv6 has designated a block for multicasting from which the same address is assigned to the members of the group. It is interesting that IPv6 does not define broadcasting, even in a limited version. IPv6 considers broadcasting as a special case of multicasting.

Computer Networks

Network Routing

When a device has multiple paths to reach a destination, it always selects one path by preferring it over others. This selection process is termed as Routing. Routing is done by special network devices called routers or it can be done by means of software processes. The software based routers have limited functionality and limited scope.

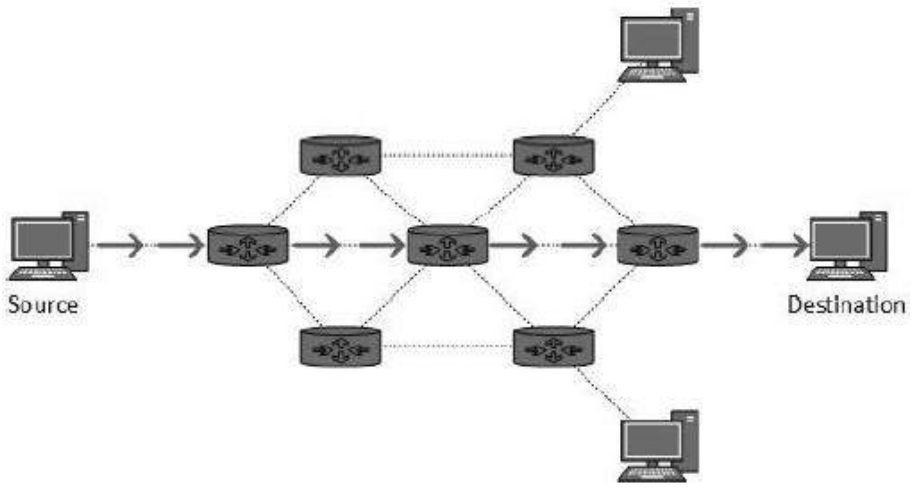
A router is always configured with some default route. A default route tells the router where to forward a packet if there is no route found for specific destination. In case there are multiple path existing to reach the same destination, router can make decision based on the following information:

- Hop Count
- Bandwidth
- Metric
- Prefix-length
- Delay

Routes can be statically configured or dynamically learnt. One route can be configured to be preferred over others.

Unicast routing

Most of the traffic on the internet and intranets known as unicast data or unicast traffic is sent with specified destination. Routing unicast data over the internet is called unicast routing. It is the simplest form of routing because the destination is already known. Hence the router just has to look up the routing table and forward the packet to next hop.



Unicast Routing Protocols

There are two kinds of routing protocols available to route unicast packets:

1- Distance Vector Routing Protocol

Distance Vector is simple routing protocol which takes routing decision on the number of hops between source and destination. A route with less number of hops is considered as the best route. Every router advertises its set best routes to other routers. Ultimately, all routers build up their network topology based on the advertisements of their peer routers, for example, Routing Information Protocol (RIP).

2- Link State Routing Protocol

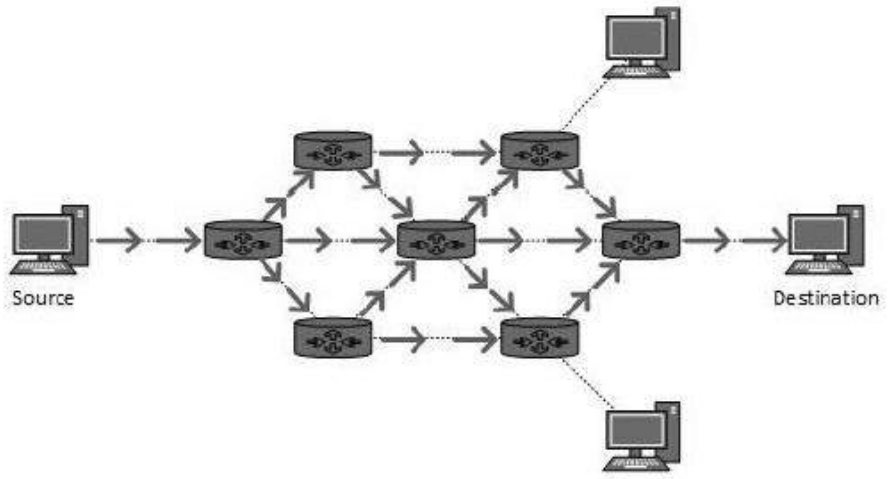
Link State protocol is slightly complicated protocol than Distance Vector. It takes into account the states of links of all the routers in a network. This technique helps routes build a common graph of the entire network. All routers then calculate their best path for routing purposes, for example, Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (ISIS).

Broadcast routing

A broadcast message is destined to all network devices. Broadcast routing can be done in two ways (algorithm):

1- A router creates a data packet and then sends it to each host one by one. In this case, the router creates multiple copies of single data packet with different destination addresses. All packets are sent as unicast but because they are sent to all, it simulates as if router is broadcasting. This method consumes lots of bandwidth and router must destination address of each node.

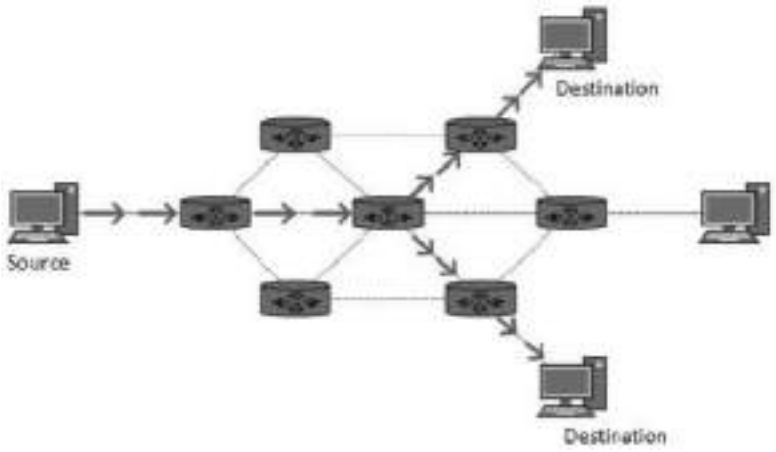
2- when router receives a packet that is to be broadcasted, it simply floods those packets out of all interfaces. This method is easy on router's CPU but may cause the problem of duplicate packets received from peer routers.



Multicast Routing

Multicast routing is special case of broadcast routing with significance difference and challenges. In broadcast routing, packets are sent to all nodes even if they do not want it. But in Multicast routing, the data is sent to only nodes which wants to receive the packets.

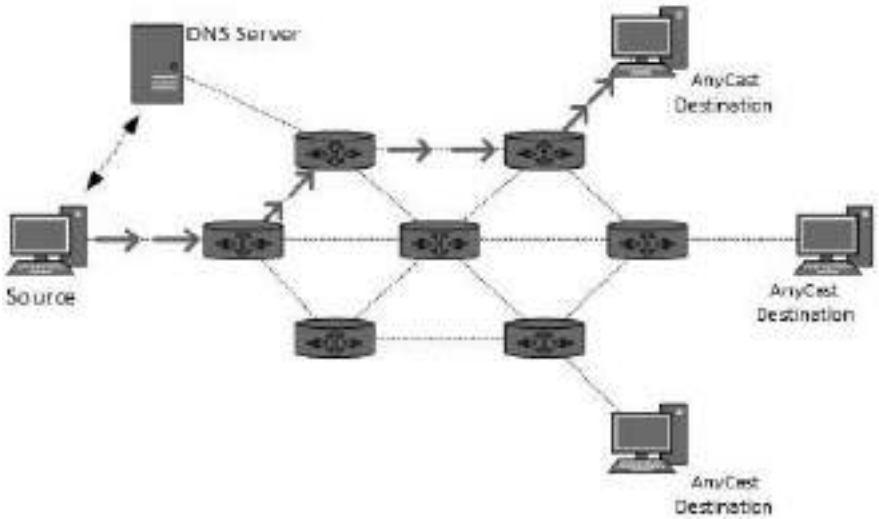
The router must know that there are nodes, which wish to receive multicast packets (or stream) then only it should forward.



Anycast Routing

Anycast packet forwarding is a mechanism where multiple hosts can have same logical address. When a packet destined to this logical address is received, it is sent to the host which is nearest in routing topology.

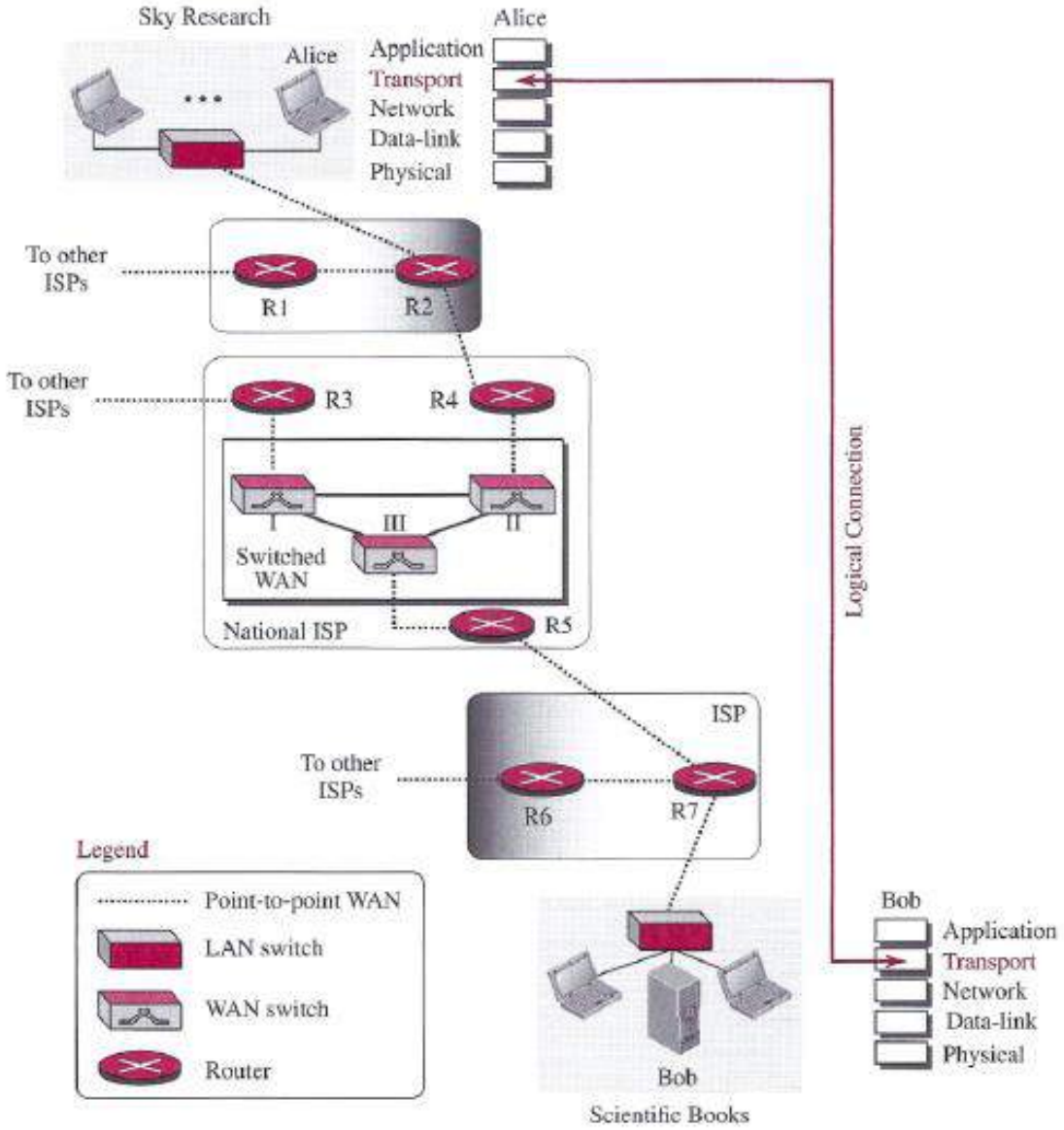
Anycast routing is done with help of DNS server. Whenever an Anycast packet is received it is enquired with DNS to where to send it. DNS provides the IP address which is the nearest IP configured on it.



Computer Networks

THE TRANSPORT LAYER

The transport layer is located between the application layer and the network layer. It provides a process-to-process communication between two application layers, one at the local host and the other at the remote host. Communication is provided using a logical connection, which means that the two application layers, which can be located in different parts of the globe, assume that there is an imaginary direct connection through which they can send and receive messages. The figure below shows the idea behind this logical connection.



The figure shows the same scenario we used in the physical layer. Alice's host in the Sky Research company creates a logical connection with Bob's host in the Scientific Books company at the transport layer. The two companies communicate at the transport layer as though there is a real connection between them. The figure shows that only the two end systems (Alice's and Bob's computers) use the services of the transport layer; all intermediate routers use only the first three layers.

Computer Networks

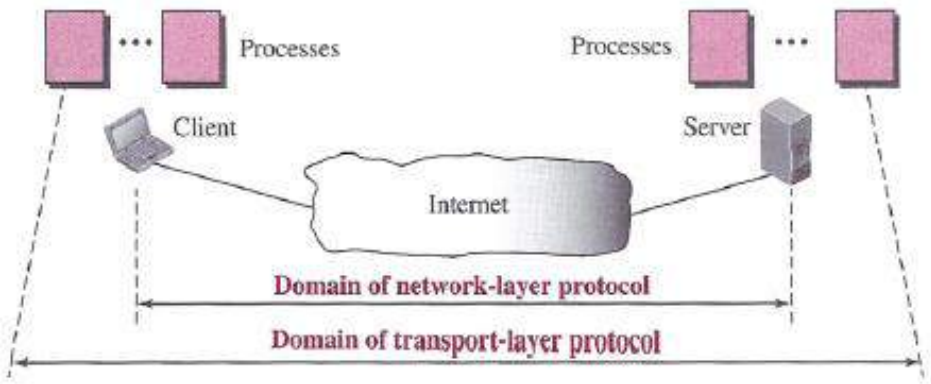
TRANSPORT LAYER SERVICES

1- Process-to-Process Communication

The first duty of a transport-layer protocol is to provide process-to-process communication. A process is an application-layer entity (running program) that uses the services of the transport layer.

Before we discuss how process-to-process communication can be accomplished, we need to understand the difference between host-to-host communication and process-to-process communication.

The network layer is responsible for communication at the computer level (host-to-host communication). A network-layer protocol can deliver the message only to the destination computer. However, this is an incomplete delivery. The message still needs to be handed to the correct process. This is where a transport-layer protocol takes over. A transport-layer protocol is responsible for delivery of the message to the appropriate process. The figure below shows the domains of a network layer and a transport layer.



2- Addressing: Port Numbers

Although there are a few ways to achieve process-to-process communication, the most common is through the client-server paradigm. A process on the local host, called a client, needs services from a process usually on the remote host, called a server. However, operating systems today support both multiuser and multiprogramming environments. A remote computer can run several server programs at the same time, just as several local computers can run one or more client programs at the same time. For communication, we must define the local host, local process, remote host, and remote process. The local host and the remote host are defined using IP addresses. To define the processes, we need second identifiers, called port numbers. In the TCP/IP protocol suite, the port numbers are integers between 0 and 65,535 (16 bits).

The client program defines itself with a port number, called the ephemeral port number. The word ephemeral means "short-lived" and is used because the life of a client is normally short. An ephemeral port number is recommended to be greater than 1023 for some client/server programs to work properly.

The server process must also define itself with a port number. This port number, however, cannot be chosen randomly. If the computer at the server site runs a server process and assigns a random number as the port number, the process at the client site that wants to access that server and use its services will not know the port number. Of course, one solution would be to send a special packet and request the port number of a specific server, but this creates more overhead.

TCP/IP has decided to use universal port numbers for servers; these are called well-known port numbers. There are some exceptions to this rule; for example, there are clients that are assigned well-known port numbers. Every client process knows the well-known port number of the corresponding server process. For example, while the daytime client process, a well-known client program, can use an ephemeral (temporary) port number, 52,000, to identify itself, the daytime server process must use the well-known (permanent) port number 13. The figure below shows this concept.

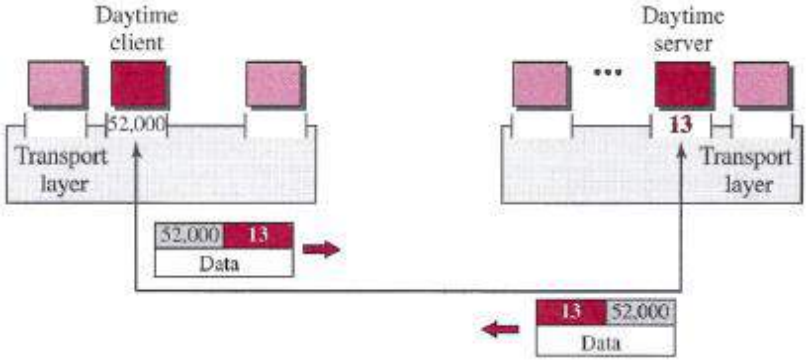
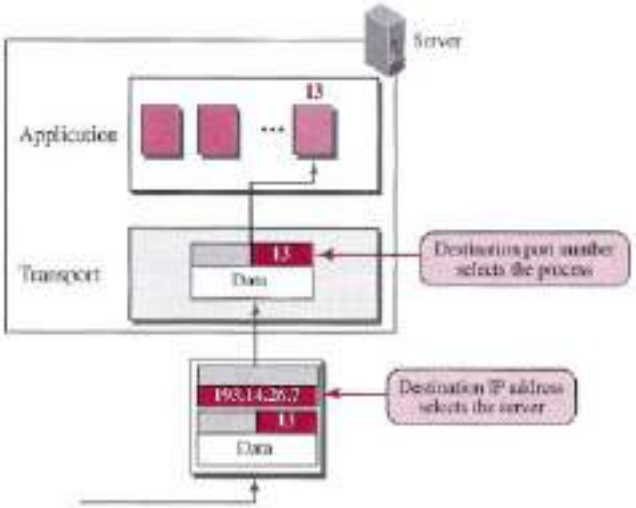


Table below gives some common port numbers for all three protocols:

Port	Protocol	UDP	TCP	SCTP	Description
7	Echo	√	√	√	Echoes back a received datagram
9	Discard	√	√	√	Discards any datagram that is received
11	Users	√	√	√	Active users
13	Daytime	√	√	√	Returns the date and the time
17	Quote	√	√	√	Returns a quote of the day
19	Chargen	√	√	√	Returns a string of characters
20	FTP-data		√	√	File Transfer Protocol
21	FTP-21		√	√	File Transfer Protocol
23	TELNET		√	√	Terminal Network
25	SMTP		√	√	Simple Mail Transfer Protocol
53	DNS	√	√	√	Domain Name Service
67	DHCP	√	√	√	Dynamic Host Configuration Protocol
69	TFTP	√	√	√	Trivial File Transfer Protocol
80	HTTP		√	√	HyperText Transfer Protocol
111	RPC	√	√	√	Remote Procedure Call
123	NTP	√	√	√	Network Time Protocol
161	SNMP-server	√			Simple Network Management Protocol
162	SNMP-client	√			Simple Network Management Protocol

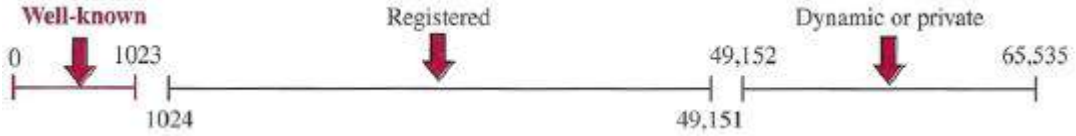
IP addresses versus port numbers

It should be clear by now that the IP addresses and port numbers play different roles in selecting the final destination of data. The destination IP address defines the host among the different hosts in the world. After the host has been selected, the port number defines one of the processes on this particular host (see the figure below).



ICANN Ranges

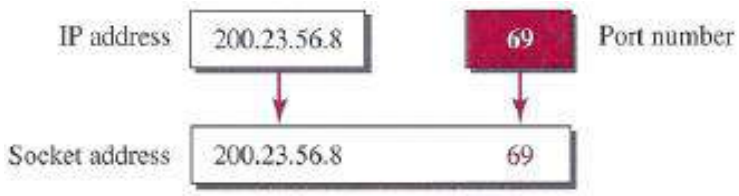
Internet Corporation for Assigned Names and Numbers (ICANN) has divided the port numbers into three ranges: well-known, registered, and dynamic (or private), as shown in the figure below.



- **Well-known ports**: The ports ranging from 0 to 1023 are assigned and controlled by ICANN.
- **Registered ports**: The ports ranging from 1024 to $49,151$ are not assigned or controlled by ICANN.
- **Dynamic ports**: The ports ranging from $49,152$ to $65,535$ are neither controlled nor registered. They can be used as temporary or private port numbers.

Socket Addresses

A transport-layer protocol in the TCP suite needs both the IP address and the port number, at each end, to make a connection. The combination of an IP address and a port number is called a socket address. The client socket address defines the client process uniquely just as the server socket address defines the server process uniquely (see the figure below).

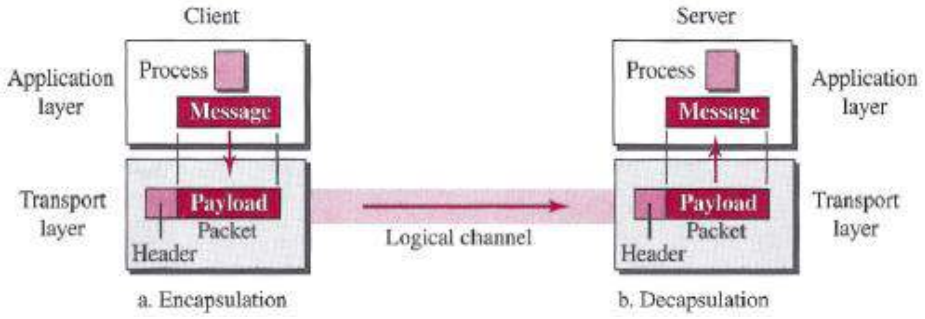


3- Encapsulation and Decapsulation

To send a message from one process to another, the transport-layer protocol encapsulates and decapsulates messages (see the figure below).

Encapsulation happens at the sender site. When a process has a message to send, it passes the message to the transport layer along with a pair of socket addresses and some other pieces of information, which depend on the transport-layer protocol. The transport layer receives the data and adds the transport-layer header. The packets at the transport layer in the Internet are called user datagrams, segments, or packets, depending on what transport-layer protocol we use. In general discussion, we refer to transport-layer payloads as packets.

Decapsulation happens at the receiver site. When the message arrives at the destination transport layer, the header is dropped and the transport layer delivers the message to the process running at the application layer. The sender socket address is passed to the process in case it needs to respond to the message received.

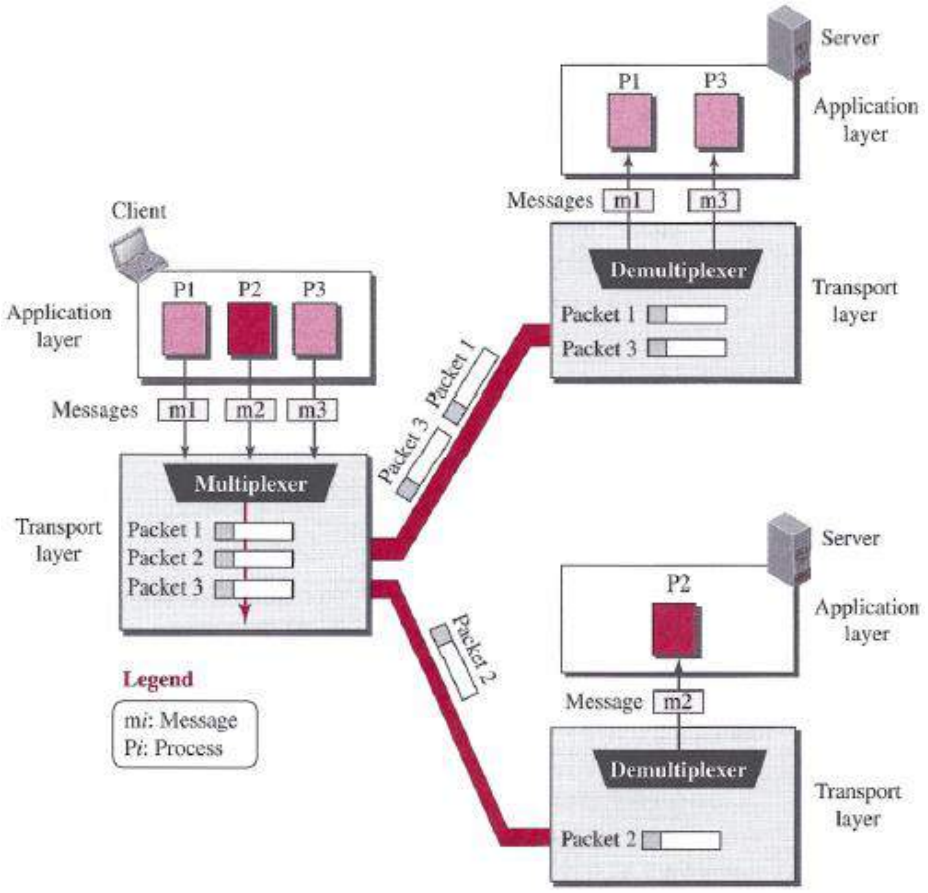


4- Multiplexing and Demultiplexing

Whenever an entity accepts items from more than one source, this is referred to as multiplexing (many to one); whenever an entity delivers items to more than one source, this is referred to as demultiplexing (one to many). The transport layer at the source performs multiplexing; the transport layer at the destination performs demultiplexing.

The figure below shows communication between a client and two servers. Three client processes are running at the client site, PI, P2, and P3. The processes PI and P3 need to send requests to the corresponding server process running in a server. The client process P2 needs to send a request to the corresponding server process running at another server. The transport layer at the client site accepts three messages from the three processes and creates three packets. It acts as a multiplexer. The packets 1 and 3 use the same logical channel to reach the transport layer of the first server. When they arrive at the server, the transport layer does the job of a demultiplexer and distributes the messages to two different processes. The transport layer at the second server receives packet 2 and delivers it to the corresponding process. Note that we still have demultiplexing although there is only one message.

Computer Networks

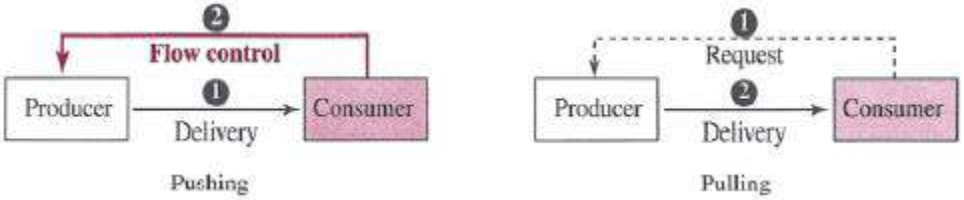


5- Flow Control

Whenever an entity produces items and another entity consumes them, there should be a balance between production and consumption rates. If the items are produced faster than they can be consumed, the consumer can be overwhelmed and may need to discard some items. If the items are produced more slowly than they can be consumed, the consumer must wait, and the system becomes less efficient. Flow control is related to the first issue. We need to prevent losing the data items at the consumer site.

Pushing or Pulling

Delivery of items from a producer to a consumer can occur in one of two ways: pushing or pulling. If the sender delivers items whenever they are produced–without a prior request from the consumer–the delivery is referred to as pushing. If the producer delivers the items after the consumer has requested them, the delivery is referred to as pulling. The figure below shows these two types of delivery.

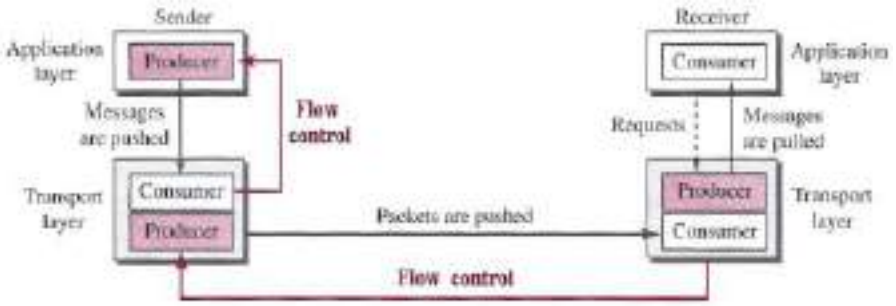


When the producer pushes the items, the consumer may be overwhelmed and there is a need for flow control, in the opposite direction, to prevent discarding of the items. In other words, the consumer needs to warn the producer to stop the delivery and to inform the producer when it is again ready to receive the items. When the consumer pulls the items, it requests them when it is ready. In this case, there is no need for flow control.

Flow Control at Transport Layer

In communication at the transport layer, we are dealing with four entities: sender process, sender transport layer, receiver transport layer, and receiver process. The sending process at the application layer is only a producer. It produces message chunks and pushes them to the transport layer. The sending transport layer has a double role: it is both a consumer and a producer. It consumes the messages pushed by the producer. It encapsulates the messages in packets and pushes them to the receiving transport layer. The receiving transport layer also has a double role: it is the consumer for the packets received from the sender and the producer that decapsulates the messages and delivers them to the application layer. The last delivery, however, is normally a pulling delivery; the transport layer waits until the application-layer process asks for messages.

The figure below shows that we need at least two cases of flow control: from the sending transport layer to the sending application layer and from the receiving transport layer to the sending transport layer.



Buffers

Although flow control can be implemented in several ways, one of the solutions is normally to use two buffers: one at the sending transport layer and the other at the receiving transport layer. A buffer is a set of memory locations that can hold packets at the sender and receiver. The flow control communication can occur by sending signals from the consumer to the producer.

When the buffer of the sending transport layer is full, it informs the application layer to stop passing chunks of messages; when there are some vacancies, it informs the application layer that it can pass message chunks again.

When the buffer of the receiving transport layer is full, it informs the sending transport layer to stop sending packets. When there are some vacancies, it informs the sending transport layer that it can send packets again.

6- Error Control

In the Internet, since the underlying network layer (IP), which is responsible to carry the packets from the sending transport layer to the receiving transport layer, is unreliable, we need to make the transport layer reliable if the application requires reliability. Reliability can be achieved to add error control service to the transport layer. Error control at the transport layer is responsible to:

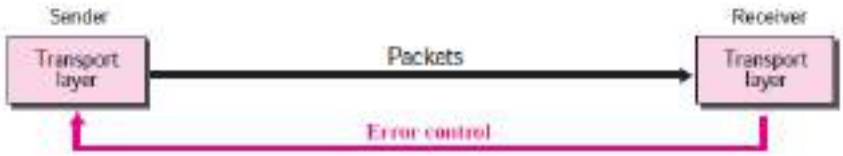
1- Detect and discard corrupted packets.

2- Keep track of lost and discarded packets and resend them.

3- Recognize duplicate packets and discard them.

4- Buffer out-of-order packets until the missing packets arrive.

Error control, unlike the flow control, involves only the sending and receiving transport layers. We are assuming that the message chunks exchanged between the application and transport layers are error free. The figure below shows the error control between the sending and receiving transport layer. As with the case of flow control, the receiving transport layer manages error control, most of the time, by informing the sending transport layer about the problems.



Sequence Numbers

Error control requires that the sending transport layer knows which packet is to be resent and the receiving transport layer knows which packet is a duplicate, or which packet has arrived out of order. This can be done if the packets are numbered. We can add a field to the transport-layer packet to hold the sequence number of the packet. When a packet is corrupted or lost, the receiving transport layer can somehow inform the sending transport layer to resend that packet using the sequence number. The receiving transport layer can also detect duplicate packets if two received packets have the same sequence number. The out-of-order packets can be recognized by observing gaps in the sequence numbers. Packets are numbered sequentially. However, because we need to include the sequence number of each packet in the header, we need to set a limit. If the header of the packet allows m bits for the sequence number, the sequence numbers range from 0 to 2^{m-1}. For example, if

m is 4, the only sequence numbers are 0 through 15, inclusive. However, we can wrap around the sequence. So the sequence numbers in this case are

0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,0,1,2,3,4,5,6, 7, 8, 9, 10,11,...

Acknowledgment

The receiver side can send an acknowledgment (ACK) for each of a collection of packets that have arrived safe and sound. The receiver can simply discard the corrupted packets. The sender can detect lost packets if it uses a timer. When a packet is sent, the sender starts a timer. If an ACK does not arrive before the timer expires, the sender resends the packet. Duplicate packets can be silently discarded by the receiver. Out-of-order packets can be either discarded (to be treated as lost packets by the sender), or stored until the missing one arrives.

7- Congestion Control

An important issue in a packet-switched network, such as the Internet, is congestion. Congestion in a network may occur if the load on the network (the number of packets sent to the network) is greater than the capacity of the network (the number of packets a network can handle). Congestion control refers to the mechanisms and techniques that control the congestion and keep the load below the capacity.

CONNECTIONLESS AND CONNECTION-ORIENTED PROTOCOLS

A transport-layer protocol, like a network-layer protocol, can provide two types of services: connectionless and connection-oriented. The nature of these services at the transport layer, however, is different from the ones at the network layer. At the network layer, a connectionless service may mean different paths for different datagrams belonging to the same message. At the transport layer, we are not concerned about the physical paths of packets (we assume a logical connection between two transport layers). Connectionless service at the transport layer means independency between packets; connection-oriented means dependency. Let us elaborate on these two services.

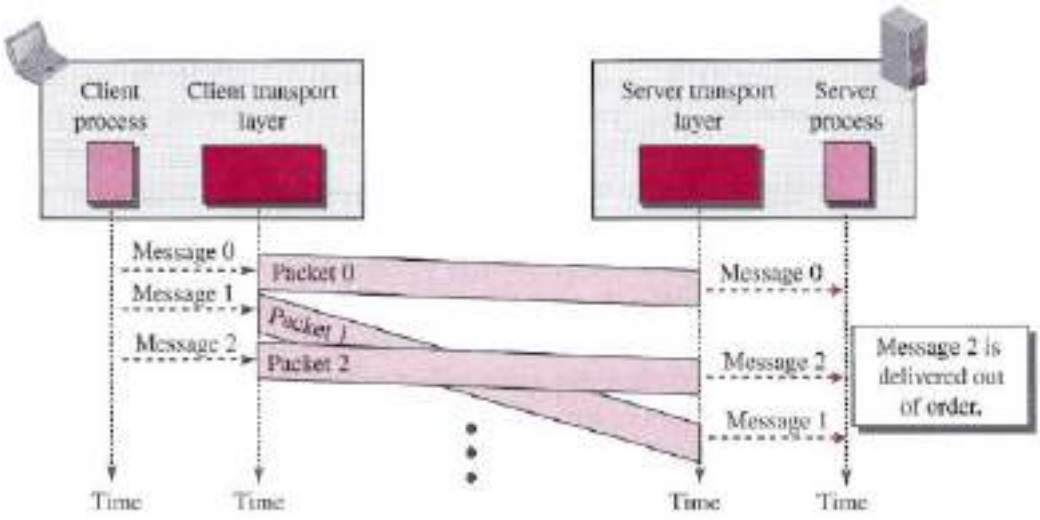
1- Connectionless Service

In a connectionless service, the source process (application program) needs to divide its message into chunks of data of the size acceptable by the transport layer and deliver them to the transport layer one by one. The transport layer treats each chunk as a single unit without any relation between the chunks. When a chunk arrives from the application layer, the transport layer encapsulates it in a packet and sends it. To show the independency of packets, assume that a client process has three chunks of

messages to send to a server process. The chunks are handed over to the connectionless transport protocol in order. However, since there is no dependency between the packets at the transport layer, the packets may arrive out of order at the destination and will be delivered out of order to the server process (See the figure below).

In the figure below, we have shown the movement of packets using a time line, but we have assumed that the delivery of the process to the transport layer and vice versa are instantaneous. The figure shows that at the client site, the three chunks of messages are delivered to the client transport layer in order (0, 1, and 2). Because of the extra delay in transportation of the second packet, the delivery of messages at the server is not in order (0, 2, 1). If these three chunks of data belong to the same message, the server process may have received a strange message.

The situation would be worse if one of the packets were lost. Since there is no numbering on the packets, the receiving transport layer has no idea that one of the messages has been lost. It just delivers two chunks of data to the server process. The above two problems arise from the fact that the two transport layers do not coordinate with each other. The receiving transport layer does not know when the first packet will come nor when all of the packets have arrived. We can say that no flow control, error control, or congestion control can be effectively implemented in a connectionless service.



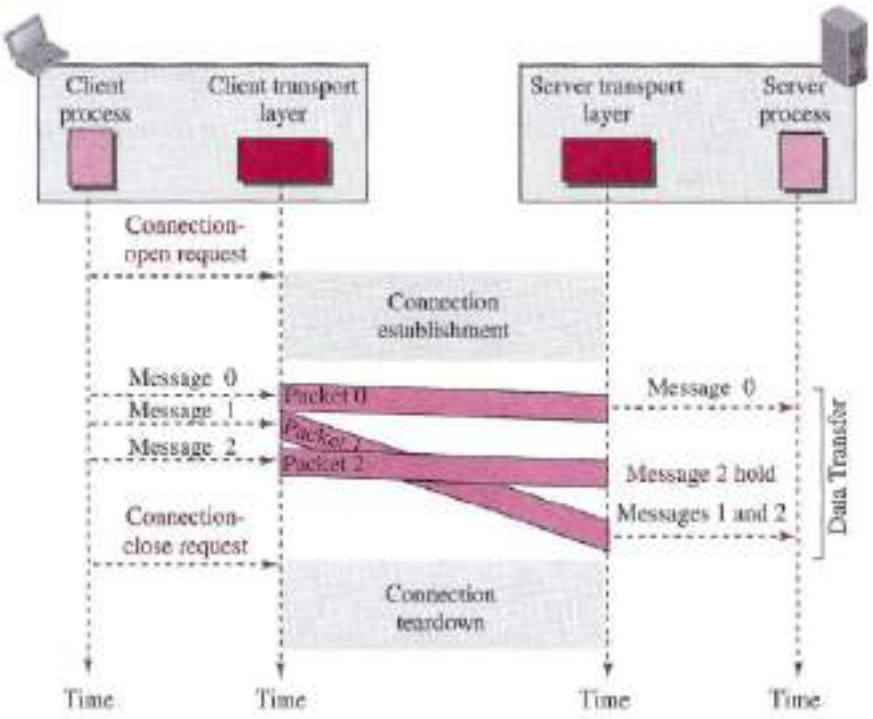
Connection-Oriented Service

In a connection-oriented service, the client and the server first need to establish a logical connection between themselves. The data exchange can only happen after the connection establishment. After data exchange, the connection needs to be torn down (See the figure below).

As we mentioned before, the connection-oriented service at the transport layer is different from the same service at the network layer. In the network layer, connection oriented service means a coordination between the two end hosts and all the routers in between. At the transport layer,

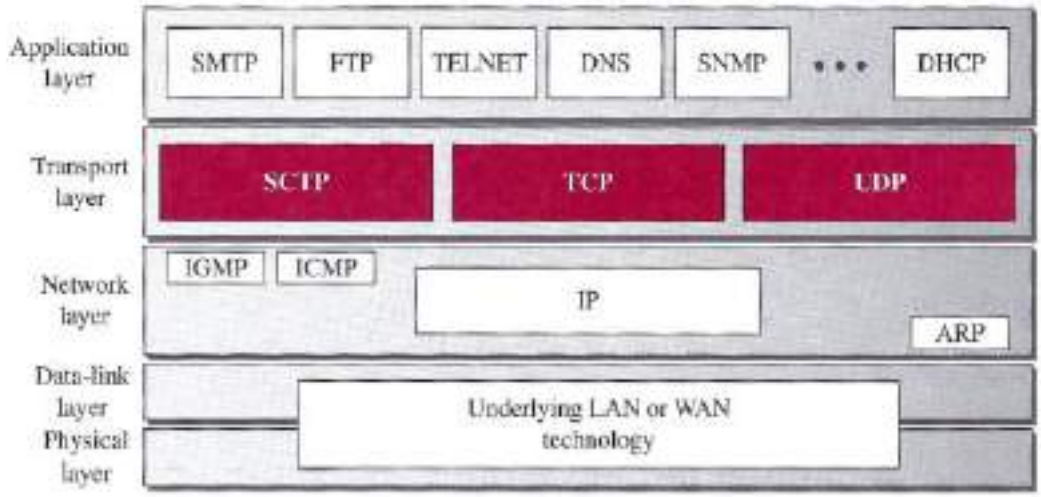
Computer Networks

connection-oriented service involves only the two hosts; the service is end to end. This means that we should be able to make a connection-oriented protocol at the transport layer over either a connectionless or connection-oriented protocol at the network layer. The figure below shows the connection establishment, data-transfer, and tear-down phases in a connection-oriented service at the transport layer. We can implement flow control, error control, and congestion control in a connection oriented protocol.



TRANSPORT LAYER PROTOCOLS

After discussing the general principle behind the transport layer in the previous chapter, we concentrate on the transport protocols in the Internet in this chapter. The figure below shows the position of these three protocols in the TCPIIP protocol suite.



Services

Each protocol provides a different type of service and should be used appropriately:

1- **UDP**: UDP is an unreliable connectionless transport-layer protocol used for its simplicity and efficiency in applications where error control can be provided by the application-layer process.

2- **TCP**: TCP is a reliable connection-oriented protocol that can be used in any application where reliability is important.

3- **SCTP**: SCTP is a new transport-layer protocol that combines the features of UDP and TCP.

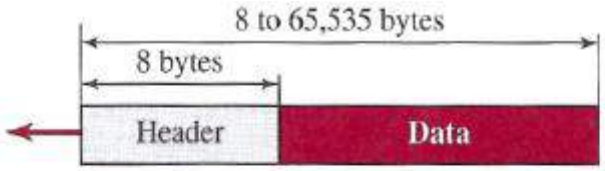
USER DATAGRAM PROTOCOL (UDP)

The User Datagram Protocol (UDP) is a connectionless, unreliable transport protocol. It does not add anything to the services of IP except for providing process-to-process communication instead of host-to-host communication. If UDP is so powerless, why would a process want to use it? With the disadvantages come some advantages. UDP is a very simple protocol using a minimum of overhead. If a process wants to send a small message and does not care much about reliability, it can use UDP. Sending a small message using UDP takes much less interaction between the sender and receiver than using TCP.

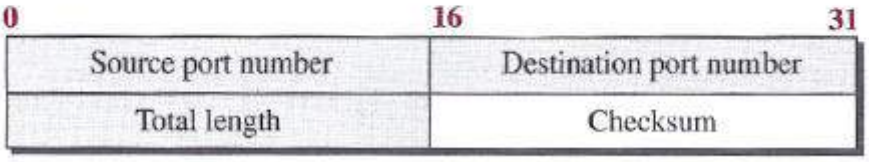
UDP packet

UDP packets, called user datagrams, have a fixed-size header of 8 bytes made of four fields, each of 2 bytes (16 bits). The figure below shows the format of a user datagram.

- The first two fields define the source and destination port numbers.
- The third field defines the total length of the user datagram, header plus data. The16 bits can define a total length of 0 to 65,535 bytes. However, the total length needs to be less because a UDP user datagram is stored in an IP datagram with the total length of 65,535 bytes.
- The last field can carry the optional checksum.



UDP user datagram



Header format

Example: The following is the content of a UDP header in hexadecimal format.

CB84000D001C001C

a- What is the source port number?

b- What is the destination port number?

c- What is the total length of the user datagram?

d- What is the length of the data?

e- Is the packet directed from a client to a server or vice versa?

f- What is the client process?

Solution

a- The source port number is the first four hexadecimal digits $(CB84)16$, which means that the source port number is 52100.

b- The destination port number is the second four hexadecimal digits $(000D)16$, which means that the destination port number is 13.

c- The third four hexadecimal digits $(001C)16$ define the length of the whole UDP packet as 28 bytes.

d- The length of the data is the length of the whole packet minus the length of the header, or $28 - 8 = 20$ bytes.

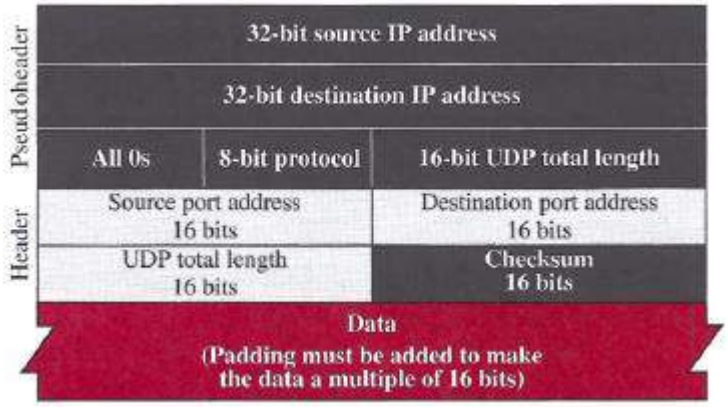
e- Since the destination port number is 13 (well-known port), the packet is from the client to the server.

f- The client process is the Daytime.

Checksum

UDP checksum calculation includes three sections: a pseudo-header, the UDP header, and the data coming from the application layer.

The pseudo-header is the part of the header of the IP packet in which the user datagram is to be encapsulated with some fields filled with 0s (see the figure below).



- If the checksum does not include the pseudo-header, a user datagram may arrive safe and sound. However, if the IP header is corrupted, it may be delivered to the wrong host.
- The protocol field is added to ensure that the packet belongs to UDP, and not to TCP. The value of the protocol field for UDP is 17. If this value is changed during transmission, the checksum calculation at the receiver will detect it and UDP drops the packet. It is not delivered to the wrong protocol.

Optional Inclusion of Checksum

The sender of a UDP packet can choose not to calculate the checksum. In this case, the checksum field is filled with all 0s before being sent.

In the situation where the sender decides to calculate the checksum, but it happens that the result is all 0s, the checksum is changed to all 1s before the packet is sent. In other words, the sender complements the sum two times. Note that this does not create confusion because the value of the checksum is never all 1s in a normal situation.

Example: What value is sent for the checksum in each one of the following hypothetical situations?

a- The sender decides not to include the checksum.

b- The sender decides to include the checksum, but the value of the sum is all 1s.

c- The sender decides to include the checksum, but the value of the sum is all 0s.

Solution

a- The value sent for the checksum field is all 0s to show that the checksum is not calculated.

b- When the sender complements the sum, the result is all 0s; the sender complements the result again before sending. The value sent for the checksum is all 1s. The second complement operation is needed to avoid confusion with the case in part a.

c- This situation never happens because it implies that the value of every term included in the calculation of the sum is all 0s, which is impossible; some fields in the pseudo-header have nonzero values.

UDP Services

Earlier we discussed the general services provided by a transport-layer protocol. In this section, we discuss what portions of those general services are provided by UDP.

1- Process-to-Process Communication

UDP provides process-to-process communication using socket addresses, a combination of IP addresses and port numbers.

2- Connectionless Services

As mentioned previously, UDP provides a connection less service. This means that each user datagram sent by UDP is an independent datagram. There is no relationship between the different user datagrams even if they are coming from the same source process and going to the same destination program. The user datagrams are not numbered. Also, unlike TCP, there is no connection establishment and no connection termination. This means that each user datagram can travel on a different path.

One of the ramifications of being connectionless is that the process that uses UDP cannot send a stream of data to UDP and expect UDP to chop them into different, related user datagrams. Instead each request must be small enough to fit into one user datagram.

Only those processes sending short messages, messages less than 65,507 bytes (65,535 minus 8 bytes for the UDP header and minus 20 bytes for the IP header), can use UDP.

UDP Applications

The following shows some typical applications that can benefit more from the services of UDP than from those of TCP.

1- UDP is suitable for a process that requires simple request-response communication with little concern for flow and error control. It is not usually used for a process such as FTP that needs to send bulk data.

2- UDP is suitable for a process with internal flow- and error-control mechanisms. For example, the Trivial File Transfer Protocol (TFTP) process includes flow and error control. It can easily use UDP.

3- UDP is a suitable transport protocol for multicasting. Multicasting capability is embedded in the UDP software but not in the TCP software.

4- UDP is used for management processes such as SNMP.

5- UDP is used for some route updating protocols such as Routing Information Protocol (RIP).

6- UDP is normally used for interactive real-time applications that cannot tolerate uneven delay between sections of a received message.

TRANSMISSION CONTROL PROTOCOL (TCP)

Transmission Control Protocol (TCP) is a connection-oriented, reliable protocol. TCP explicitly defines connection establishment, data transfer, and connection teardown phases to provide a connection-oriented service. TCP uses a combination of Go-Back-N (GBN) and Selective Repeat (SR) protocols to provide reliability. To achieve this goal, TCP uses checksum (for error detection), retransmission of lost or corrupted packets, cumulative and selective acknowledgments, and timers.

- Cumulative acknowledgements: A single acknowledgement informing the sender that all the frames up to a certain number have been received.
- Selective acknowledgements: Acknowledgement for a particular frame.

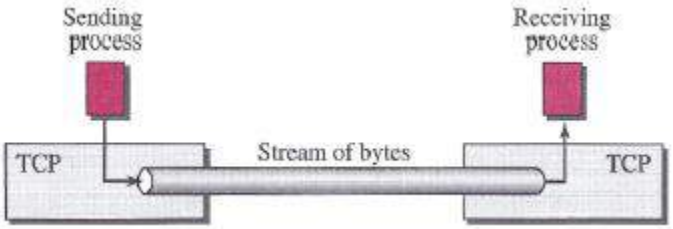
TCP SERVICES

1- Process-to-Process Communication

As with UDP, TCP provides process-to-process communication using port numbers.

2- Stream Delivery Service

TCP, unlike UDP, is a stream-oriented protocol. In UDP, a process sends messages with predefined boundaries to UDP for delivery. UDP adds its own header to each of these messages and delivers it to IP for transmission. Each message from the process is called a user datagram, and becomes, eventually, one IP datagram. Neither IP nor UDP recognizes any relationship between the datagrams. TCP, on the other hand, allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes. TCP creates an environment in which the two processes seem to be connected by an imaginary "tube" that carries their bytes across the Internet. This imaginary environment is depicted in the figure below. The sending process produces (writes to) the stream and the receiving process consumes (reads from) it.



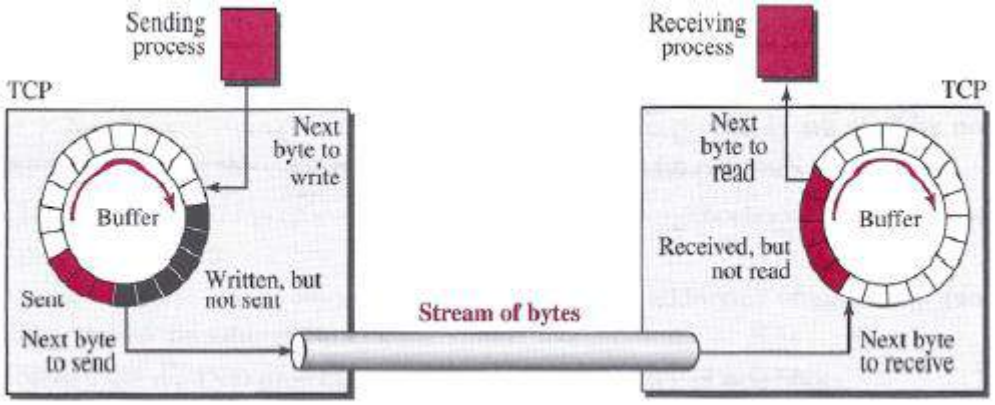
Sending and Receiving Buffers

Because the sending and the receiving processes may not necessarily write or read data at the same rate, TCP needs buffers for storage. There are two buffers, the sending buffer and the receiving buffer, one for each direction. One way to implement a buffer is to use a circular array of 1-byte locations as shown in the figure below. For simplicity, we have shown two buffers of 20 bytes each; normally the

buffers are hundreds or thousands of bytes, depending on the implementation. We also show the buffers as the same size, which is not always the case.

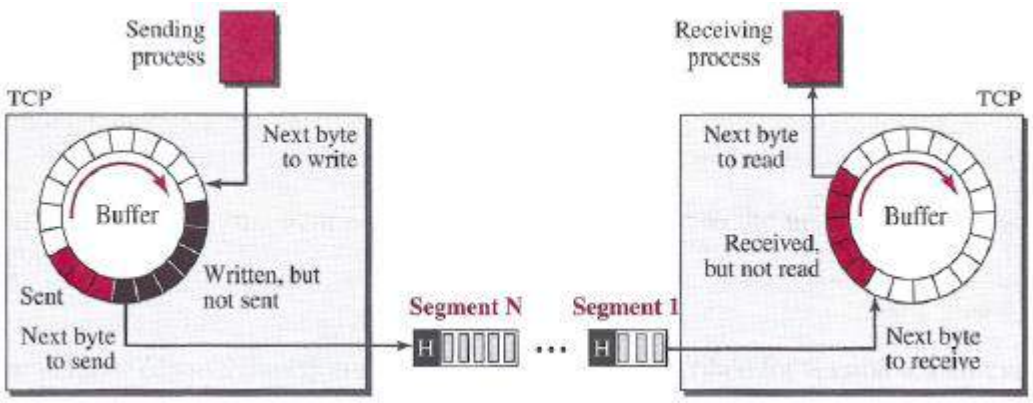
The figure shows the movement of the data in one direction.

- At the sender, the buffer has three types of chambers:
  - ❖ The white section contains empty chambers that can be filled by the sending process (producer).
  - ❖ The colored area holds bytes that have been sent but not yet acknowledged. The TCP sender keeps these bytes in the buffer until it receives an acknowledgment. After the bytes in the colored chambers are acknowledged, the chambers are recycled and available for use by the sending process. This is why we show a circular buffer.
  - ❖ The shaded area contains bytes to be sent by the sending TCP.
- The operation of the buffer at the receiver is simpler. The circular buffer is divided into two areas (shown as white and colored).
  - ❖ The white area contains empty chambers to be filled by bytes received from the network.
  - ❖ The colored sections contain received bytes that can be read by the receiving process. When a byte is read by the receiving process, the chamber is recycled and added to the pool of empty chambers.



Segments

Although buffering handles the disparity between the speed of the producing and consuming processes, we need one more step before we can send data. The network layer, as a service provider for TCP, needs to send data in packets, not as a stream of bytes. At the transport layer, TCP groups a number of bytes together into a packet called a segment. TCP adds a header to each segment (for control purposes) and delivers the segment to the network layer for transmission. The segments are encapsulated in an IP datagram and transmitted. This entire operation is transparent to the receiving process. The figure below shows how segments are created from the bytes in the buffers. Note that segments are not necessarily all the same size.



3- Full-Duplex Communication

TCP offers full-duplex service, where data can flow in both directions at the same time. Each TCP endpoint then has its own sending and receiving buffer, and segments move in both directions.

4- Multiplexing and Demultiplexing

Like UDP, TCP performs multiplexing at the sender and de multiplexing at the receiver. However, since TCP is a connection-oriented protocol, a connection needs to be established for each pair of processes.

5- Connection-Oriented Service

TCP, unlike UDP, is a connection-oriented protocol. When a process at site A wants to send to and receive data from another process at site B, the following three phases occur:

1- The two TCP's establish a logical connection between them.

2- Data are exchanged in both directions.

3- The connection is terminated.

Note that this is a logical connection, not a physical connection. The TCP segment is encapsulated in an IP datagram and can be sent out of order, or lost or corrupted, and then resent. Each may be routed over a different path to reach the destination. There is no physical connection. TCP creates a stream-oriented environment in which it accepts the responsibility of delivering the bytes in order to the other site.

6- Reliable Service

TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data.

NUMBERING SYSTEM

Although the TCP software keeps track of the segments being transmitted or received, there is no field for a segment number value in the segment header. Instead, there are two fields, called the sequence number and the acknowledgment number. These two fields refer to a byte number and not a segment number.

Byte Number

TCP numbers all data bytes (octets) that are transmitted in a connection. Numbering is independent in each direction. When TCP receives bytes of data from a process, TCP stores them in the sending buffer and numbers them. The numbering does not necessarily start from 0. Instead, TCP chooses an arbitrary number between 0 and $2^{32} - 1$ for the number of the first byte. For example, if the number happens to be 1057 and the total data to be sent is 6000 bytes, the bytes are numbered from 1057 to 7056.

Sequence Number

After the bytes have been numbered, TCP assigns a sequence number to each segment that is being sent. The sequence number, in each direction, is defined as follows:

1- The sequence number of the first segment is the ISN (initial sequence number), which is a random number.

2- The sequence number of any other segment is the sequence number of the previous segment plus the number of bytes carried by the previous segment.

Example: Suppose a TCP connection is transferring a file of 5000 bytes. The first byte is numbered 10001. What are the sequence numbers for each segment if data are sent in five segments, each carrying 1000 bytes?

Solution: The following shows the sequence number for each segment:

Segment 1 → Sequence Number: 10001 Range: 10001 to 11000
Segment 2 → Sequence Number: 11001 Range: 11001 to 12000
Segment 3 → Sequence Number: 12001 Range: 12001 to 13000
Segment 4 → Sequence Number: 13001 Range: 13001 to 14000
Segment 5 → Sequence Number: 14001 Range: 14001 to 15000

Note:

- When a segment carries a combination of data and control information (piggybacking), it uses a sequence number.
- If a segment does not carry user data, it does not logically define a sequence number. The field is there, but the value is not valid. However, some segments, when carrying only control information, need a sequence number to allow an acknowledgment from the receiver. These segments are used for connection establishment, termination, or abortion. Each of these segments consume one sequence number as though it carries one byte, but there are no actual data.

Acknowledgment Number

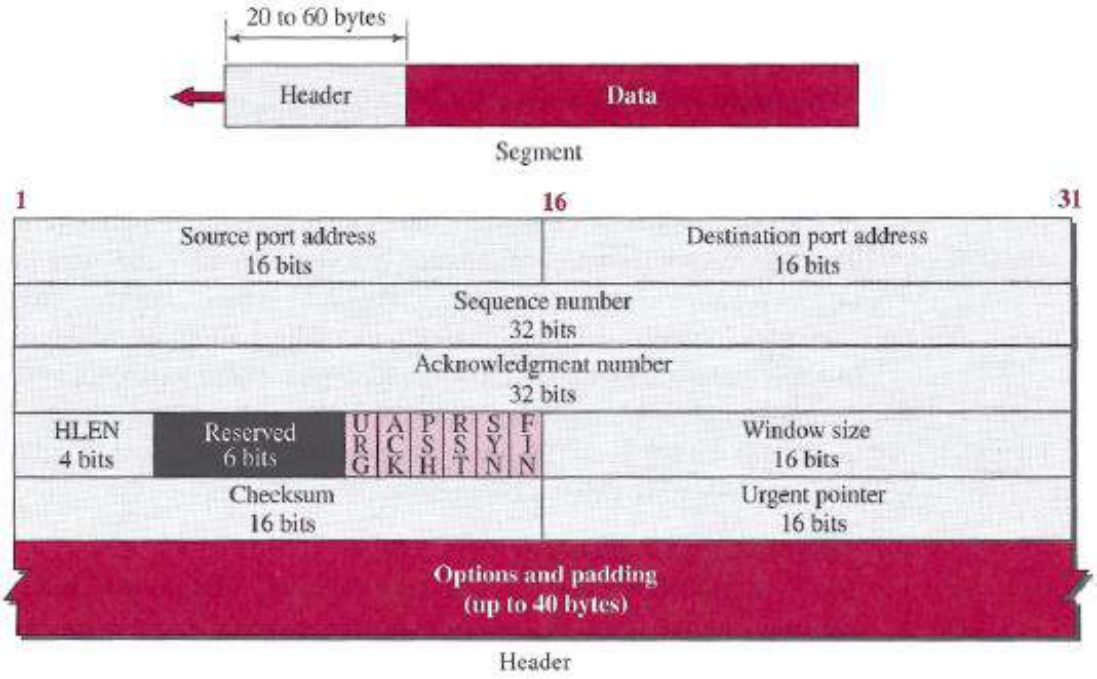
As we discussed previously, communication in TCP is full duplex; when a connection is established, both parties can send and receive data at the same time. Each party numbers the bytes, usually with a different starting byte number. The sequence number in each direction shows the number of the first byte carried by the segment. Each party also uses an acknowledgment number to confirm the bytes it has received. However, the acknowledgment number defines the number of the next byte that the party expects to receive. In addition, the acknowledgment number is cumulative, which means that the party takes the number of the last byte that it has received, safe and sound, adds 1 to it, and announces this sum as the acknowledgment number. The term cumulative here means that if a party uses 5643 as an acknowledgment number, it has received all bytes from the beginning up to 5642. Note that this does not mean that the party has received 5642 bytes, because the first byte number does not have to be 0.

TCP PACKET

A packet in TCP is called a **segment**.

Format

The format of a segment is shown in the figure below. The segment consists of a header of 20 to 60 bytes, followed by data from the application program. The header is 20 bytes if there are no options and up to 60 bytes if it contains options.



1- **Source port address**: This is a 16-bit field that defines the port number of the application program in the host that is sending the segment.

2- **Destination port address**: This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment.

3- **Sequence number**: This 32-bit field defines the number assigned to the first byte of data contained in this segment. As we said before, TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered. The sequence number tells the destination which byte in this sequence is the first byte in the segment. During connection establishment each party uses a random number generator to create an initial sequence number (ISN), which is usually different in each direction.

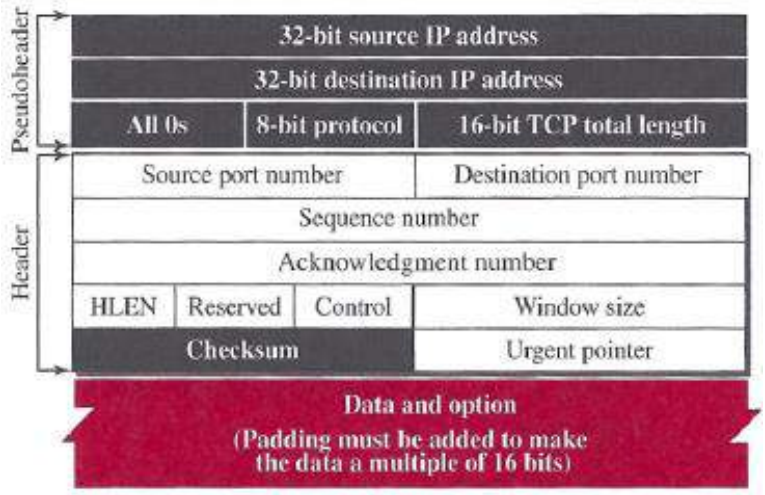
4- **Acknowledgment number**: This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number x from the other party, it returns (x+1) as the acknowledgment number. Acknowledgment and data can be piggybacked together.

5- Header length: This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field is always between 5 (5x4=20) and 15 (15x4=60).

6- Control: This field defines 6 different control bits or flags, as shown in the figure below. One or more of these bits can be set at a time. These bits enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP. A brief description of each bit is shown in the figure below.

7- Window size: This field defines the window size of the sending TCP in bytes. Note that the length of this field is 16 bits, which means that the maximum size of the window is 65,535 bytes. This value is normally referred to as the receiving window (rwnd) and is determined by the receiver. The sender must obey the dictation of the receiver in this case.

8- Checksum. This 16-bit field contains the checksum. The calculation of the checksum for TCP follows the same procedure as the one described for UDP. However, the use of the checksum in the UDP datagram is optional, whereas the use of the checksum for TCP is mandatory. The same pseudo-header, serving the same purpose, is added to the segment. For the TCP pseudo-header, the value for the protocol field is 6. See the figure below.



9- Urgent pointer: This 16-bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data. It defines a value that must be added to the sequence number to obtain the number of the last urgent byte in the data section of the segment.

10- Options: There can be up to 40 bytes of optional information in the TCP header.

TCP CONNECTION

TCP is connection-oriented. As discussed before, a connection-oriented transport protocol establishes a logical path between the source and destination. All of the segments belonging to a message are then sent over this logical path. Using a single logical pathway for the entire message facilitates the acknowledgment process as well as retransmission of damaged or lost frames. You may wonder how TCP, which uses the services of IP, a connectionless protocol, can be connection-oriented. The point is that a TCP connection is logical, not physical. TCP operates at a higher level. TCP uses the services of IP to deliver individual segments to the receiver, but it controls the connection itself. If a segment is lost or corrupted, it is retransmitted. Unlike TCP, IP is unaware of this retransmission. If a segment arrives out of order, TCP holds it until the missing segments arrive; IP is unaware of this reordering.

In TCP, connection-oriented transmission requires three phases: connection establishment, data transfer, and connection termination.

1- Connection Establishment

TCP transmits data in full-duplex mode. When two TCPs in two machines are connected, they are able to send segments to each other simultaneously. This implies that each party must initialize communication and get approval from the other party before any data are transferred.

Three- Way Handshaking

The connection establishment in TCP is called three-way handshaking. In our example, an application program, called the client, wants to make a connection with another application program, called the server, using TCP as the transport-layer protocol. The process starts with the server. The server program tells its TCP that it is ready to accept a connection.

This request is called a passive open. Although the server TCP is ready to accept a connection from any machine in the world, it cannot make the connection itself.

The client program issues a request for an active open. A client that wishes to connect to an open server tells its TCP to connect to a particular server. TCP can now start the three-way handshaking process, as shown in the figure below.

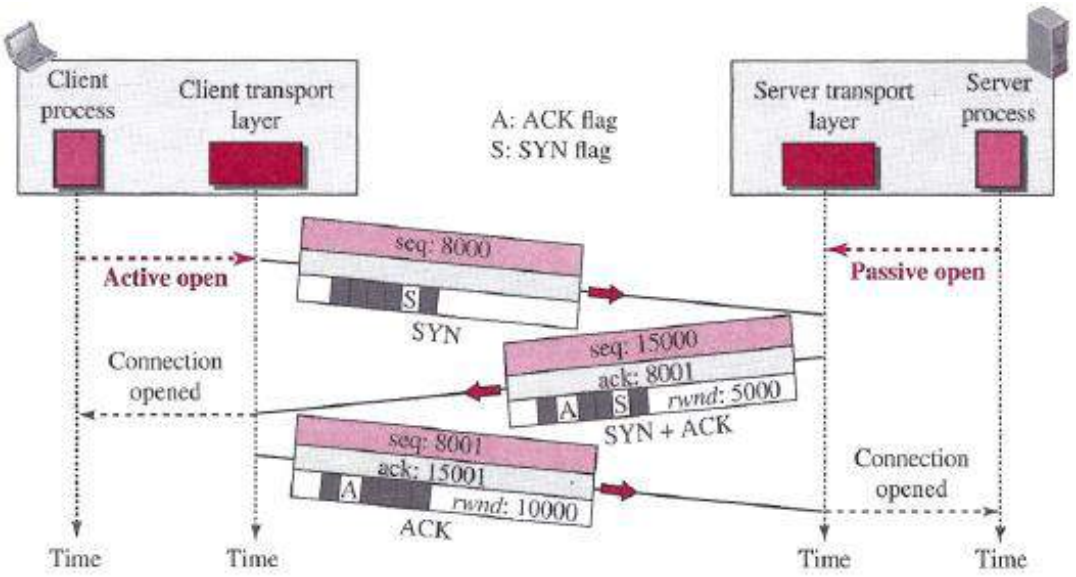
To show the process we use time lines. Each segment has values for all its header fields and perhaps for some of its option fields too. However, we show only the few fields necessary to understand each phase. We show the sequence number, the acknowledgment number, the control flags (only those that are set), and window size if relevant. The three steps in this phase are as follows.

1- The client sends the first segment, a SYN segment, in which only the SYN flag is set. This segment is for synchronization of sequence numbers. The client in our example chooses a random number as the first sequence number and sends this number to the server. This sequence number is called the initial sequence number (ISN). Note that this segment does not contain an acknowledgment

number. It does not define the window size either; a window size definition makes sense only when a segment includes an acknowledgment. Note that the SYN segment is a control segment and carries no data. However, it consumes one sequence number because it needs to be acknowledged. We can say that the SYN segment carries one imaginary byte.

2- The server sends the second segment, a SYN + ACK segment with two flag bits set as: SYN and ACK. This segment has a dual purpose. First, it is a SYN segment for communication in the other direction. The server uses this segment to initialize a sequence number for numbering the bytes sent from the server to the client. The server also acknowledges the receipt of the SYN segment from the client by setting the ACK flag and displaying the next sequence number it expects to receive from the client. Because the segment contains an acknowledgment, it also needs to define the receive window size, rwnd (to be used by the client). Since this segment is playing the role of a SYN segment, it needs to be acknowledged. It, therefore, consumes one sequence number.

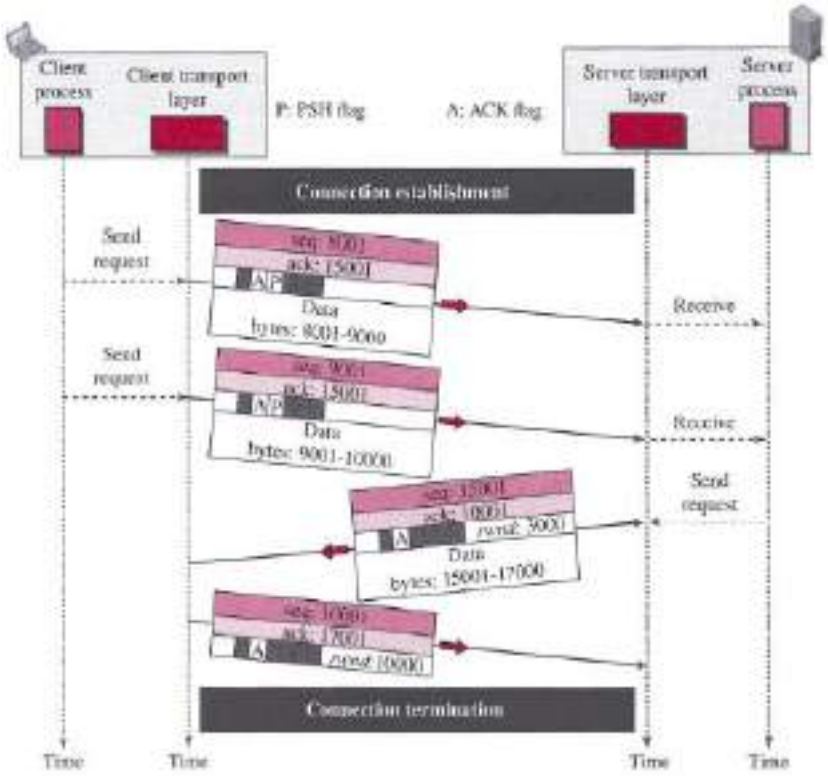
3- The client sends the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field. Note that the ACK segment does not consume any sequence numbers if it does not carry data, but some implementations allow this third segment in the connection phase to carry the first chunk of data from the client. In this case, the segment consumes as many sequence numbers as the number of data bytes.



2- Data Transfer

After connection is established, bidirectional data transfer can take place. The client and server can send data and acknowledgments in both directions. The acknowledgment is piggybacked with the data. The figure below shows an example.

Computer Networks



In this example, after a connection is established, the client sends $2{,}000$ bytes of data in two segments. The server then sends $2{,}000$ bytes in one segment. The client sends one more segment. The first three segments carry both data and acknowledgment, but the last segment carries only an acknowledgment because there is no more data to be sent. Note the values of the sequence and acknowledgment numbers. The data segments sent by the client have the PSH (push) flag set so that the server TCP knows to deliver data to the server process as soon as they are received. The segment from the server, on the other hand, does not set the push flag. Most TCP implementations have the option to set or not to set this flag.

Pushing Data

We saw that the sending TCP uses a buffer to store the stream of data coming from the sending application program. The sending TCP can select the segment size. The receiving TCP also buffers the data when they arrive and delivers them to the application program when the application program is ready or when it is convenient for the receiving TCP. This type of flexibility increases the efficiency of TCP.

However, there are occasions in which the application program has no need for this flexibility. For example, consider an application program that communicates interactively with another application program on the other end. The application program on one site wants to send a chunk of data to the application program at the other site and receive an immediate response. Delayed transmission and delayed delivery of data may not be acceptable by the application program.

TCP can handle such a situation. The application program at the sender can request a push operation. This means that the sending TCP must not wait for the window to be filled. It must create a segment and send it immediately. The sending TCP must also set the push bit (PSH) to let the receiving TCP know that the segment includes data that must be delivered to the receiving application program as soon as possible and not to wait for more data to come. This means to change the byte-oriented TCP to a chunk-oriented TCP, but TCP can choose whether or not to use this feature.

Urgent Data

TCP is a stream-oriented protocol. This means that the data is presented from the application program to TCP as a stream of bytes. Each byte of data has a position in the stream. However, there are occasions in which an application program needs to send urgent bytes, some bytes that need to be treated in a special way by the application at the other end. The solution is to send a segment with the URG bit set. The sending application program tells the sending TCP that the piece of data is urgent. The sending TCP creates a segment and inserts the urgent data at the beginning of the segment. The rest of the segment can contain normal data from the buffer. The urgent pointer field in the header defines the end of the urgent data (the last byte of urgent data). For example, if the segment sequence number is 15000 and the value of the urgent pointer is 200, the first byte of urgent data is the byte 15000 and the last byte is the byte 15200. The rest of the bytes in the segment (if present) are non-urgent.

It is important to mention that TCP's urgent data is neither a priority service nor an out-of-band data service as some people think. Rather, TCP urgent mode is a service by which the application program at the sender side marks some portion of the byte stream as needing special treatment by the application program at the receiver side. The receiving TCP delivers bytes (urgent or non-urgent) to the application program in order, but informs the application program about the beginning and end of urgent data. It is left to the application program to decide what to do with the urgent data.

3- Connection Termination

Either of the two parties involved in exchanging data (client or server) can close the connection, although it is usually initiated by the client. Most implementations today allow two options for connection termination: three-way handshaking and four-way handshaking with a half-close option.

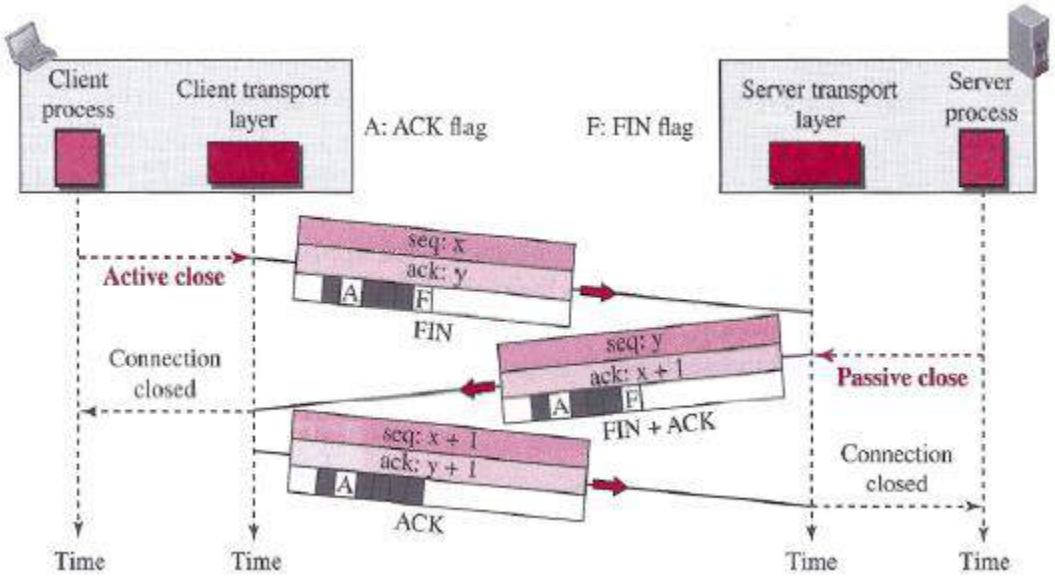
Three- Way Handshaking

Most implementations today allow three-way handshaking for connection termination, as shown in the figure below.

1- In this situation, the client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which the FIN flag is set. Note that a FIN segment can include the last chunk of data sent by the client or it can be just a control segment as shown in the figure, if it is only a control segment, it consumes only one sequence number because it needs to be acknowledged.

2- The server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN + ACK segment, to confirm the receipt of the FIN segment from the client and at the same time to announce the closing of the connection in the other direction. This segment can also contain the last chunk of data from the server. If it does not carry data, it consumes only one sequence number because it needs to be acknowledged.

3- The client TCP sends the last segment, an ACK segment, to confirm the receipt of the FIN segment from the TCP server. This segment contains the acknowledgment number, which is one plus the sequence number received in the FIN segment from the server. This segment cannot carry data and consumes no sequence numbers.

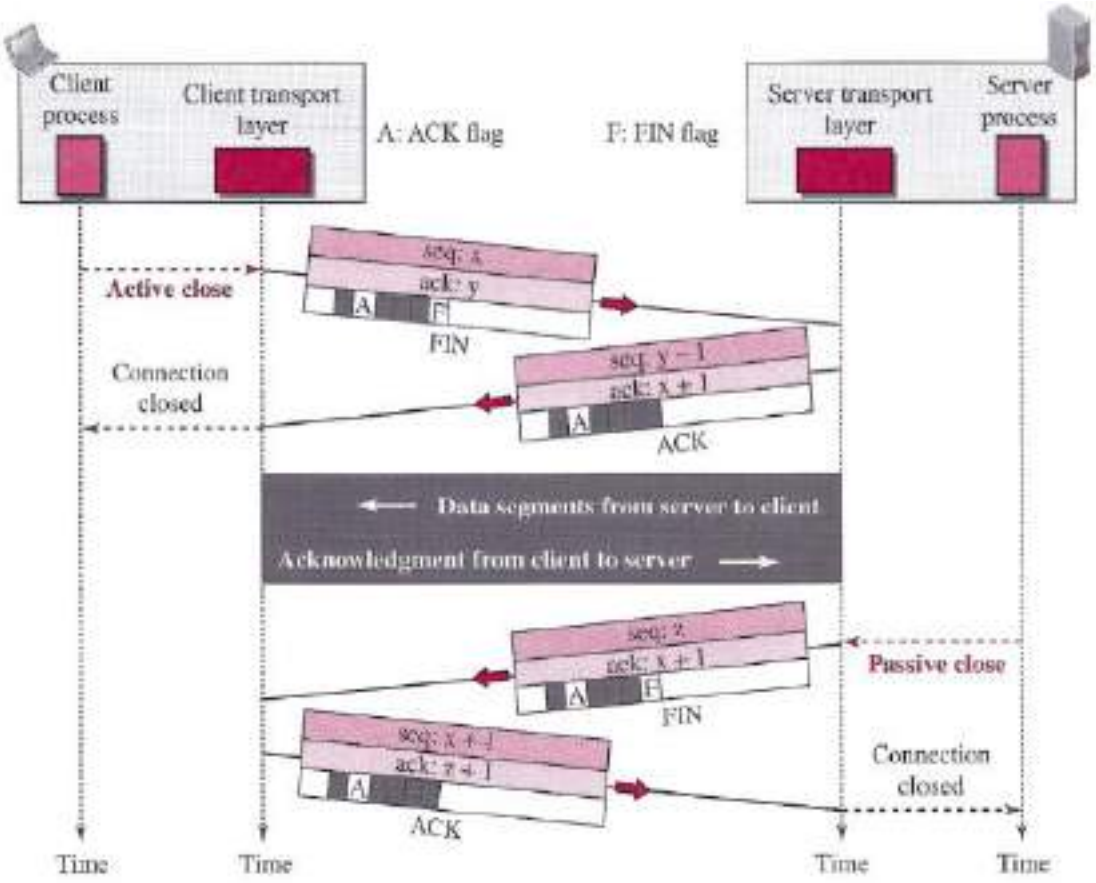


Half-Close

In TCP, one end can stop sending data while still receiving data. This is called a half- close. Either the server or the client can issue a half-close request. It can occur when the server needs all the data before processing can begin. A good example is sorting. When the client sends data to the server to be sorted, the server needs to receive all the data before sorting can start. This means the client, after sending all data, can close the connection in the client-to-server direction. However, the server-to-client direction must remain open to return the sorted data. The server, after receiving the data, still needs time for sorting; its outbound direction must remain open. The figure below shows an example of a half-close.

The data transfer from the client to the server stops. The client half-closes the connection by sending a FIN segment. The server accepts the half-close by sending the ACK segment. The server, however, can still send data. When the server has sent all of the processed data, it sends a FIN segment, which is acknowledged by an ACK from the client.

After half-closing the connection, data can travel from the server to the client and acknowledgments can travel from the client to the server. The client cannot send any more data to the server.



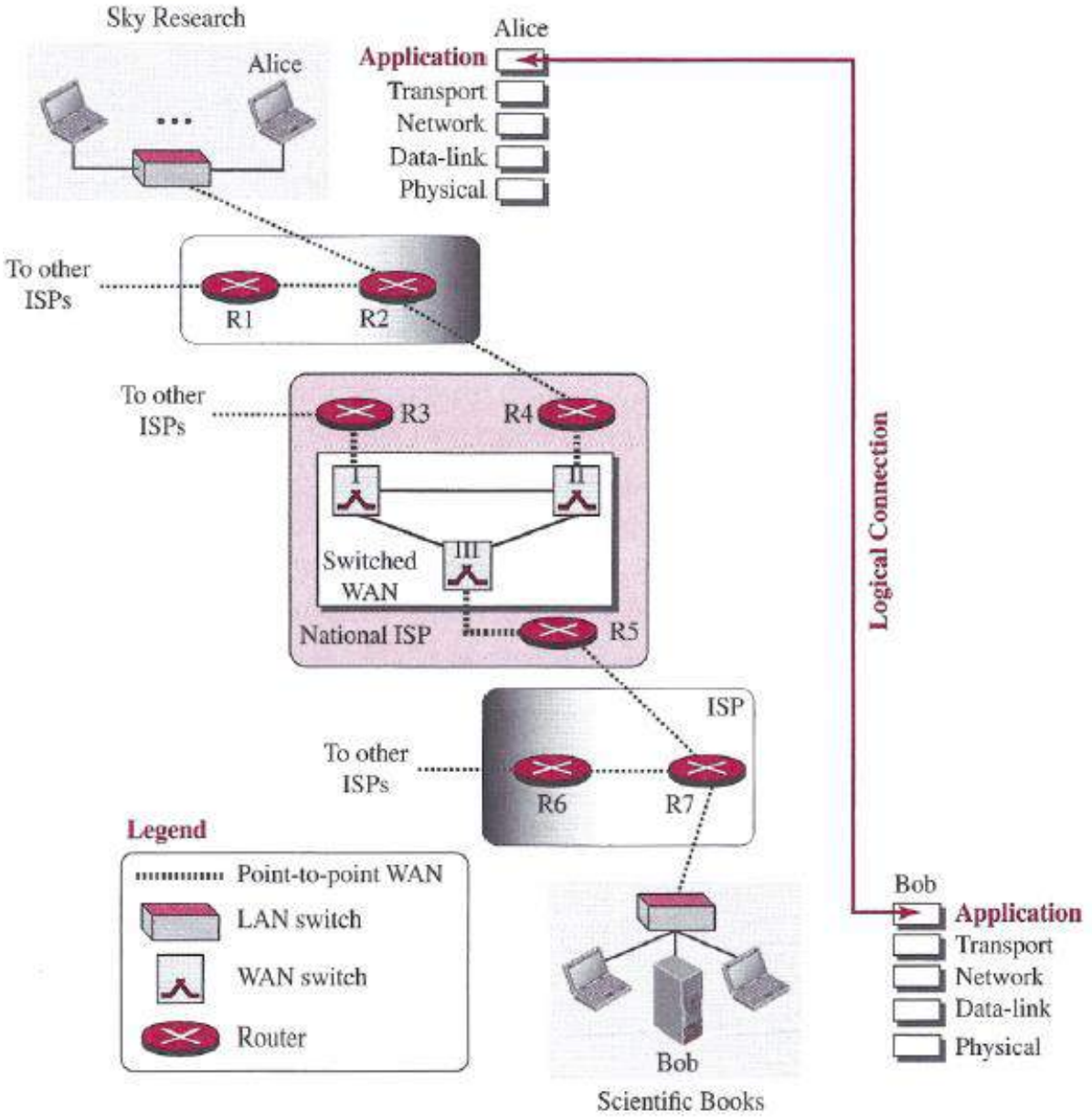
Connection Reset

TCP at one end may deny a connection request, may abort an existing connection, or may terminate an idle connection. All of these are done with the RST (reset) flag.

Computer Networks

THE APPLICATION LAYER

The application layer provides services to the user. Communication is provided using a logical connection, which means that the two application layers assume that there is an imaginary direct connection through which they can send and receive messages. The figure below shows the idea behind this logical connection.



The figure shows the same scenario we have seen for other layers, but this time the logical connection is between two application layers. A scientist working in a research company, Sky Research, needs to order a book related to her research from an online bookseller, Scientific Books. Logical connection takes place between the application layer of a computer at Sky Research and the application layer of a server at Scientific Books. We call the first host Alice and the second one Bob. The communication

at the application layer is logical, not physical. Alice and Bob assume that there is a two-way logical channel between them through which they can send and receive messages. The actual communication, however, takes place through several devices (Alice, R2, R4, R5, R7, and Bob) and several physical channels, as shown in the figure.

PROVIDING SERVICES

All communication networks that started before the Internet were designed to provide services to network users. Most of these networks, however, were originally designed to provide one specific service. For example, the telephone network was originally designed to provide voice service: to allow people all over the world to talk to each other. This network, however, was later used for some other services, such as facsimile (fax), enabled by users adding some extra hardware at both ends.

The Internet was originally designed for the same purpose: to provide service to users around the world. The layered architecture of the TCP/IP protocol suite, however, makes the Internet more flexible than other communication networks such as postal or telephone networks. Each layer in the suite was originally made up of one or more protocols, but new protocols can be added or some protocols can be removed or replaced by the Internet authorities. However, if a protocol is added to each layer, it should be designed in such a way that it uses the services provided by one of the protocols at the lower layer. If a protocol is removed from a layer, care should be taken to change the protocol at the next higher layer that supposedly uses the services of the removed protocol.

The application layer, however, is somewhat different from other layers in that it is the highest layer in the suite. The protocols in this layer do not provide services to any other protocol in the suite; they only receive services from the protocols in the transport layer. This means that protocols can be removed from this layer easily. New protocols can be also added to this layer as long as the new protocols can use the services provided by one of the transport-layer protocols.

Since the application layer is the only layer that provides services to the Internet user, the flexibility of the application layer allows new application protocols to be easily added to the Internet, which has been occurring during the lifetime of the Internet. When the Internet was created, only a few application protocols were available to the users; today we cannot give a number for these protocols because new ones are being added constantly.

STANDARD AND NONSTANDARD PROTOCOLS

To provide smooth operation of the Internet, the protocols used in the first four layers of the TCP/IP suite need to be standardized and documented. They normally become part of the package that is included in operating systems such as Windows or UNIX. To be flexible, however, the application-layer protocols can be both standard and nonstandard.

1- Standard Application-Layer Protocols

There are several application-layer protocols that have been standardized and documented by the Internet authority, and we are using them in our daily interaction with the Internet. Each standard protocol is a pair of computer programs that interact with the user and the transport layer to provide a specific service to the user. In the case of these application protocols, we should know what types of services they provide, how they work, the options that we can use with these applications, and so on. The study of these protocols enables a network manager to easily solve the problems that may occur when using these protocols. The deep understanding of how these protocols work will also give us some ideas about how to create new nonstandard protocols.

2- Nonstandard Application-Layer Protocols

A programmer can create a nonstandard application-layer program if he can write two programs that provide service to the user by interacting with the transport layer. It is the creation of a nonstandard (proprietary) protocol, which does not even need the approval of the Internet authorities if privately used, that has made the Internet so popular worldwide.

A private company can create a new customized application protocol to communicate with all of its offices around the world using the services provided by the first four layers of the TCP/IP protocol suite without using any of the standard application programs. What is needed is to write programs, in one of the computer languages, that use the available services provided by the transport-layer protocols.

APPLICATION-LAYER PARADIGMS

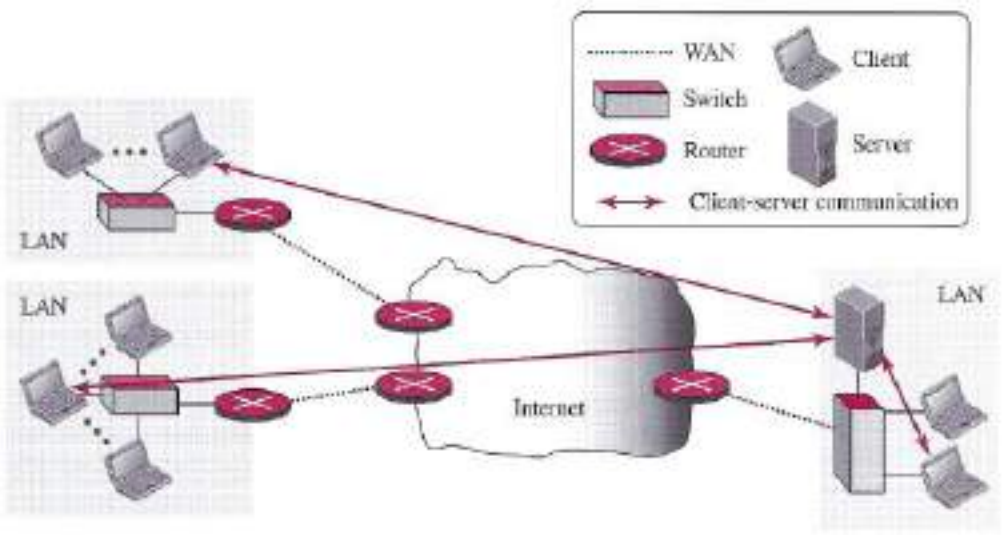
It should be clear that to use the Internet we need two application programs to interact with each other: one running on a computer somewhere in the world, the other running on another computer somewhere else in the world. The two programs need to send messages to each other through the Internet infrastructure. However, what the relationship should be between these programs. Should both application programs be able to request services and provide services, or should the application programs just do one or the other?

Two paradigms have been developed during the lifetime of the Internet: the client-server paradigm and the peer-to-peer paradigm.

1- Traditional Paradigm: Client-Server

The traditional paradigm is called the client-server paradigm. It was the most popular paradigm until a few years ago. In this paradigm, the service provider is an application program, called the server process; it runs continuously, waiting for another application program, called the client process, to make a connection through the Internet and ask for service. There are normally some server processes that can provide a specific type of service, but there are many clients that request service from any of these server processes. The server process must be running all the time; the client process is started when the client needs to receive service.

Although the communication in the client-server paradigm is between two application programs, the role of each program is totally different. In other words, we cannot run a client program as a server program or vice versa. The figure below shows an example of a client-server communication in which three clients communicate with one server to receive the services provided by this server.



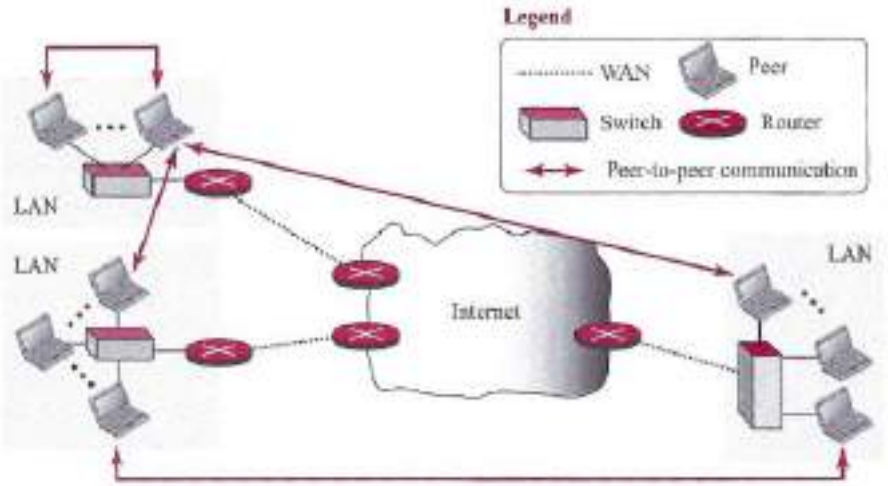
One problem with this paradigm is that the concentration of the communication load is on the shoulder of the server, which means the server should be a powerful computer. Even a powerful computer may become overwhelmed if a large number of clients try to connect to the server at the same time.

Another problem is that there should be a service provider willing to accept the cost and create a powerful server for a specific service, which means the service must always return some type of income for the server in order to encourage such an arrangement.

Several traditional services are still using this paradigm, including the World Wide Web (WWW) and its vehicle Hyper Text Transfer Protocol (HTTP), file transfer protocol (FTP), secure shell (SSH), e-mail, and so on.

2- New Paradigm: Peer-to-Peer

A new paradigm, called the peer-to-peer paradigm (often abbreviated P2P paradigm) has emerged to respond to the needs of some new applications. In this paradigm, there is no need for a server process to be running all the time and waiting for the client processes to connect. The responsibility is shared between peers. A computer connected to the Internet can provide service at one time and receive service at another time. A computer can even provide and receive services at the same time. The figure below shows an example of communication in this paradigm.



One of the areas that really fits in this paradigm is the Internet telephony. Communication by phone is indeed a peer-to-peer activity; no party needs to be running forever waiting for the other party to call. Another area in which the peer-to-peer paradigm can be used is when some computers connected to the Internet have something to share with each other. For example, if an Internet user has a file

available to share with other Internet users, there is no need for the file holder to become a server and run a server process all the time waiting for other users to connect and retrieve the file.

Although the peer-to-peer paradigm has been proved to be easily scalable and cost-effective in eliminating the need for expensive servers to be running and maintained all the time, there are also some challenges. The main challenge has been security; it is more difficult to create secure communication between distributed services than between those controlled by some dedicated servers. The other challenge is applicability; it appears that not all applications can use this new paradigm. For example, not many Internet users are ready to become involved, if one day the Web can be implemented as a peer-to-peer service. There are some new applications, such as BitTorrent, Skype, IPTV, and Internet telephony, that use this paradigm.

3- Mixed Paradigm

An application may choose to use a mixture of the two paradigms by combining the advantages of both. For example, a light-load client-server communication can be used to find the address of the peer that can offer a service. When the address of the peer is found, the actual service can be received from the peer by using the peer-to-peer paradigm.

STANDARD CLIENT-SERVER PROTOCOLS

During the lifetime of the Internet, several client-server application programs have been developed. We do not have to redefine them, but we need to understand what they do. For each application, we also need to know the options available to us. The study of these applications and the ways they provide different services can help us to create customized applications in the future.

1- World Wide Web (abbreviated WWW or Web)

The idea of the Web was first proposed by Tim Berners-Lee in 1989 at CERN, the European Organization for Nuclear Research, to allow several researchers at different locations throughout Europe to access each others' researches. The commercial Web started in the early 1990s.

The Web today is a repository of information in which the documents, called web pages, are distributed all over the world and related documents are linked together. The popularity and growth of the Web can be related to two terms in the above statement: distributed and linked.

Distribution allows the growth of the Web. Each web server in the world can add a new web page to the repository and announce it to all Internet users without overloading a few servers.

Linking allows one web page to refer to another web page stored in another server somewhere else in the world. The linking of web pages was achieved using a concept called hypertext, which was introduced many years before the advent of the Internet. The idea was to use a machine that automatically retrieved another document stored in the system when a link to it appeared in the document.

The Web implemented this idea electronically to allow the linked document to be retrieved when the link was clicked by the user. Today, the term hypertext, coined to mean linked text documents, has been changed to hypermedia, to show that a web page can be a text document, an image, an audio file, or a video file.

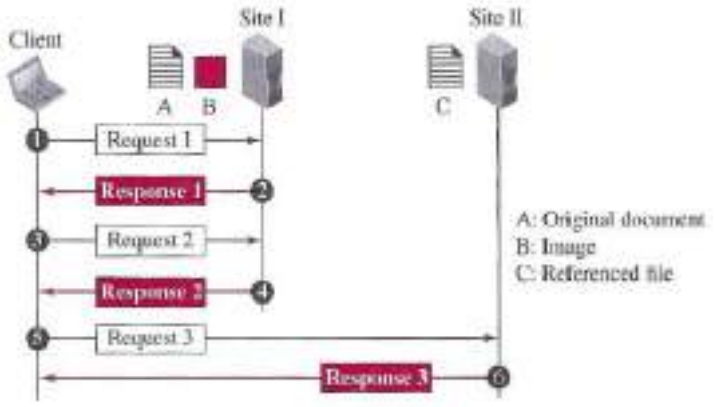
The purpose of the Web has gone beyond the simple retrieving of linked documents. Today, the Web is used to provide electronic shopping and gaming. One can use the Web to listen to radio programs or view television programs whenever one desires without being forced to listen to or view these programs when they are broadcast.

Architecture

The WWW today is a distributed client–server service, in which a client using a browser can access a service using a server. However, the service provided is distributed over many locations called sites. Each site holds one or more web pages. Each web page, however, can contain some links to other web pages in the same or other sites. In other words, a web page can be simple or composite.

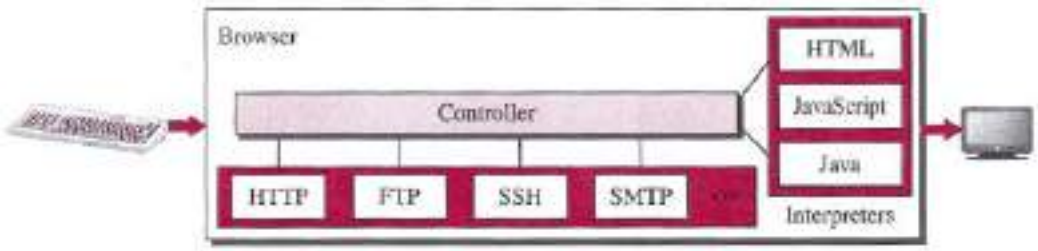
A simple web page has no links to other web pages; a composite web page has one or more links to other web pages. Each web page is a file with a name and address.

Example: Assume we need to retrieve a scientific document that contains one reference to another text file and one reference to a large image. The figure below shows the situation. The main document and the image are stored in two separate files (file A and file B) in the same site; the referenced text file (file C) is stored in another site. Since we are dealing with three different files, we need three transactions if we want to see the whole document. The first transaction (request/response) retrieves a copy of the main document (file A), which has references (pointers) to the second and third files. When a copy of the main document is retrieved and browsed, the user can click on the reference to the image to invoke the second transaction and retrieve a copy of the image (file B). If the user needs to see the contents of the referenced text file, she can click on its reference (pointer) invoking the third transaction and retrieving a copy of file C. Note that although files A and B both are stored in site I, they are independent files with different names and addresses. Two transactions are needed to retrieve them. A very important point we need to remember is that file A, file B, and file C in the example are independent web pages, each with independent names and addresses. Although references to file B or C are included in file A, it does not mean that each of these files cannot be retrieved independently. A second user can retrieve file B with one transaction. A third user can retrieve file C with one transaction.



Web Client (Browser)

A variety of vendors offer commercial browsers that interpret and display a web page, and all of them use nearly the same architecture. Each browser usually consists of three parts: a controller, client protocols, and interpreters. (see The figure below).



The controller receives input from the keyboard or the mouse and uses the client programs to access the document. After the document has been accessed, the controller uses one of the interpreters to display the document on the screen. The client protocol can be one of the protocols described later, such as HTTP or FTP. The interpreter can be HTML, Java, or JavaScript, depending on the type of document. Some commercial browsers include Internet Explorer, Netscape Navigator, and Firefox.

Web Server

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

Uniform Resource Locator (URL)

A web page, as a file, needs to have a unique identifier to distinguish it from other web pages. To define a web page, we need three identifiers: host, port, and path.

However, before defining the web page, we need to tell the browser what client-server application we want to use, which is called the protocol. This means we need four identifiers to define the web page. The first is the type of vehicle to be used to fetch the web page; the last three make up the combination that defines the destination object (web page).

1- **Protocol**: The first identifier is the abbreviation for the client-server program that we need in order to access the web page. Although most of the time the protocol is HTTP (Hyper Text Transfer Protocol) we can also use other protocols such as FTP (File Transfer Protocol).

2- **Host**: The host identifier can be the IP address of the server or the unique name given to the server. IP addresses can be defined in dotted decimal notation such as 64.23.56.17; the name is normally the domain name that uniquely defines the host, such as forouran.com, which we discuss in Domain Name System (DNS) later.

3- **Port**: The port, a 16-bit integer, is normally predefined for the client-server application. For example, if the HTTP protocol is used for accessing the web page, the well-known port number is 80. However, if a different port is used, the number can be explicitly given.

4- **Path**: The path identifies the location and the name of the file in the underlying operating system. The format of this identifier normally depends on the operating system. In UNIX, a path is a set of directory names followed by the file name, all separated by a slash. For example, /top/next/last/myfile is a path that uniquely defines a file named myfile, stored in the directory last, which itself is part of the directory next, which itself is under the directory top. In other words, the path lists the directories from the top to the bottom, followed by the file name.

To combine these four pieces together, the uniform resource locator (URL) has been designed; it uses three different separators between the four pieces as shown below:

protocol:/ /host/path Used most of the time

protocol:/ /host :port/path Used when port number is needed

2- HyperText Transfer Protocol (HTTP)

- The HyperText Transfer Protocol (HTTP) is used to define how the client-server programs can be written to retrieve web pages from the Web.
- An HTTP client sends a request; an HTTP server returns a response.
- The server uses the port number 80; the client uses a temporary port number.
- HTTP uses the services of TCP, which is a connection-oriented and reliable protocol. This means that, before any transaction between the client and the server can take place, a connection needs to be established between them. After the transaction, the connection should be terminated. The client and server, however, do not need to worry about errors in messages exchanged or loss of any message, because the TCP is reliable and will take care of this matter.

Nonpersistent versus Persistent Connections

- The hypertext concept embedded in web page documents may require several requests and responses.
- If the web pages, objects to be retrieved, are located on different servers, we do not have any other choice than to create a new TCP connection for retrieving each object.
- However, if some of the objects are located on the same server, we have two choices:
  - To retrieve each object using a new TCP connection, this method is referred to as a nonpersistent connection
  - To make a TCP connection and retrieve them all, this method is referred to as a persistent connection.

Nonpersistent Connections

In a nonpersistent connection, one TCP connection is made for each request/response. The following lists the steps in this strategy:

1- The client opens a TCP connection and sends a request.

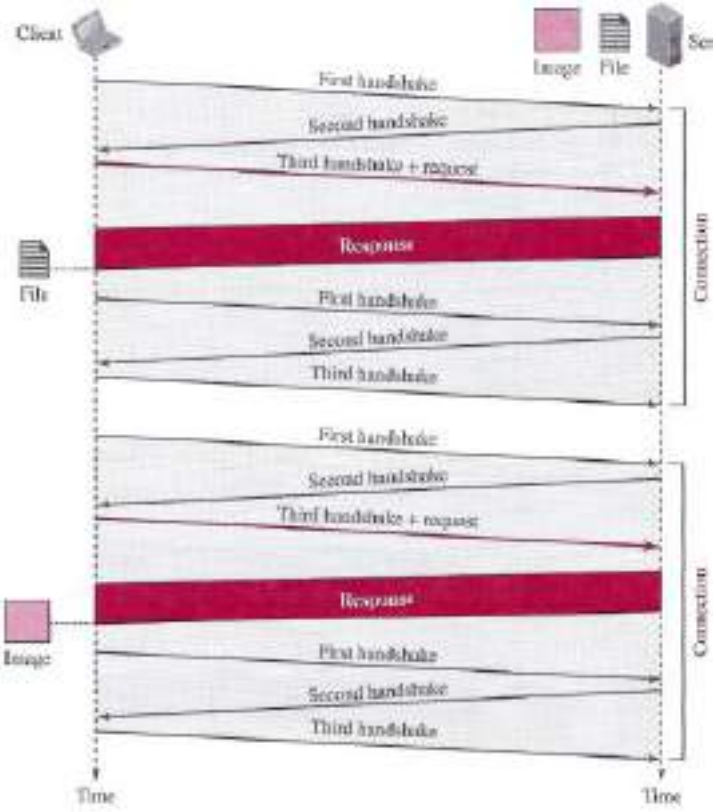
2- The server sends the response and closes the connection.

3- The client reads the data until it encounters an end-of-file marker; it then closes the connection.

- In this strategy, if a file contains links to N different pictures in different files (all located on the same server), the connection must be opened and closed N + 1 times.
- The nonpersistent strategy imposes high overhead on the server because the server needs N + 1 different buffers each time a connection is opened.

The figure below shows an example of a nonpersistent connection.

- The client needs to access a file that contains one link to an image. The text file and image are located on the same server.
- Here we need two connections. For each connection, TCP requires at least three handshake messages to establish the connection, but the request can be sent with the third one. After the connection is established, the object can be transferred. After receiving an object, another three handshake messages are needed to terminate the connection,

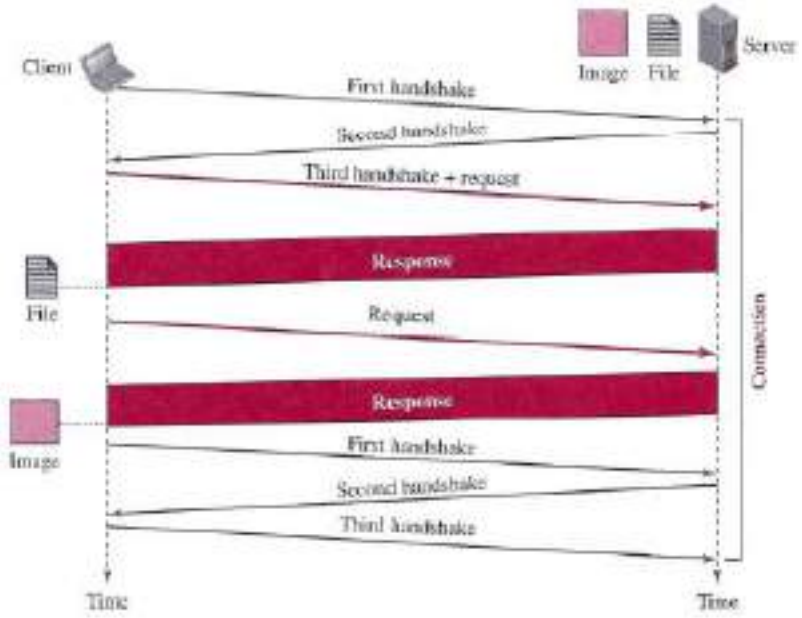


Persistent Connections

In a persistent connection, the server leaves the connection open for more requests after sending a response. The server can close the connection at the request of a client or if a time-out has been reached. The sender usually sends the length of the data with each response. However, there are

some occasions when the sender does not know the length of the data. This is the case when a document is created dynamically or actively. In these cases, the server informs the client that the length is not known and closes the connection after sending the data so the client knows that the end of the data has been reached. Time and resources are saved using persistent connections. Only one set of buffers and variables needs to be set for the connection at each site. The round trip time for connection establishment and connection termination is saved.

The figure below shows the same scenario as in previous figure, but using a persistent connection. Only one connection establishment and connection termination is used, but the request for the image is sent separately.



Cookies

The World Wide Web was originally designed as a stateless entity. A client sends a request; a server responds. Their relationship is over. The original purpose of the Web, retrieving publicly available documents, exactly fits this design. Today the Web has other functions that need to remember some information about the clients; some are listed below:

- Websites are being used as electronic stores that allow users to browse through the store, select wanted items, put them in an electronic cart, and pay at the end with a credit card.
- Some websites need to allow access to registered clients only.
- Some websites are used as portals: the user selects the web pages he wants to see.
- Some websites are just advertising agencies.

For these purposes, the cookie mechanism was devised.

Creating and Storing Cookies

The creation and storing of cookies depend on the implementation; however, the principle is the same.

1- When a server receives a request from a client, it stores information about the client in a file or a string. The information may include the domain name of the client, the contents of the cookie (information the server has gathered about the client such as name, registration number, and so on), a timestamp, and other information depending on the implementation.

2- The server includes the cookie in the response that it sends to the client.

3- When the client receives the response, the browser stores the cookie in the cookie directory, which is sorted by the server domain name.

Using Cookies

When a client sends a request to a server, the browser looks in the cookie directory to see if it can find a cookie sent by that server. If found, the cookie is included in the request. When the server receives the request, it knows that this is an old client, not a new one. Note that the contents of the cookie are never read by the browser or disclosed to the user. It is a cookie made by the server and eaten by the server. Now let us see how a cookie is used for the four previously mentioned purposes:

- An electronic store (e-commerce) can use a cookie for its client shoppers. When a client selects an item and inserts it in a cart, a cookie that contains information about the item, such as its number and unit price, is sent to the browser. If the client selects a second item, the cookie is updated with the new selection information, and so on. When the client finishes shopping and wants to check out, the last cookie is retrieved and the total charge is calculated.
- The site that restricts access to registered clients only sends a cookie to the client when the client registers for the first time. For any repeated access, only those clients that send the appropriate cookie are allowed.
- A web portal uses the cookie in a similar way. When a user selects her favorite pages, a cookie is made and sent. If the site is accessed again, the cookie is sent to the server to show what the client is looking for.
- A cookie is also used by advertising agencies. An advertising agency can place banner ads on some main website that is often visited by users. The advertising agency supplies only a URL that gives the advertising agency's address instead of the banner itself. When a user visits the main website and clicks the icon of a corporation, a request is sent to the advertising agency. The

advertising agency sends the requested banner, but it also includes a cookie with the ID of the user. Any future use of the banners adds to the database that profiles the Web behavior of the user. The advertising agency has compiled the interests of the user and can sell this information to other parties. This use of cookies has made them very controversial. Hopefully, some new regulations will be devised to preserve the privacy of users.

Web Caching: Proxy Servers

HTTP supports proxy servers. A proxy server is a computer that keeps copies of responses to recent requests. The HTTP client sends a request to the proxy server. The proxy server checks its cache. If the response is not stored in the cache, the proxy server sends the request to the corresponding server. Incoming responses are sent to the proxy server and stored for future requests from other clients. The proxy server reduces the load on the original server, decreases traffic, and improves latency. However, to use the proxy server, the client must be configured to access the proxy instead of the target server.

Note that the proxy server acts as both server and client. When it receives a request from a client for which it has a response, it acts as a server and sends the response to the client. When it receives a request from a client for which it does not have a response, it first acts as a client and sends a request to the target server. When the response has been received, it acts again as a server and sends the response to the client.

Proxy Server Location

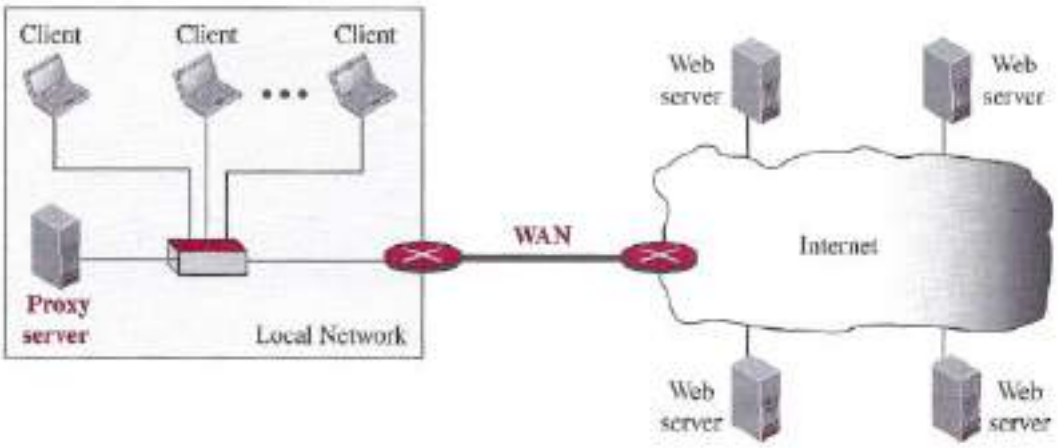
The proxy servers are normally located at the client site. This means that we can have a hierarchy of proxy servers, as shown below:

1- A client computer can also be used as a proxy server, in a small capacity, that stores responses to requests often invoked by the client.

2- In a company, a proxy server may be installed on the computer LAN to reduce the load going out of and coming into the LAN.

3- An ISP with many customers can install a proxy server to reduce the load going out of and coming into the ISP network.

Example: The figure below shows an example of a use of a proxy server in a local network, such as the network on a campus or in a company. The proxy server is installed in the local network. When an HTTP request is created by any of the clients (browsers), the request is first directed to the proxy server. If the proxy server already has the corresponding web page, it sends the response to the client. Otherwise, the proxy server acts as a client and sends the request to the web server in the Internet. When the response is returned, the proxy server makes a copy and stores it in its cache before sending it to the requesting client.



Cache Update

A very important question is how long a response should remain in the proxy server before being deleted and replaced. Several different strategies are used for this purpose. One solution is to store the list of sites whose information remains the same for a while. For example, a news agency may change its news page every morning. This means that a proxy server can get the news early in the morning and keep it until the next day. Another recommendation is to add some headers to show the last modification time of the information. The proxy server can then use the information in this header to guess how long the information would be valid.

HTTP Security

HTTP per se does not provide security. However, HTTP can be run over the Secure Socket Layer (SSL). In this case, HTTP is referred to as HTTPS. HTTPS provides confidentiality, client and server authentication, and data integrity.

3- Electronic Mail

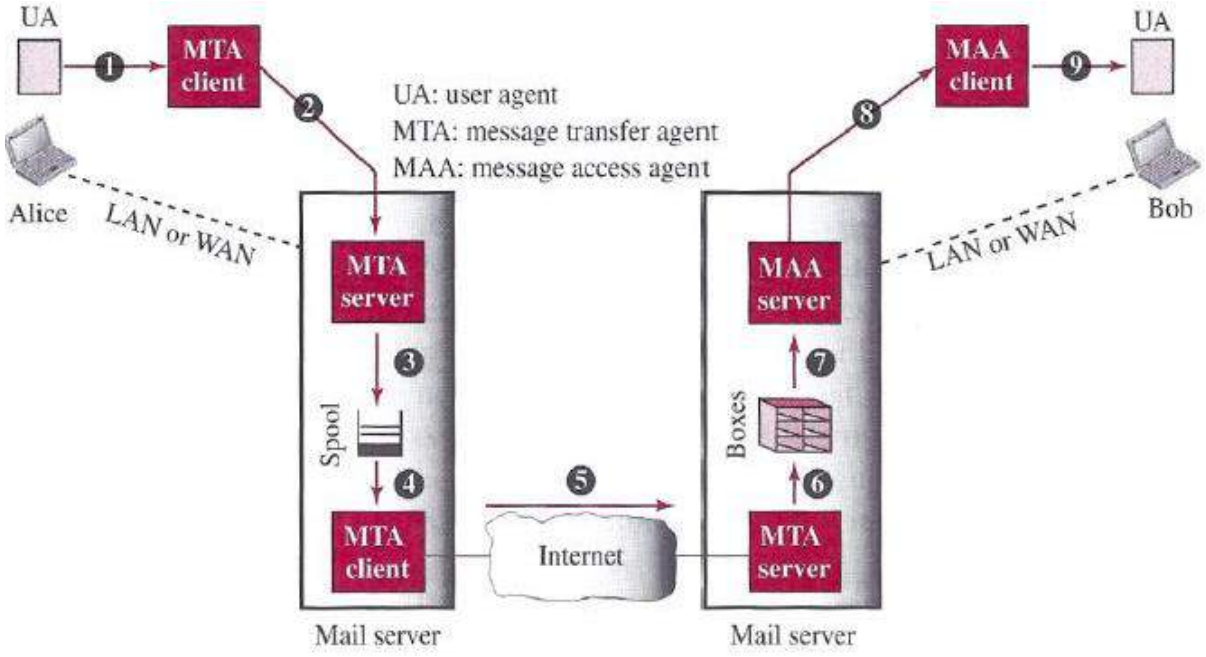
Electronic mail (or e-mail) allows users to exchange messages. The nature of this application, however, is different from other applications discussed so far. In an application such as HTTP or FTP, the server program is running all the time, waiting for a request from a client. When the request arrives, the server provides the service. There is a request and there is a response.

In the case of electronic mail, the situation is different.

- First, e-mail is considered a one-way transaction. When Alice sends an email to Bob, she may expect a response, but this is not a mandate. Bob may or may not respond. If he does respond, it is another one-way transaction.
- Second, it is neither feasible nor logical for Bob to run a server program and wait until someone sends an e-mail to him. Bob may turn off his computer when he is not using it. This means that the idea of client/server programming should be implemented in another way: using some intermediate computers (servers). The users run only client programs when they want and the intermediate servers apply the client/server paradigm, as we discuss in the next section.

Architecture

To explain the architecture of e-mail, we give a common scenario, as shown in the figure below. Another possibility is the case in which Alice or Bob is directly connected to the corresponding mail server, in which LAN or WAN connection is not required, but this variation in the scenario does not affect our discussion.



In the common scenario, the sender and the receiver of the e-mail, Alice and Bob respectively, are connected via a LAN or a WAN to two mail servers. The administrator has created one mailbox for

each user where the received messages are stored. A mailbox is part of a server hard drive, a special file with permission restrictions. Only the owner of the mailbox has access to it. The administrator has also created a queue (spool) to store messages waiting to be sent.

A simple e-mail from Alice to Bob takes nine different steps, as shown in the figure. Alice and Bob use three different agents: a user agent (UA), a message transfer agent (MTA), and a message access agent (MAA). When Alice needs to send a message to Bob, she runs a UA program to prepare the message and send it to her mail server. The mail server at her site uses a queue (spool) to store messages waiting to be sent. The message, however, needs to be sent through the Internet from Alice's site to Bob's site using an MTA. Here two message transfer agents are needed: one client and one server. Like most client-server programs on the Internet, the server needs to run all the time because it does not know when a client will ask for a connection. The client, on the other hand, can be triggered by the system when there is a message in the queue to be sent. The user agent at the Bob site allows Bob to read the received message. Bob later uses an MAA client to retrieve the message from an MAA server running on the second server.

There are two important points we need to emphasize here:

- First, Bob cannot bypass the mail server and use the MTA server directly. To use the MTA server directly, Bob would need to run the MTA server all the time because he does not know when a message will arrive. This implies that Bob must keep his computer on all the time if he is connected to his system through a LAN. If he is connected through a WAN, he must keep the connection up all the time. Neither of these situations is feasible today.
- Second, note that Bob needs another pair of client-server programs: message access programs. This is because an MTA client-server program is a push program: the client pushes the message to the server. Bob needs a pull program. The client needs to pull the message from the server.

User Agent

The first component of an electronic mail system is the user agent (UA). It provides service to the user to make the process of sending and receiving a message easier. A user agent is a software package (program) that composes, reads, replies to, and forwards messages. It also handles local mailboxes on the user computers. There are two types of user agents: command-driven and GUI-based.

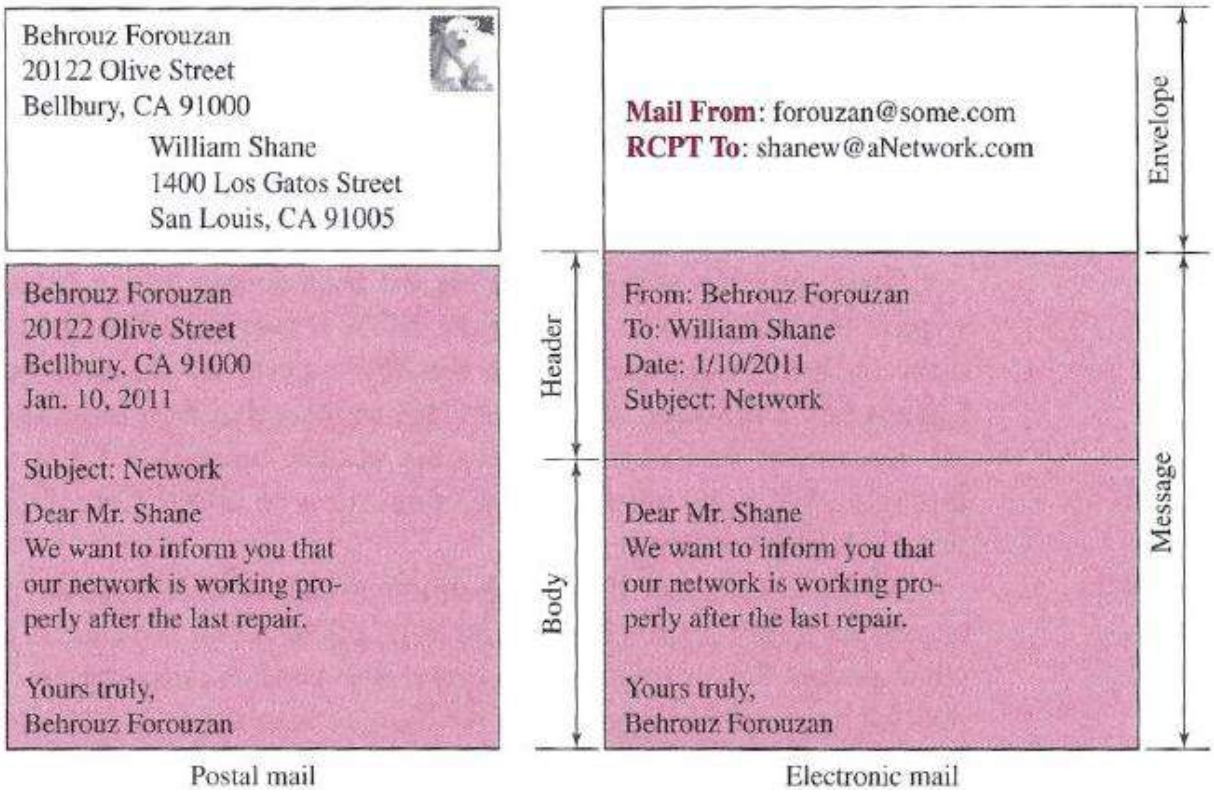
- Command-driven user agents belong to the early days of electronic mail. They are still present as the underlying user agents. A command-driven user agent normally accepts a one-character command from the keyboard to perform its task. For example, a user can type the character r, at the command prompt, to reply to the sender of the message, or type the character R to reply to the sender and all recipients. Some examples of command-driven user agents are mail, pine, and elm.

- Modern user agents are GUI-based. They contain graphical user interface (GUI) components that allow the user to interact with the software by using both the keyboard and the mouse. They have graphical components such as icons, menu bars, and windows that make the services easy to access. Some examples of GUI-based user agents are Eudora and Outlook.

- **Sending Mail**

To send mail, the user, through the UA, creates mail that looks very similar to postal mail. It has an envelope and a message (see the figure below).

- The envelope usually contains the sender address, the receiver address, and other information.
- The message contains the header and the body.
  - The header of the message defines the sender, the receiver, the subject of the message, and some other information.
  - The body of the message contains the actual information to be read by the recipient.



Postal mail Electronic mail

- **Receiving Mail**

The user agent is triggered by the user (or a timer). If a user has mail, the UA informs the user with a notice. If the user is ready to read the mail, a list is displayed in which each line contains a summary of the information about a particular message in the mailbox. The summary usually includes the sender mail address, the subject, and the time the mail was sent or received. The user can select any of the messages and display its contents on the screen.

– **Addresses**

To deliver mail, a mail handling system must use an addressing system with unique addresses. In the Internet, the address consists of two parts: a local part and a domain name, separated by an @ sign (see the figure below).

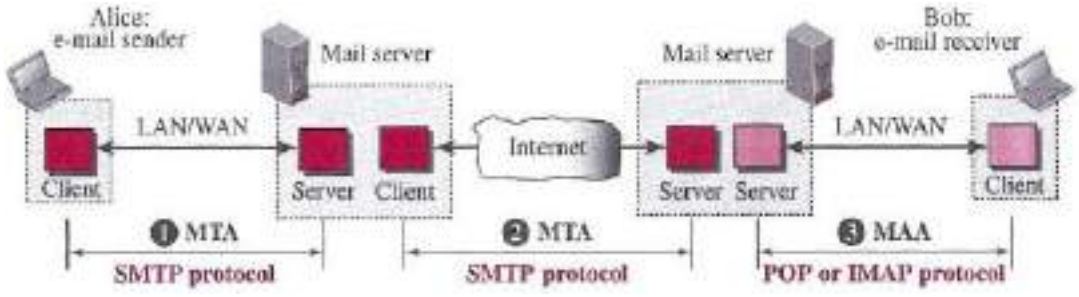
– The local part defines the name of a special file, called the user mailbox, where all the mail received for a user is stored for retrieval by the message access agent.
– The second part of the address is the domain name. An organization usually selects one or more hosts to receive and send e-mail; they are sometimes called mail servers or exchangers. The domain name assigned to each mail exchanger either comes from the DNS database or is a logical name (for example, the name of the organization).

– **Mailing List or Group List**

Electronic mail allows one name, an alias, to represent several different e-mail addresses; this is called a mailing list. Every time a message is to be sent, the system checks the recipient's name against the alias database; if there is a mailing list for the defined alias, separate messages, one for each entry in the list, must be prepared and handed to the MTA.

Message Transfer Agent: SMTP

Based on the common scenario, we can say that the e-mail is one of those applications that needs three uses of client-server paradigms to accomplish its task. It is important that we distinguish these three when we are dealing with e-mail. The figure below shows these three client-server applications. We refer to the first and the second as Message Transfer Agents (MTAs), the third as Message Access Agent (MAA).



The formal protocol that defines the MTA client and server in the Internet is called Simple Mail Transfer Protocol (SMTP). SMTP is used two times, between the sender and the sender's mail server and between the two mail servers. As we will see shortly, another protocol is needed between the mail server and the receiver. SMTP simply defines how commands and responses must be sent back and forth.

Commands and Responses

SMTP uses commands and responses to transfer messages between an MTA client and an MTA server. The command is from an MTA client to an MTA server; the response is from an MTA server to the MTA client. Each command or reply is terminated by a two character (carriage return and line feed) end-of-line token.

– **Commands**

Commands are sent from the client to the server. The format of a command is shown below:

Keyword: argument(s)

It consists of a keyword followed by zero or more arguments. SMTP defines 14 commands, listed in the table below.

Keyword	*Argument(s)*	*Keyword*	*Argument(s)*
HELO	Sender's host name	NOOP	
MAIL FROM	Sender of the message	TURN	
RCPT TO	Intended recipient	EXPN	Mailing list
DATA	Body of the mail	HELP	Command name
QUIT		SEND FROM	Intended recipient
RSET		SMOL FROM	Intended recipient
VRFY	Name of recipient	SMAL FROM	Intended recipient

HELO: This command is used by the client to identify itself. The argument is the domain name of the client host. The format is:

HELO: challenger.atc.fhda.edu

MAIL FROM: This command is used by the client to identify the sender of the message. The argument is the e-mail address of the sender (local part plus the domain name). The format is:

MAIL FROM: forouzan@challenger.atc.fhda.edu

RCPT TO: This command is used by the client to identify the intended recipient of the message. The argument is the e-mail address of the recipient. If there are multiple recipients, the command is repeated. The format is:

RCPT TO: betsy@mcgraw-hill.com

DATA: This command is used to send the actual message. All lines that follow the DATA command are treated as the mail message. The message is terminated by a line containing just one period. The format is:

DATA
This is the message
to be sent to the McGraw-Hill
Company.
.

QUIT: This command terminates the message. The format is:

QUIT

RSET: This command aborts the current mail transaction. The stored information about the sender and recipient is deleted. The connection will be reset:

RSET

VRFY: This command is used to verify the address of the recipient, which is sent as the argument. The sender can ask the receiver to confirm that a name identifies a valid recipient. Its format is:

VRFY: betsy@mcgraw-hill.com

NOOP: This command is used by the client to check the status of the recipient. It requires an answer from the recipient. Its format is:

NOOP

TURN: This command lets the sender and the recipient switch positions, whereby the sender becomes the recipient and vice versa. However, most SMTP implementations today do not support this feature. The format is:

TURN

EXPN: This command asks the receiving host to expand the mailing list sent as the arguments and to return the mailbox addresses of the recipients that comprise the list. The format is:

EXPN: x y z

HELP: This command asks the recipient to send information about the command sent as the argument. The format is:

HELP: mail

SEND FROM: This command specifies that the mail is to be delivered to the terminal of the recipient, and not the mailbox. If the recipient is not logged in, the mail is bounced back. The format is:

SEND FROM: forouzan@fhda.atc.edu

SMOL FROM: This command specifies that the mail is to be delivered to the terminal or the mailbox of the recipient. This means that if the recipient is logged in, the mail is delivered only to the terminal. If the recipient is not logged in, the mail is delivered to the mailbox. The format is:

SMOL FROM: forouzan@fhda.atc.edu

SMAL FROM: This command specifies that the mail is to be delivered to the terminal and the mailbox of the recipient. This means that if the recipient is logged in, the mail is delivered to the terminal and the mailbox. If the recipient is not logged in, the mail is delivered only to the mailbox. The format is:

SMAL FROM: forouzan@fhda.atc.edu

– **Responses**

Responses are sent from the server to the client. A response is a three-digit code that may be followed by additional textual information. The table below shows then most common response types.

Code	*Description*
	Positive Completion Reply
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
	Positive Intermediate Reply
354	Start mail input
	Transient Negative Completion Reply
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted; insufficient storage
	Permanent Negative Completion Reply
500	Syntax error; unrecognized command
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

Mail Transfer Phases

The process of transferring a mail message occurs in three phases: connection establishment, mail transfer, and connection termination.

- **Connection Establishment**

After a client has made a TCP connection to the well-known port 25, the SMTP server starts the connection phase. This phase involves the following three steps, which are illustrated in the figure below.

1- The server sends code 220 (service ready) to tell the client that it is ready to receive mail. If the server is not ready, it sends code 421 (service not available).

2- The client sends the HELO message to identify itself using its domain name address. This step is necessary to inform the server of the domain name of the client. Remember that during TCP connection establishment, the sender and receiver know each other through their IP addresses.

3- The server responds with code 250 (request command completed) or some other code depending on the situation.

- **Message Transfer**

After connection has been established between the SMTP client and server, a single message between a sender and one or more recipients can be exchanged. This phase involves eight steps. Steps 3 and 4 are repeated if there is more than one recipient (see the figure below).

1- The client sends the MAIL FROM message to introduce the sender of the message. It includes the mail address of the sender (mailbox and the domain name). This step is needed to give the server the return mail address for returning errors and reporting messages.

2- The server responds with code 250 or some other appropriate code.

3- The client sends the RCPT TO (recipient) message, which includes the mail address of the recipient.

4- The server responds with code 250 or some other appropriate code.

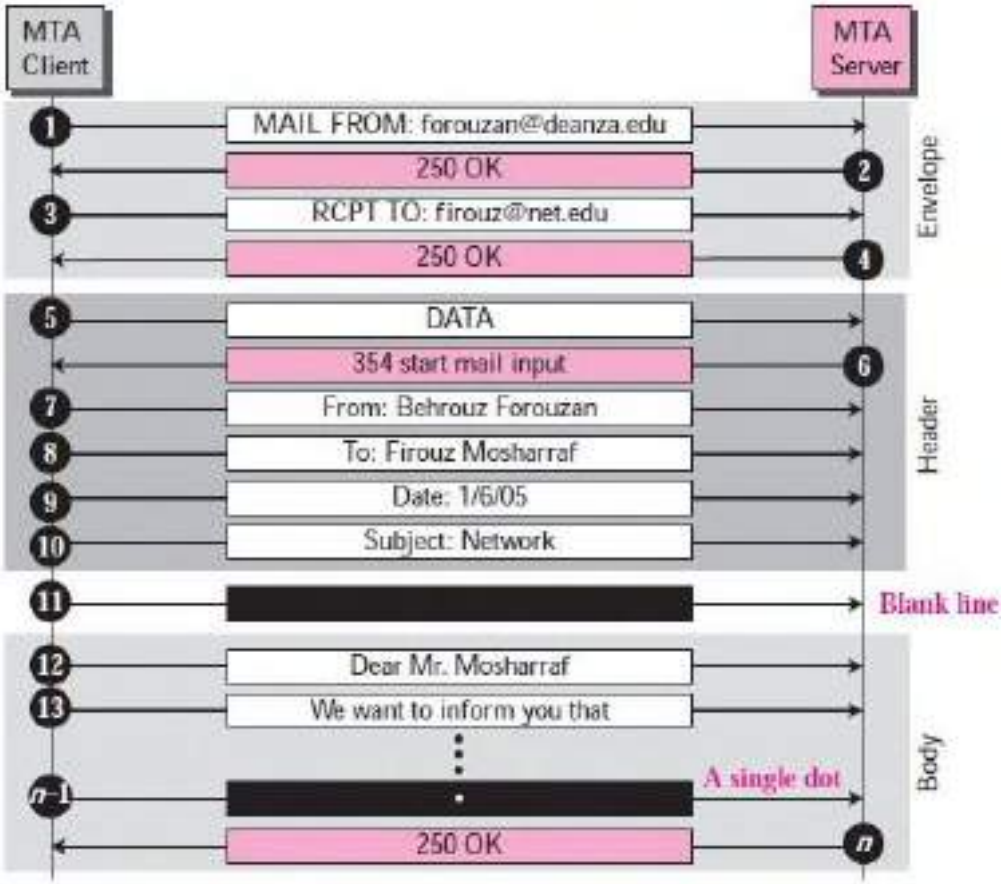
5- The client sends the DATA message to initialize the message transfer.

6- The server responds with code 354 (start mail input) or some other appropriate message.

Computer Networks

7- The client sends the contents of the message in consecutive lines. Each line is terminated by a two-character end-of-line token (carriage return and line feed). The message is terminated by a line containing just one period.

8- The server responds with code 250 (OK) or some other appropriate code.



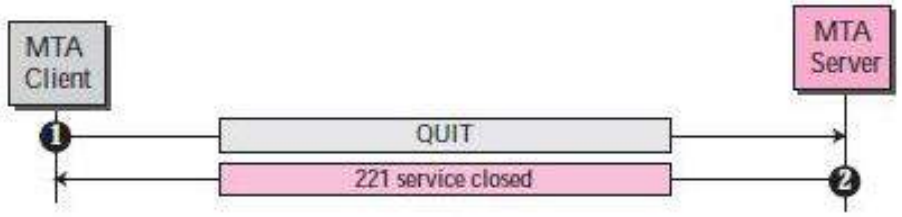
– **Connection Termination**

After the message is transferred successfully, the client terminates the connection. This phase involves two steps (see the figure below).

1- The client sends the QUIT command.

2- The server responds with code 221 or some other appropriate code.

After the connection termination phase, the TCP connection must be closed.



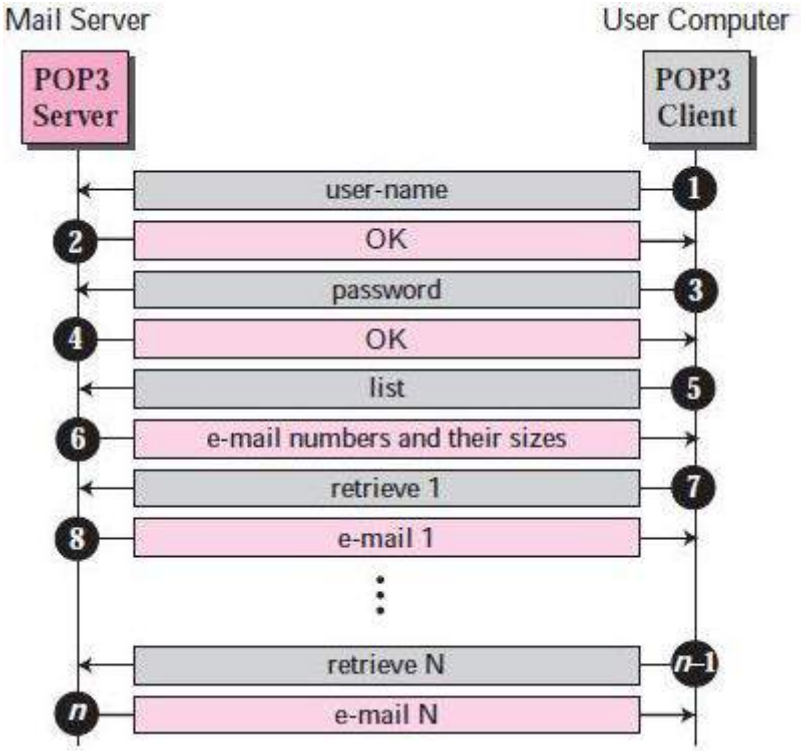
Message Access Agent: POP and IMAP

The first and second stages of mail delivery use SMTP. However, SMTP is not involved in the third stage because SMTP is a push protocol; it pushes the message from the client to the server. In other words, the direction of the bulk data (messages) is from the client to the server. On the other hand, the third stage needs a pull protocol; the client must pull messages from the server. The direction of the bulk data is from the server to the client. The third stage uses a message access agent. Currently two message access protocols are available: Post Office Protocol, version 3 (POP3) and Internet Mail Access Protocol, version 4 (IMAP4).

POP3

Post Office Protocol, version 3 (POP3) is simple but limited in functionality. The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.

Mail access starts with the client when the user needs to download its e-mail from the mailbox on the mail server. The client opens a connection to the server on TCP port 110. It then sends its user name and password to access the mailbox. The user can then list and retrieve the mail messages, one by one. The figure below shows an example of downloading using POP3. Unlike other figures in this chapter, we have put the client on the right hand side because the e-mail receiver (Bob) is running the client process to pull messages from the remote mail server.



POP3 has two modes: the delete mode and the keep mode. In the delete mode, the mail is deleted from the mailbox after each retrieval. In the keep mode, the mail remains in the mailbox after retrieval.

The delete mode is normally used when the user is working at his permanent computer and can save and organize the received mail after reading or replying. The keep mode is normally used when the user accesses his mail away from his primary computer (e.g., a laptop). The mail is read but kept in the system for later retrieval and organizing.

IMAP4

Another mail access protocol is Internet Mail Access Protocol, version 4 (IMAP4). IMAP4 is similar to POP3, but it has more features; IMAP4 is more powerful and more complex. POP3 is deficient in several ways. It does not allow the user to organize his mail on the server; the user cannot have different folders on the server. In addition, POP3 does not allow the user to partially check the contents of the mail before downloading.

IMAP4 provides the following extra functions:

1- A user can check the e-mail header prior to downloading.

2- A user can search the contents of the e-mail for a specific string of characters prior to downloading.

3- A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.

4- A user can create, delete, or rename mailboxes on the mail server.

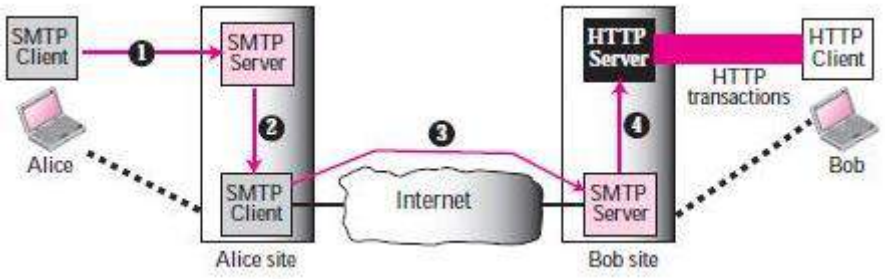
5- A user can create a hierarchy of mailboxes in a folder for e-mail storage.

Web-Based Mail

E-mail is such a common application that some websites today provide this service to anyone who accesses the site. Three common sites are Hotmail, Yahoo, and Google mail. The idea is very simple. There are two cases:

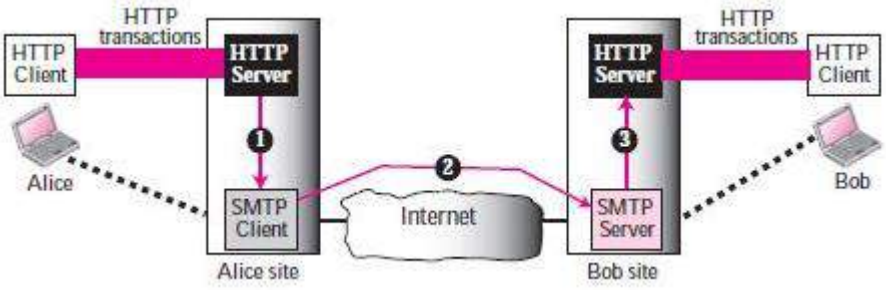
Case I

In the first case, Alice, the sender, uses a traditional mail server; Bob, the receiver, has an account on a web-based server. Mail transfer from Alice's browser to her mail server is done through SMTP. The transfer of the message from the sending mail server to the receiving mail server is still through SMTP. However, the message from the receiving server (the web server) to Bob's browser is done through HTTP. In other words, instead of using POP3 or IMAP4, HTTP is normally used. When Bob needs to retrieve his e-mails, he sends a request HTTP message to the website (Hotmail, for example). The website sends a form to be filled in by Bob, which includes the log-in name and the password. If the log-in name and password match, the list of e-mails is transferred from the web server to Bob's browser in HTML format. Now Bob can browse through his received e-mails and then, using more HTTP transactions, can get his e-mails one by one. This is shown in the figure below.



Case II

In the second case, both Alice and Bob use Web servers, but not necessarily the same server. Alice sends the message to the Web server using HTTP transactions. Alice sends an HTTP request message to her Web server using the name and address of Bob's mailbox as the URL. The server at the Alice site passes the message to the SMTP client and sends it to the server at the Bob site using SMTP protocol. Bob receives the message using HTTP transactions. However, the message from the server at the Alice site to the server at the Bob site still takes place using SMTP protocol. The figure below shows the idea.



E-MAIL SECURITY

The protocol discussed in this chapter does not provide any security provisions per se. However, e-mail exchanges can be secured using two application-layer securities designed in particular for e-mail systems. Two of these protocols, Pretty Good Privacy (PGP) and Secure MIME (SMIME).

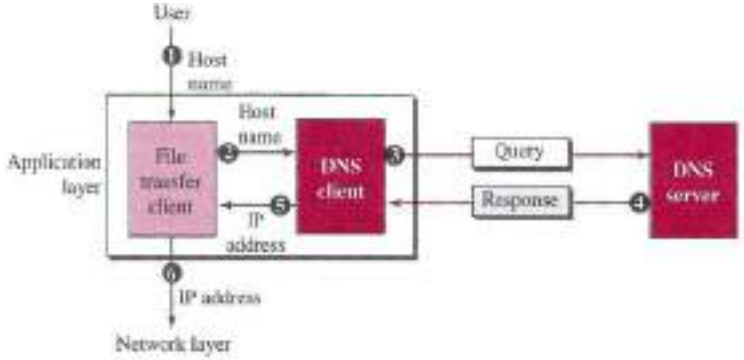
4- Domain Name System (DNS)

The last client-server application program we discuss has been designed to help other application programs. To identify an entity, TCP/IP protocols use the IP address, which uniquely identifies the connection of a host to the Internet. However, people prefer to use names instead of numeric addresses. Therefore, the Internet needs to have a directory system that can map a name to an address. This is analogous to the telephone network. A telephone network is designed to use telephone numbers, not names. People can either keep a private file to map a name to the corresponding telephone number or can call the telephone directory to do so. We discuss how this directory system in the Internet can map names to IP addresses.

Since the Internet is so huge today, a central directory system cannot hold all the mapping. In addition, if the central computer fails, the whole communication network will collapse. A better solution is to distribute the information among many computers in the world. In this method, the host that needs mapping can contact the closest computer holding the needed information. This method is used by the Domain Name System (DNS). We first discuss the concepts and ideas behind the DNS. We then describe the DNS protocol itself.

The figure below shows how TCP/IP uses a DNS client and a DNS server to map a name to an address. A user wants to use a file transfer client to access the corresponding file transfer server running on a remote host. The user knows only the file transfer server name, such as afilesource.com. However, the TCP/IP suite needs the IP address of the file transfer server to make the connection. The following six steps map the host name to an IP address:

1- The user passes the host name to the file transfer client.
2- The file transfer client passes the host name to the DNS client.
3- Each computer, after being booted, knows the address of one DNS server. The DNS client sends a message to a DNS server with a query that gives the file transfer server name using the known IP address of the DNS server.
4- The DNS server responds with the IP address of the desired file transfer server.
5- The DNS server passes the IP address to the file transfer client.
6- The file transfer client now uses the received IP address to access the file transfer server.



Note that the purpose of accessing the Internet is to make a connection between the file transfer client and server, but before this can happen, another connection needs to be made between the DNS client and DNS server. In other words, we need at least two connections in this case. The first is for mapping the name to an IP address; the second is for transferring files. We will see later that the mapping may need more than one connection.

Name Space

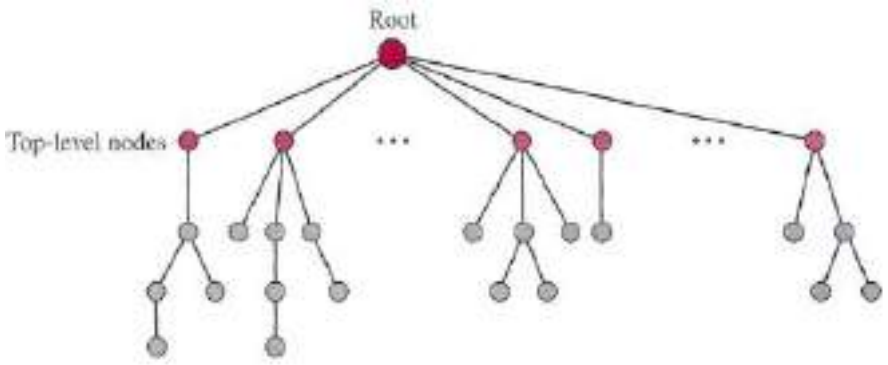
To be unambiguous, the names assigned to machines must be carefully selected from a name space with complete control over the binding between the names and IP addresses. In other words, the names must be unique because the addresses are unique. A name space that maps each address to a unique name can be organized in two ways: flat or hierarchical.

- **In a flat name space**, a name is assigned to an address. A name in this space is a sequence of characters without structure. The names may or may not have a common section; if they do, it has no meaning. The main disadvantage of a flat name space is that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.
- **In a hierarchical name space**, each name is made of several parts. The first part can define the nature of the organization, the second part can define the name of an organization, the third part can define departments in the organization, and so on. In this case, the authority to assign and control the name spaces can be decentralized. A central authority can assign the part of the name that defines the nature of the organization and the name of the organization. The responsibility for the rest of the name can be given to the organization itself. The organization can add suffixes (or prefixes) to the name to define its host or resources. The management of the organization need not worry that the prefix chosen for a host is taken by another organization because, even if part of an address is the same, the whole address is different. For example, assume two organizations call one of their computers caesar. The first organization is given a name by the central authority, such as first. com, the second organization is given the name second. com. When each of these organizations adds the name caesar to the name they have already been given, the end result is two distinguishable names: ceasar.first.com and ceasar. second. com. The names are unique.

Domain Name Space

To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127 (see the figure below).

Computer Networks

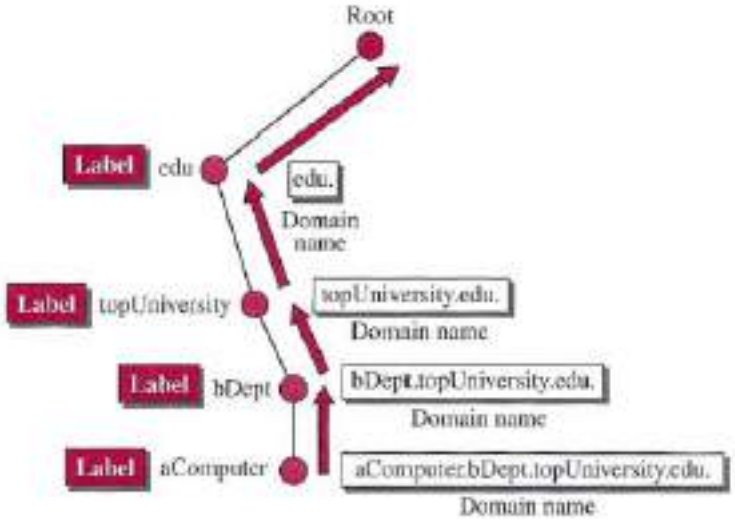


- **Label**

Each node in the tree has a label, which is a string with a maximum of 63 characters. The root label is a null string (empty string). DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

- **Domain Name**

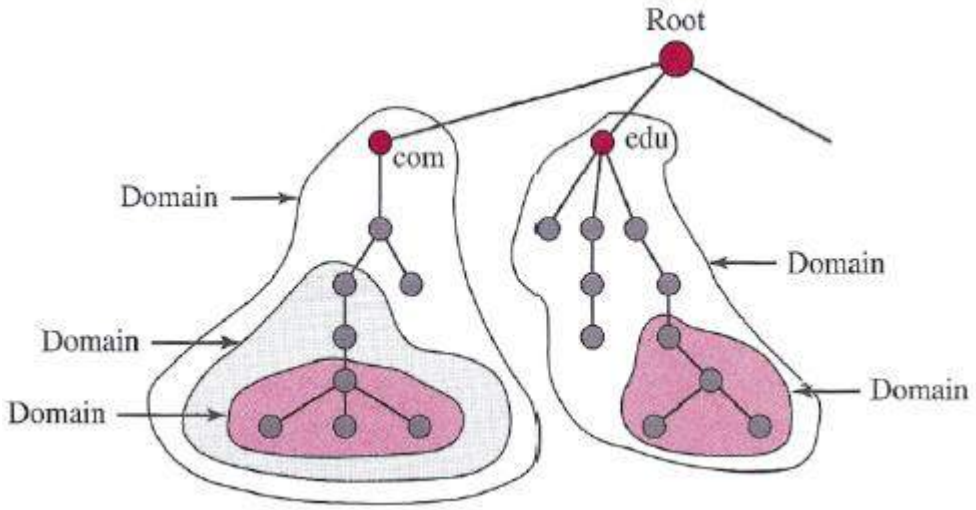
Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.). The domain names are always read from the node up to the root. The last label is the label of the root (null). This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing. The figure below shows some domain names.



- If a label is terminated by a null string, it is called **a fully qualified domain name** (**FQDN**). The name must end with a null label, but because null means nothing, the label ends with a dot.
- If a label is not terminated by a null string, it is called **a partially qualified domain name** (**PQDN**). A PQDN starts from a node, but it does not reach the root. It is used when the name to be resolved belongs to the same site as the client. Here the resolver can supply the missing part, called the suffix, to create an FQDN.

Domain

A domain is a subtree of the domain name space. The name of the domain is the name of the node at the top of the subtree. The figure below shows some domains. Note that a domain may itself be divided into domains.

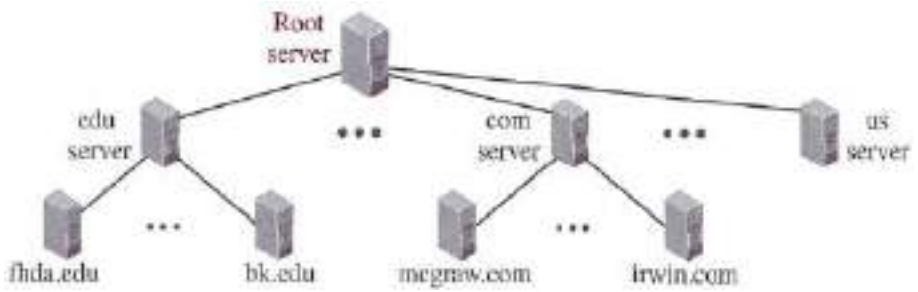


Distribution of Name Space

The information contained in the domain name space must be stored. However, it is very inefficient and also not reliable to have just one computer store such a huge amount of information. It is inefficient because responding to requests from all over the world places a heavy load on the system. It is not reliable because any failure makes the data inaccessible.

Hierarchy of Name Servers

The solution to these problems is to distribute the information among many computers called DNS servers. One way to do this is to divide the whole space into many domains based on the first level. In other words, we let the root stand alone and create as many domains (subtrees) as there are first-level nodes. Because a domain created this way could be very large, DNS allows domains to be divided further into smaller domains (subdomains). Each server can be responsible (authoritative) for either a large or small domain. In other words, we have a hierarchy of servers in the same way that we have a hierarchy of names (see The figure below).

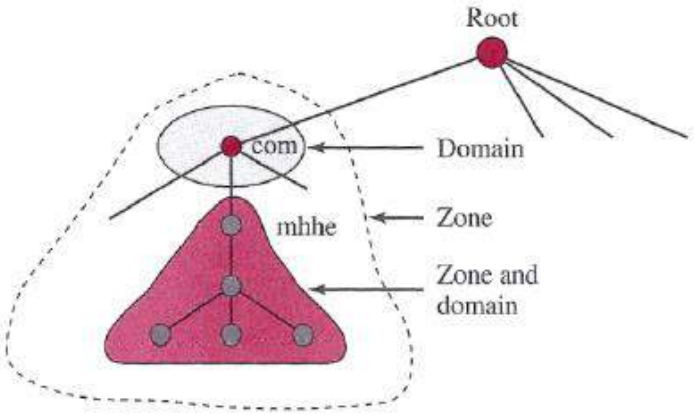


Zone

Since the complete domain name hierarchy cannot be stored on a single server, it is divided among many servers. What a server is responsible for or has authority over is called a zone. We can define a zone as a contiguous part of the entire tree.

If a server accepts responsibility for a domain and does not divide the domain into smaller domains, the "domain" and the "zone" refer to the same thing. The server makes a database called a zone file and keeps all the information for every node under that domain.

However, if a server divides its domain into subdomains and delegates part of its authority to other servers, "domain" and "zone" refer to different things. The information about the nodes in the subdomains is stored in the servers at the lower levels, with the original server keeping some sort of reference to these lower-level servers. Of course, the original server does not free itself from responsibility totally. It still has a zone, but the detailed information is kept by the lower-level servers (see the figure below).



Root Server

A root server is a server whose zone consists of the whole tree. A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers. There are several root servers, each covering the whole domain name space. The root servers are distributed all around the world.

Primary and Secondary Servers

DNS defines two types of servers: primary and secondary.

- **A primary server** is a server that stores a file about the zone for which it is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk.

- **A secondary server** is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. The secondary server neither creates nor updates the zone files. If updating is required, it must be done by the primary server, which sends the updated version to the secondary.

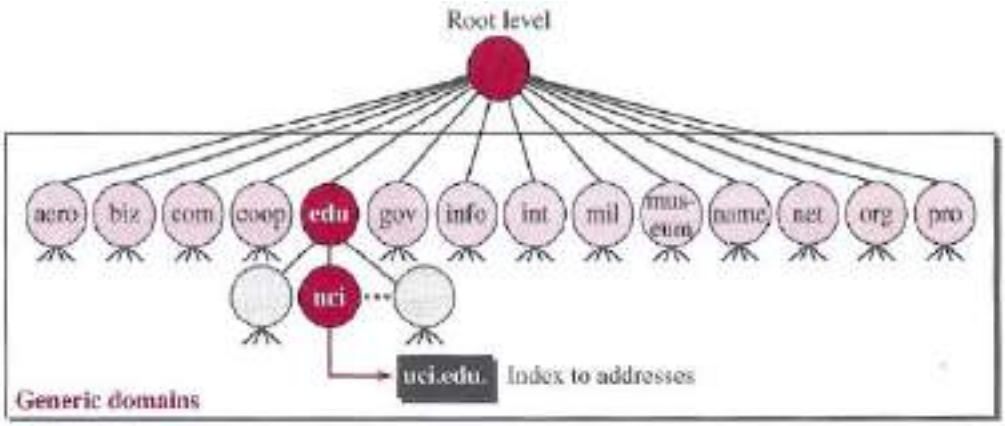
The primary and secondary servers are both authoritative for the zones they serve. The idea is not to put the secondary server at a lower level of authority but to create redundancy for the data so that if one server fails, the other can continue serving clients. Note also that a server can be a primary server for a specific zone and a secondary server for another zone. Therefore, when we refer to a server as a primary or secondary server, we should be careful about which zone we refer to.

DNS in the Internet

DNS is a protocol that can be used in different platforms. In the Internet, the domain name space (tree) was originally divided into three different sections: generic domains, country domains, and the inverse domains. However, due to the rapid growth of the Internet, it became extremely difficult to keep track of the inverse domains, which could be used to find the name of a host when given the IP address. The inverse domains are now deprecated (see RFC 3425). We, therefore, concentrate on the first two.

- **Generic Domains**

The generic domains define registered hosts according to their generic behavior. Each node in the tree defines a domain, which is an index to the domain name space database (see the figure below).



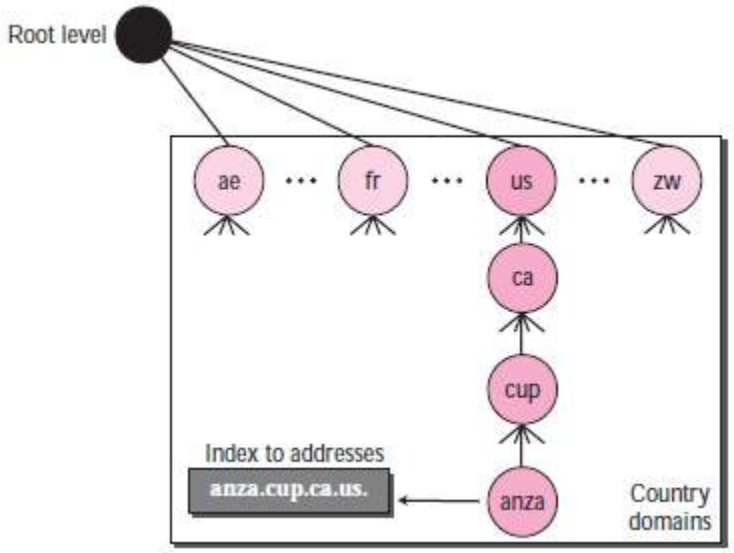
Looking at the tree, we see that the first level in the generic domains section allows 14 possible labels. These labels describe the organization types as listed in the table below.

Computer Networks

Label	*Description*
aero	Airlines and aerospace companies
biz	Businesses or firms (similar to "com")
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International organizations
mil	Military groups
museum	Museums and other non-profit organizations
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

- **Country Domains**

The country domains section uses two-character country abbreviations (e.g., us for United States). Second labels can be organizational, or they can be more specific national designations. The United States, for example, uses state abbreviations as a subdivision of us (e.g., ca.us.). The figure below shows the country domains section. The address uci.ca.us. can be translated to University of California, Irvine, in the state of California in the United States.



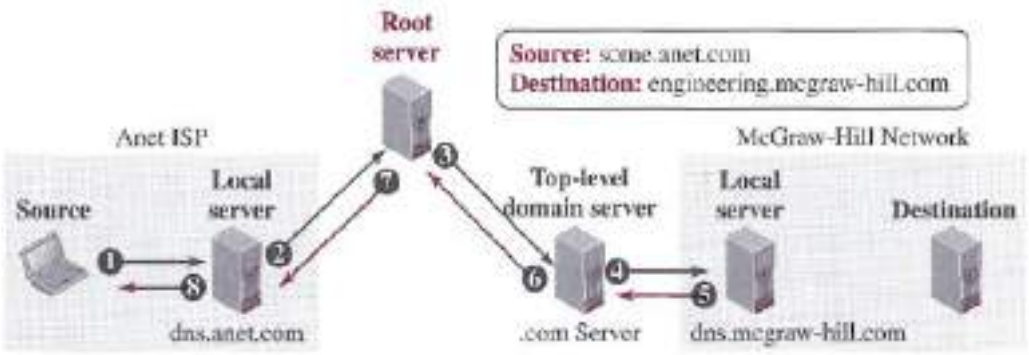
Resolution

Mapping a name to an address is called name-address resolution. DNS is designed as a client-server application. A host that needs to map an address to a name or a name to an address calls a DNS client called a resolver. The resolver accesses the closest DNS server with a mapping request. If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other

servers or asks other servers to provide the information. After the resolver receives the mapping, it interprets the response to see if it is a real resolution or an error, and finally delivers the result to the process that requested it. A resolution can be either recursive or iterative.

- **Recursive Resolution**

The figure below shows a simple example of a recursive resolution. We assume that an application program running on a host named some.anet.com needs to find the IP address of another host named engineering. mcgraw-hill. com to send a message to. The source host is connected to the Anet ISP; the destination host is connected to the McGraw-Hill network.

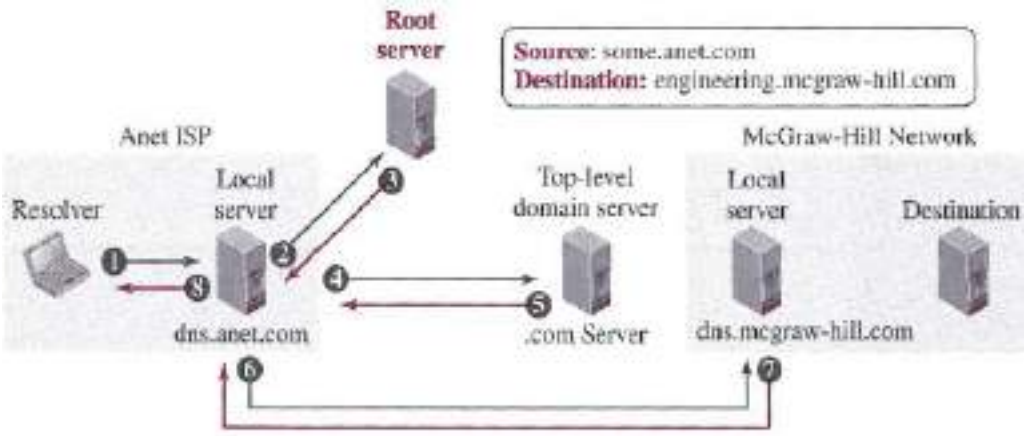


The application program on the source host calls the DNS resolver (client) to find the IP address of the destination host. The resolver, which does not know this address, sends the query to the local DNS server (for example, dns.anet.com) running at the Anet ISP site (event 1). We assume that this server does not know the IP address of the destination host either. It sends the query to a root DNS server, whose IP address is supposed to be known to this local DNS server (event 2). Root servers do not normally keep the mapping between names and IP addresses, but a root server should at least know about one server at each top level domain (in this case, a server responsible for com domain). The query is sent to this top-level-domain server (event 3). We assume that this server does not know the name-address mapping of this specific destination, but it knows the IP address of the local DNS server in the McGraw-Hill company (for example, dns.mcgraw-hill.comi. The query is sent to this server (event 4), which knows the IP address of the destination host. The IP address is now sent back to the top-level DNS server (event 5), then back to the root server (event 6), then back to the ISP DNS server, which may cache it for the future queries (event 7), and finally back to the source host (event 8).

- **Iterative Resolution**

In iterative resolution, each server that does not know the mapping sends the IP address of the next server back to the one that requested it. The figure below shows the flow of information in an iterative

resolution in the same scenario as the one depicted in Recursive Resolution. Normally the iterative resolution takes place between two local servers; the original resolver gets the final answer from the local server. Note that the messages shown by events 2, 4, and 6 contain the same query. However, the message shown by event 3 contains the IP address of the top-level domain server, the message shown by event 5 contains the IP address of the McGraw-Hill local DNS server, and the message shown by event 7 contains the IP address of the destination. When the Anet local DNS server receives the IP address of the destination, it sends it to the resolver (event 8).



1

Computer Networks II

Course Number: CPE 407

2

Computer Networks II

Course Contents

- Introduction:
- General Review: Network Types (LAN, WAN, MAN, SAN, VPN, …), OSI and TCP/IP Model, IP addressing, Review,
- Wired LANs: Ethernet, IEEE Standards and Standard Ethernet, MAC Sublayer, Physical Layer, Bridged Ethernet, Switched Ethernet, Full-Duplex Ethernet,
- Wireless LANs, IEEE 802.11, Bluetooth, Connecting LANs, Backbone Networks,
- Virtual LANs: Connecting Devices, Backbone Networks, and Virtual LANs , Wireless WANs,
- Cellular a view on: Telephone, Satellite Networks, Synchronous Optical Network (SONET/SDH), SONET Architecture, SONET layers, SONET Network,
- Virtual-Circuit Networks: Frame Relay, ATM.

Introduction
Fundamentals of Data Networks

Introduction

- Data communications and networking
  - Change the way we do business and the way we live
  - Business decisions have to be made more quickly
  - Decision depends on immediate access to accurate information
  - Business today rely on computer networks and internetworks
  - Before get hooked up, we need to know:
    - How networks operate
    - What types of technologies are available
    - Which design best fills which set of needs

5

Introduction

- Development of the PC changes a lot in business, industry, science and education.
- Similar revolution is occurring in data communication and networking
  - Technologies advances are making it possible for communications links to carry more and faster signals
  - Services are evolving to allow the use of this expanded capacity
  - For example telephone services extended to have:
    - Conference calling
    - Call waiting
    - Voice mail
    - Caller ID

6

Data Communications

Communication:

- Means sharing information
  - Local (face to face) or remote (over distance)
- Telecommunication
  - Telephone, telegraph and television
  - Means communication at a distance
  - Tele is Greek for far

7

Data Communications

Data:

- Refers to information
  - Presented in any form
  - Agreed upon by the parties (creating & using)

Data communication : is the exchange of data between two devices via some form of transmission medium (wire cable).

8

Data Communications

- Communication system made up of a combination of hardware and software
- Effectiveness of data communication system depends on:
  1. Delivery : The system must deliver data to correct destination. Data received by the indented user only
  2. Accuracy: The system must deliver data accurately (no change).
     - Data changed & uncorrected is unusable

9

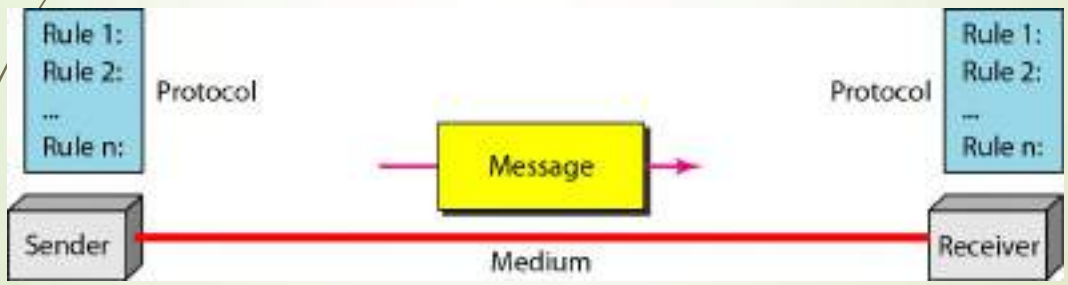
Data Communications

3. Timeliness: The system must deliver data in timely manner
   - Data arrived late are useless
   - In the same order (video and audio) & without delay (Real time transmission)
4. Jitter: Variation in the packet arrival time (uneven quality in the video is the result)

10

Components

- A data communication system is made up of five components



11

Components

1. Message: the information (data) to be communicated
   - Consist of text, numbers, pictures, audio, or video
2. Sender: the device that sends the data message
   - Computer, workstation, telephone handset, video camera, …
3. Receiver: the device that receives the message
   - Computer, workstation, telephone handset, television, ….

12

Components

4. Medium: The physical path by which a message travels from sender to receiver
   - twisted pair, coaxial cable, fiber-optic, radio waves

13

Components

5. Protocol: a set of rules that govern data communications
   - An agreement between the communicating devices
   - Devices may be connected but not communicating (no protocol)
   - Arabic speaker with Japanese speaker

14

Data Representation

- Text
- Numbers
- Images
- Audio
- Video

15

Data Representation

- **Text**:
  - Sequence of bits (0s or 1s)
  - Different sets of patterns to represent text symbols (each set is called: *code*)
  - ASCII: 7 bits (128 symbols)
  - common coding system today is:
  - Unicode uses: 32 bits to represent a symbol or character in any language

Unicode

(4,294,967,296)

16

Data Representation

- Numbers:
  - Represented by bit patterns
  - The number is directly converted to a binary number

17

Data Representation

- Images:
  - Represented by bit patterns
  - A matrix of pixels
  - Resolution: size of the pixels
  - High resolution: more memory is needed
  - Each pixel is assigned a bit pattern
    - 1-bit pattern (black and white dots image)
    - 2-bit pattern (4 levels of gray)
    - RGB (color images)

18

Data Representation

- Audio:
  - Continuous not discrete
  - Change to digital signal
- Video:
  - Recording or broadcasting of a picture or movie
  - Change to digital signal

19

Data Flow

- Communication between two devices can be:
  - Simplex
  - Half-Duplex
  - Full-Duplex

20

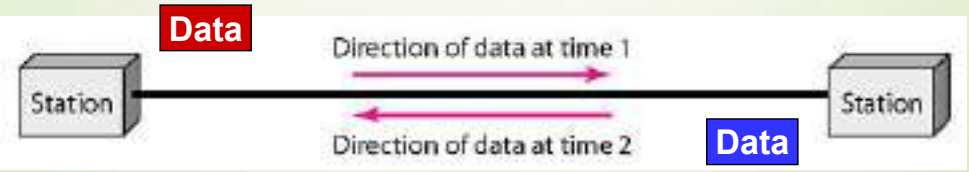
Data Flow

- **Simplex (one way street)**
  - The communication is unidirectional
  - Only one device on a link can transmit; the other can only receive
  - Use the entire capacity of the channel to send data
  - Example: Keyboards, Monitors

21

Data Flow

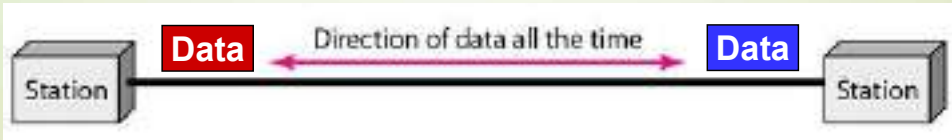
- **Half-Duplex (one-lane with two-directional traffic)**
  - Each station can both transmit and receive, but not at the same time
  - When one device is sending, the other can only receive, and vice versa
  - The entire capacity of a channel is taken over by the transmitting device
  - Example: Walkie-talkies



22

Data Flow

- **Full-Duplex (Duplex) (two-way street)**
  - Both stations can transmit and receive at same time
  - Signals going in either direction sharing the capacity of the link
  - Sharing can occur in two ways:
    - Link has two physically separate transmission paths
      - One for sending and the other for receiving
    - The capacity of the channel is divided between signals travelling in both directions
  - Example: Telephone network



23

Networks

- Network : A set of devices (nodes) connected by communication links

  Node : computer, printer, …

  - Distributed Processing :
    - Most networks used it
    - Task is divided among multiple computers instead of one single large computer

24

Networks

- Network Criteria
  - Network must meet a certain number of criteria
  - The most important of the network criterions are:
    - Performance
    - Reliability
    - Security

25

Networks

- Performance
  - Transit time: A mount of time required for a message to travel from one device to another
  - Response time: Elapsed time between an inquiry and a response

26

Networks

- Performance
  - Performance depends on :

    1- Number of users: large number slow response time.

    2- Type of transmission medium: fiber-optic cabling faster than others cables.

    3- Capabilities of the connected hardware: affect both the speed and capacity of transmission.

    4- Efficiency of the software: process data at the sender and receiver and intermediate affects network performance.

27

Networks

- Performance
  - Performance is evaluated by two contradictory networking metrics:
    - Throughput (high): a measure of how fast we can actually send data through a network
    - Delay (low)

28

Networks

- Reliability
  - Reliability is measured by:
    1. Frequency of failure
    2. Recovery time of a network after a failure
    3. Network's robustness in a catastrophe: protect by good back up network system

29

Networks

- Security
  - Protecting data from unauthorized access
  - Protecting data from damage and development
  - Implementing policies and procedures for recovery from breaches and data losses (Recovery plan)

30

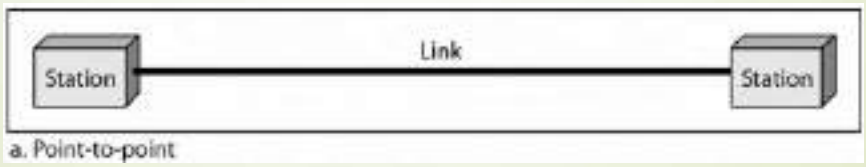
Networks

- Physical Structures:
  - Type of connection
    - Network: Two or more devices connected through links
    - Link: Communication pathway that transfers data from one device two another
    - Two devices must be connected in some way to the same link at the same time. Two possible types:
      - Point-to-Point
      - Multipoint

31

Networks

- **Point-to-Point**
  - Dedicated link between two devices
  - Entire capacity of the link is reserved for transmission between those two devices
  - Use an actual length of wire or cable



a. Point-to-point

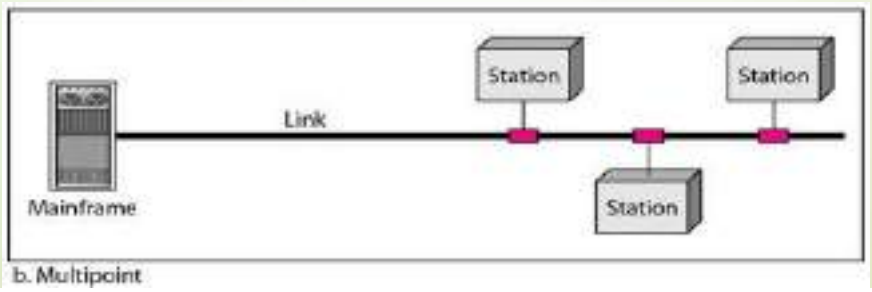
32

Networks

- **Point-to-Point**
  - Other options, such as microwave or satellite is possible
  - Example: Television remote control

Networks

- Multipoint (multidrop)
  - More than two devices share a single link
  - Capacity is shared
  - Channel is shared either spatially or temporally
    - Spatially shared: if devices use link at same time
    - Timeshare: if users must take turns



b. Multipoint

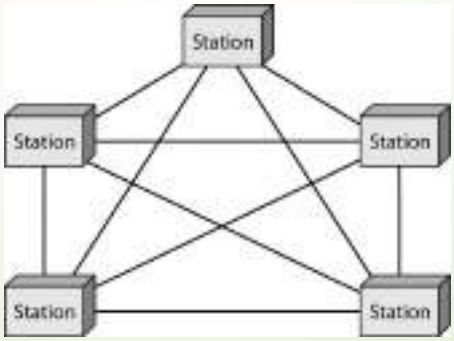
Networks

- **Physical Topology**
  - The way a network is laid out physically
  - 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 (nodes) to one another.
  - Four topologies : Mesh, Star, Bus, and Ring

35

Physical Topology

- **Mesh**
  - Every link is dedicated point-to-point link
  - The term dedicated means that the link carries traffic only between the two devices it connects



36

Physical Topology

- **Mesh**
  - To link ***n*** devices fully connected mesh has:
    n (n - 1) / 2 physical channels (Full-Duplex)
  - Every Device on the network must have
    n - 1 ports

37

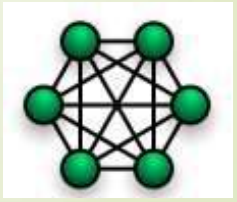
Physical Topology

- **Mesh**
  - Example:

    8 devices in mesh has links: n(n-1) / 2

    number of links = 8 (8-1)/2 = **28**

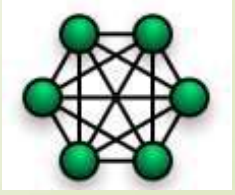
    number of ports per device = n – 1 = 8 –1 = **7**



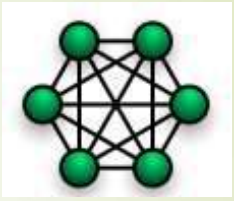
38

Physical Topology

- **Mesh**
  - Advantages
    - Each connection carry its own data load (no traffic problems)
    - A mesh topology is robust
    - Privacy or security
    - Fault identification and fault isolation



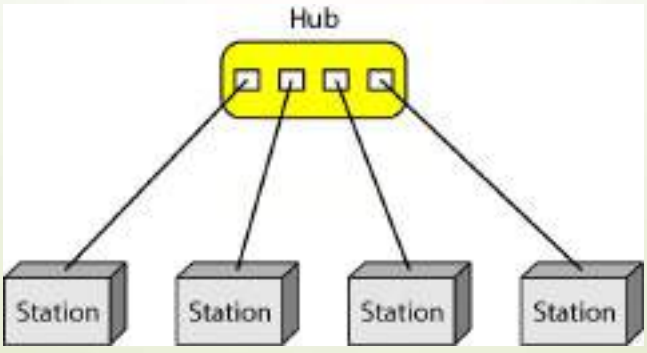
Physical Topology



- **Mesh:**
  - Disadvantages
    - Big amount of cabling
    - Big number of I/O ports
    - Installation and reconnection are difficult
    - Sheer bulk of the wiring can be greater than the available space
    - Hardware connect to each I/O could be expensive
- Mesh topology is implemented in a limited fashion; e.g., as backbone of hybrid network

Physical Topology

- **Star**:
  - Dedicated point-to-point to a central controller (Hub)
  - No direct traffic between devices
  - The control acts as an exchange



41

Physical Topology

- **Star**
  - Advantages
    - Less expensive than mesh (1 Link + 1 port per device)
    - Easy to install and reconfigure
    - Less cabling
    - Additions, moves, and deletions required one connection
    - Robustness : one fail does not affect others
    - Easy fault identification and fault isolation

42

Physical Topology

- **Star**
  - Disadvantages
    - Dependency of the whole topology on one single point (hub)
    - More cabling than other topologies (ring or bus)
- Used in LAN

43

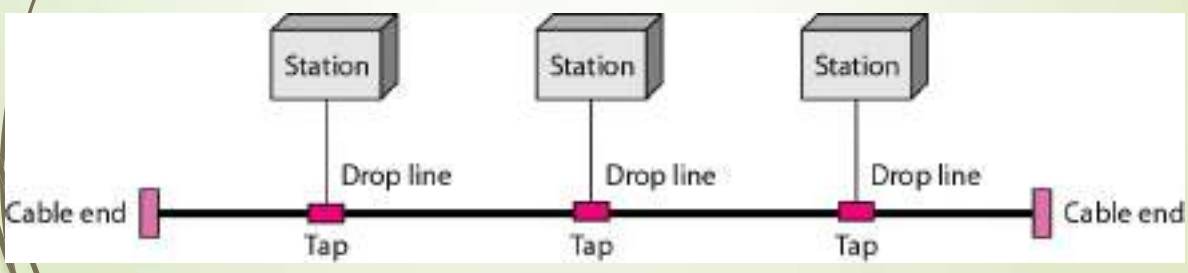
Physical Topology

- **Bus**
  - It is multipoint
  - One long cable acts as a backbone
  - Used in the design of early LANS, and Ethernet LANs

44

Physical Topology

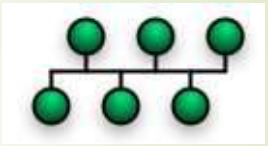
- **Bus**
  - Nodes connect to cable by drop lines and taps
  - Signal travels along the backbone and some of its energy is transformed to heat
  - Limit of number of taps and the distance between taps



45

Physical Topology

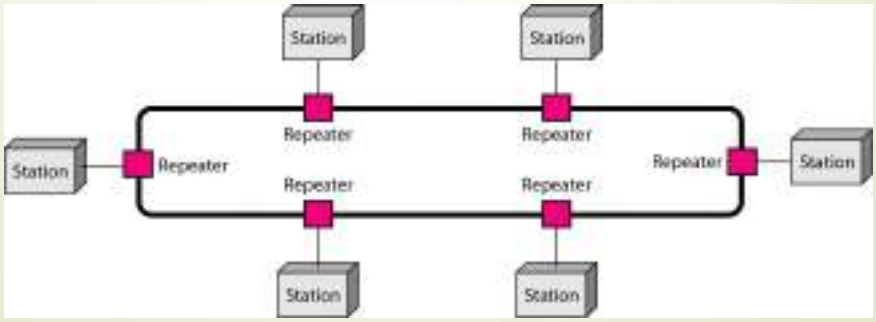
- **Bus**
  - Advantages
    - Ease of installation
    - Less cables than mesh, star topologies
  - Disadvantages
    - Difficult reconnection and fault isolation (limit of taps)
    - Adding new device requires modification of backbone
    - Fault or break stops all transmission
    - The damaged area reflects signals back in the direction of the origin, creating noise in both directions



46

Physical Topology

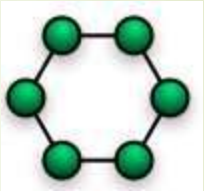
- **Ring**
  - Each device has dedicated point-to-point connection with only the two devices on either side of it
  - A signal is passed along the ring in one direction from device to device until it reaches its destination
  - Each devices incorporates a Repeater



47

Physical Topology

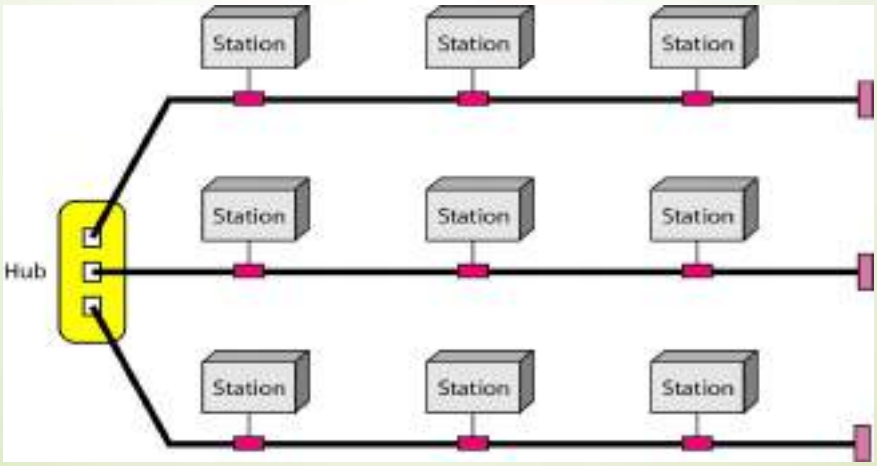
- **Ring**
  - Advantages
    - Easy of install and reconfigure
    - Connect to immediate neighbors
    - Move two connections for any moving (Add/Delete)
    - Easy of fault isolation
  - Disadvantage
    - Unidirectional
    - One broken device can disable the entire network. This weakness can be solved by using a dual ring or a switch capable of closing off the break



48

Physical Topology

- **Hybrid Topology**
  - Example: having a main star topology with each branch connecting several stations in a bus topology



49

Categories of Networks

- Network Category depends on its size
- Two primary categories
  - **LAN**: Covers area < 2miles
  - **WAN**: Can be worldwide
  - **MAN**: Between LAN & WAN, span 10s of miles

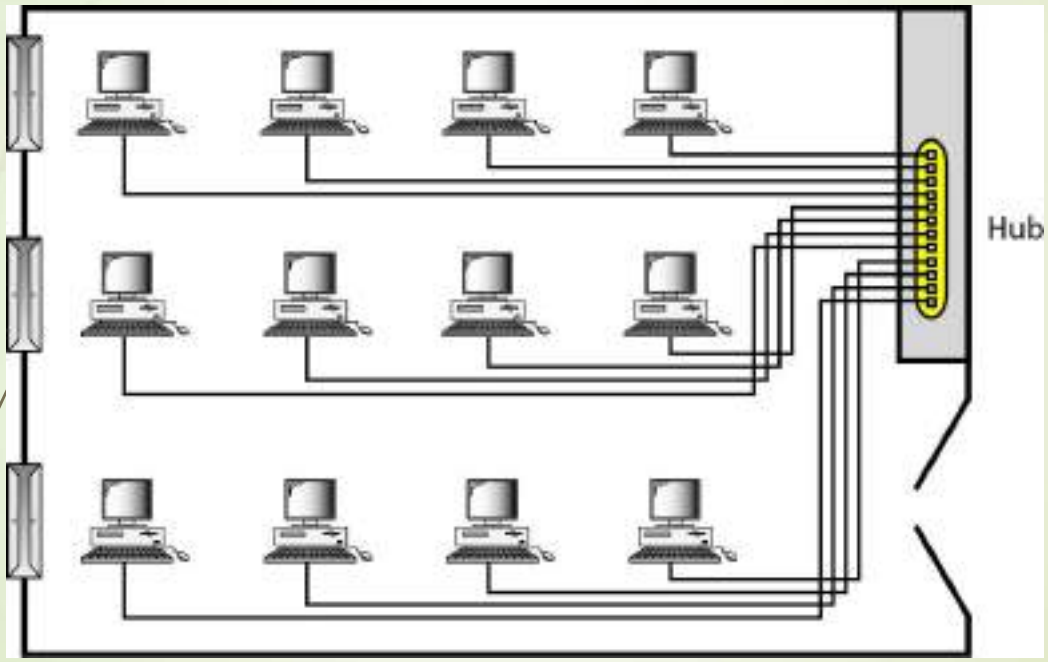
50

Local Area Network (LAN)

- Privately owned
- Links devices in the same office, building, or campus
- Simple LAN: 2 PCs & 1 printer in home or office
- Size is limited to a few kilometers
- Allow resources to be shared (hardware, software, or data)

51

Local Area Network (LAN)



An isolated LAN connecting 12 computers to a hub in a closet

52

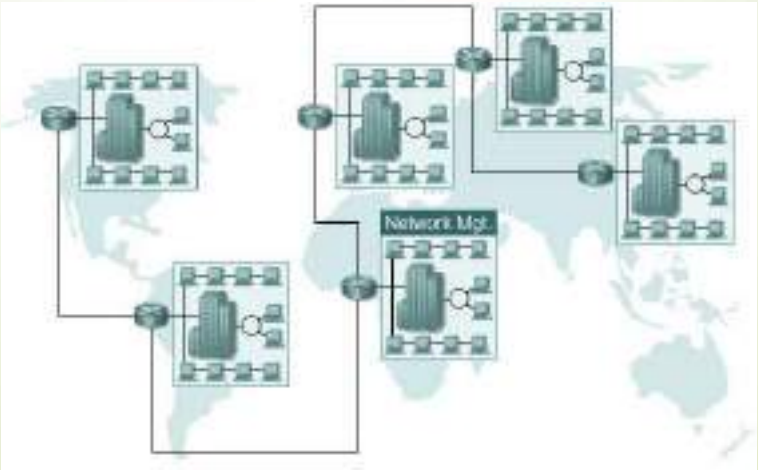
Local Area Network (LAN)

- LAN is distinguished by:
  - Size (# users of OS, or licensing restrictions)
  - Transmission medium (only one type)
  - Topology (bus, ring, star)
- Data Rates (speed):
  - Early: 4 to 16 Mbps
  - Today: 100 to 1000 Mbps

53

Wide Area Networks (WAN)

- Provides long-distance transmission of data over large geographic areas (country, continent, world)

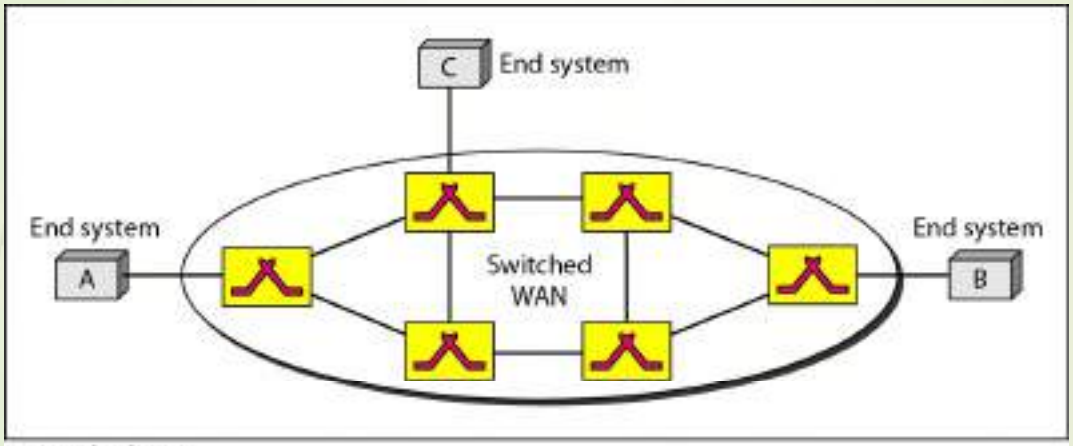


54

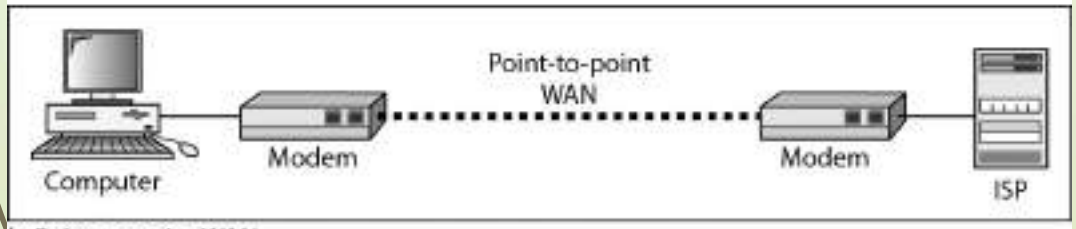
Wide Area Networks (WAN)

- Switched WAN
  - Backbone of the Internet
- Dialup line point-to-point WAN
  - Leased line from a telephone company

Wide Area Networks (WAN)



a. Switched WAN



b. Point-to-point WAN

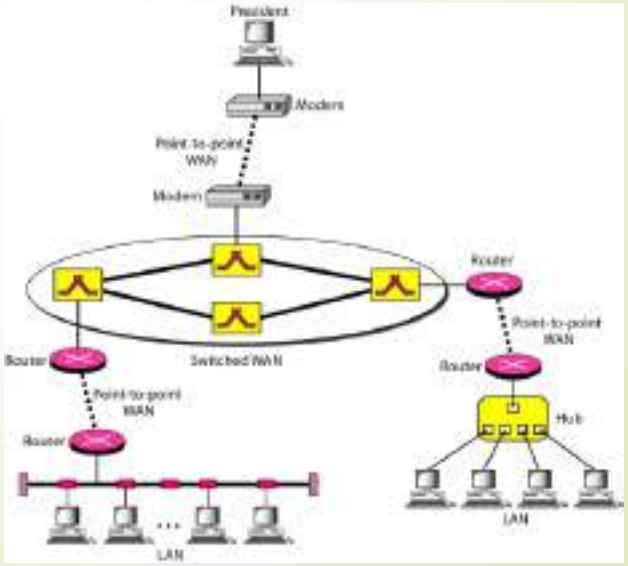
Metropolitan Area Networks (MAN)

- Size between LAN and WAN
- Inside a town or a city
- Example: the part of the telephone company network that can provide a high-speed DSL to the customer

57

Interconnection of Networks: Internetworks

- Two or more networks connected together



58

Campus area network (CAN)

- A campus area network, or CAN, is a network used in educational environments such as universities or school districts. While each department in a school might use its own LAN, all the school's LANs could connect through a CAN.
- Campus area networks combine several independent networks into one cohesive unit. For example, the English and engineering departments at a university might connect through a CAN to communicate with each other directly.

59

Storage area network (SAN)

- A storage area network, or a SAN, is a network that teams use to store mass amounts of sensitive data.
- It provides a way to centralize data on a non-localized network that differs from the main operating one. One example of a SAN is if your team stores customer information on a separate network to maintain the high speeds of your main network.

60

Virtual private network (VPN)

- A virtual private network, or VPN, is a private network that's available through the internet. This type of network functions similarly to an EPN (**Enterprise private network**) because it provides a secure, private connection.
- VPNs typically don't require the same infrastructure as EPNs. Both the general public and companies can use VPNs to ensure privacy and security.

61

The Internet

- 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.
- Internet is a communication system that has brought a wealth of information to our fingertips and organized it for our use
- An internet is 2 or more networks that can communicate with each other
- The Internet is a collaboration of more than hundreds of thousands of interconnected networks

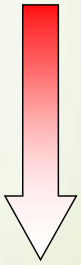
62

The Internet

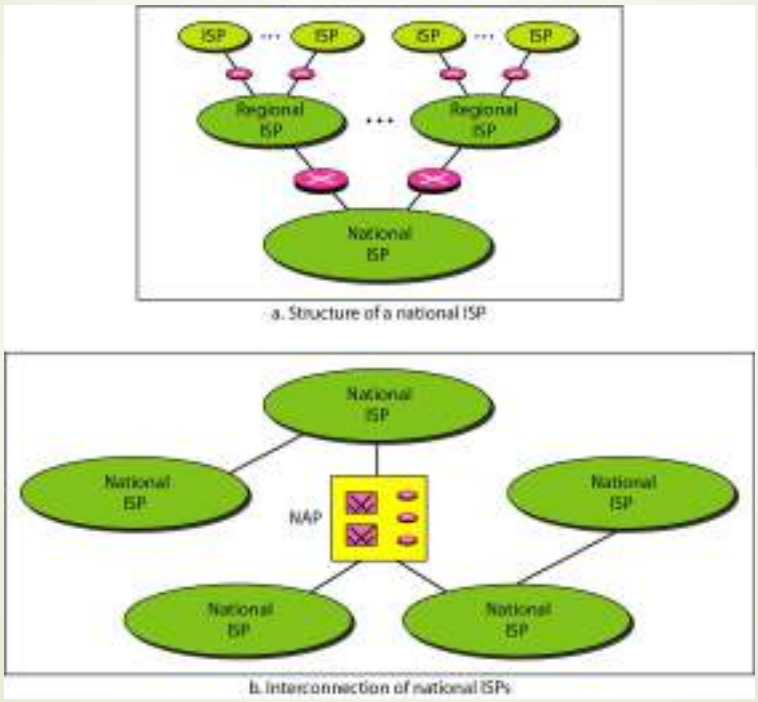
- An internet (small i) is two or more networks
- Notable internet is called the Internet (hundreds of thousands interconnected networks)
  - Private individuals + government agencies + school + research facilities + Corporations + libraries in more than 100 countries
- This communication system came in 1969
- Mid-1960 (ARPA) Advanced Research Projects Agency in (DOD) was interested to connect mainframes in research organizations
- 1967, ARPA presented its ideas for ARPANET
  - Host computer connecting to (IMP) interface message processor.
  - Each IMP communicate with other IMP
- 1969, four nodes (universities) connected via IMPs to form a network
  - Software (NCP) Network Control Protocol provided communication between the hosts.
- 1972, Vint Cerf and Bob Kahn invented (TCP) Transmission Control Protocol
- Later TCP was split to (TCP) Transmission Control Protocol and (IP) Internetworking Protocol

The Internet

- Internet Today
  - Made of many LANs and WANs
  - Every day new networks area added and removed
  - Internet services Providers (ISPs) offer services to the end users
    - International service providers
    - National service providers
    - Regional service providers
    - Local service providers



The Internet



Hierarchical organization of the Internet

65

Protocols and Standards

- Protocol synonymous with rule
- Standards: agreed-upon rules
- Protocols
  - A protocol is a set of rules that govern data communications
  - Defines What, How, and When it is communicated

66

Protocols and Standards

- Elements of a protocol:
  - ***Syntax:*** structure or format of data
    - Example: 8-bits address of sender, 8-bits address of receiver
  - ***Semantics:*** meaning of each section of bits
    - Example: Does the address is a route to be taken or the final destination of the message
  - ***Timing:*** when data should be sent and how fast they can be sent
    - Example: sender produces data at 100 Mbps but the receiver can process data at only 1 Mbps ⇒ overload and data loose

67

Standards

- Essential in creating and maintaining an open and competitive market for equipment manufactures
- Guaranteeing national and international interoperability of data and telecommunication technology and processes
- Providing guidelines to manufacturers, vendors, government agencies, and other service providers to ensure the kind of interconnectivity necessary in today's marketplace and in international communications

68

Standards

- **Two categories**
  - De facto: not approved by an organized body but adopted as standards through widespread use
  - De jure: Legislated by an officially recognized body

69

Standards

- Standards are developed through the cooperation of:
  - Standards Creation Committees
    - ISO, ITU-T, CCITT, ANSI, IEEE, EIA
  - Forums
    - Created by special-interest groups
    - Present their conclusions to the standards bodies
  - Regulatory Agencies
    - Ministry of Telecommunication(Iraq).
    - Purpose: Protecting the public by regulating radio, television, and communication

70

Standards

- Internet standards
  - Tested thoroughly tested specification that is useful to be adhered to by those who work with the Internet
  - Formalized regulation that must be followed
  - Specification become Internet standard
    - Begins as Internet draft for 6 months
    - Upon recommendation from the Internet authorities draft published as Request for Comment (RFC)
    - RFC is edited, assigned a number, and made available to all interested parties

2. Network Models

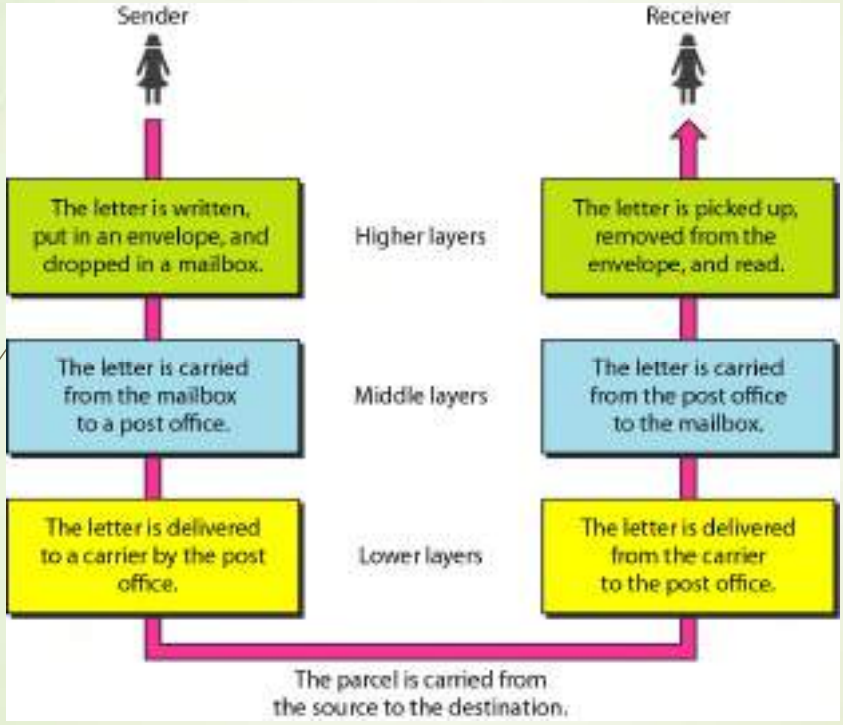
2-1 LAYERED TASKS

We use the concept of layers in our daily life. As an example, let us consider two friends who communicate through postal mail. The process of sending a letter to a friend would be complex if there were no services available from the post office.

Topics discussed in this section:

Sender, Receiver, and Carrier
Hierarchy

Figure 2.1 Tasks involved in sending a letter



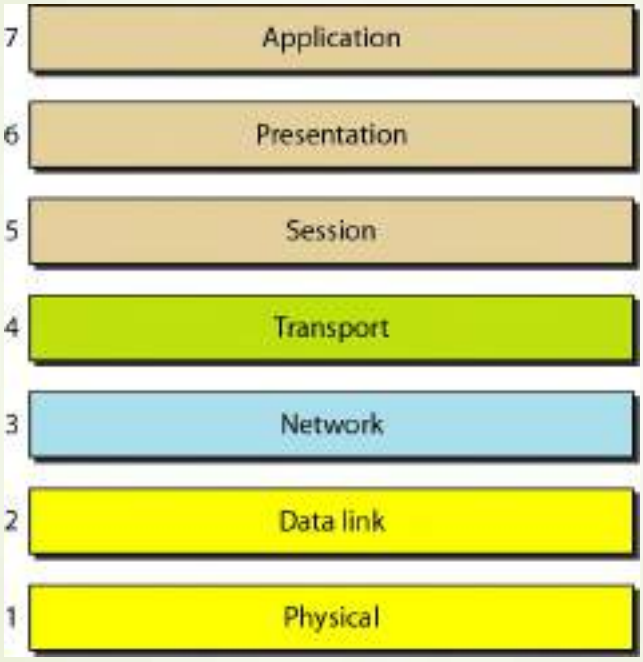
2-2 THE OSI MODEL

Established in 1947, the International Standards Organization (ISO) is a multinational body dedicated to worldwide agreement on international standards. An ISO standard that covers all aspects of network communications is the Open Systems Interconnection (OSI) model. It was first introduced in the late 1970s.

ISO is the organization.
OSI is the model.

5

Figure 2.2 *Seven layers of the OSI model*



6

Figure 2.3 *The interaction between layers in the OSI model*

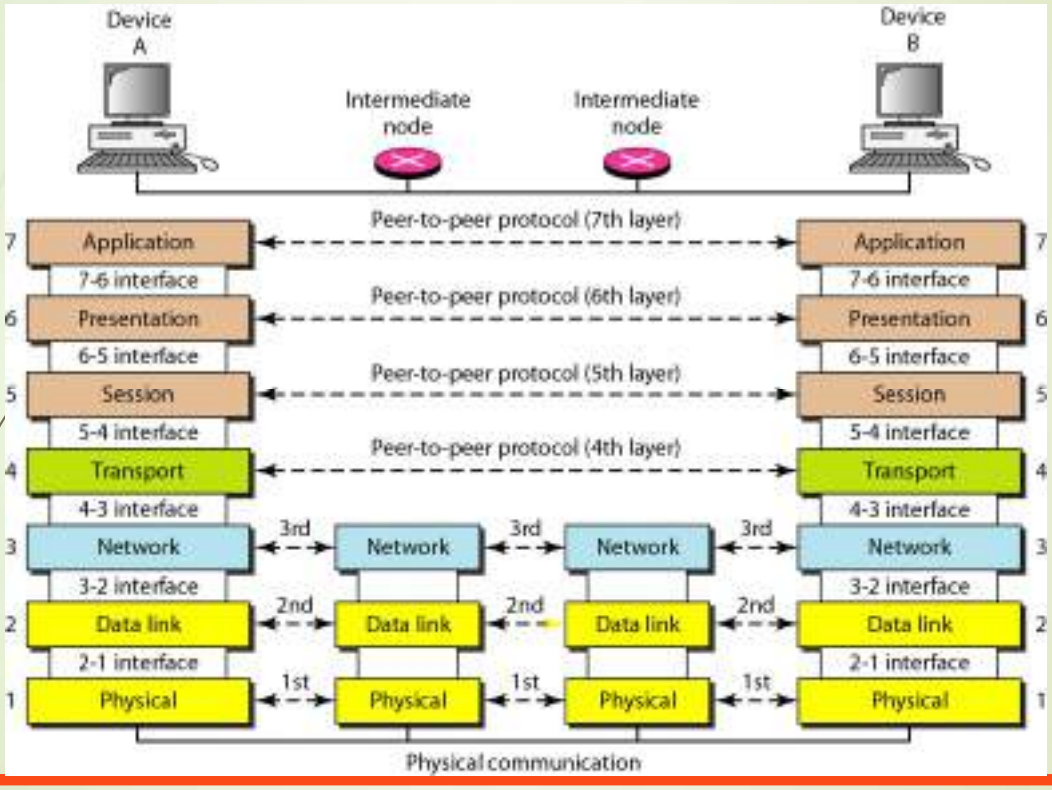
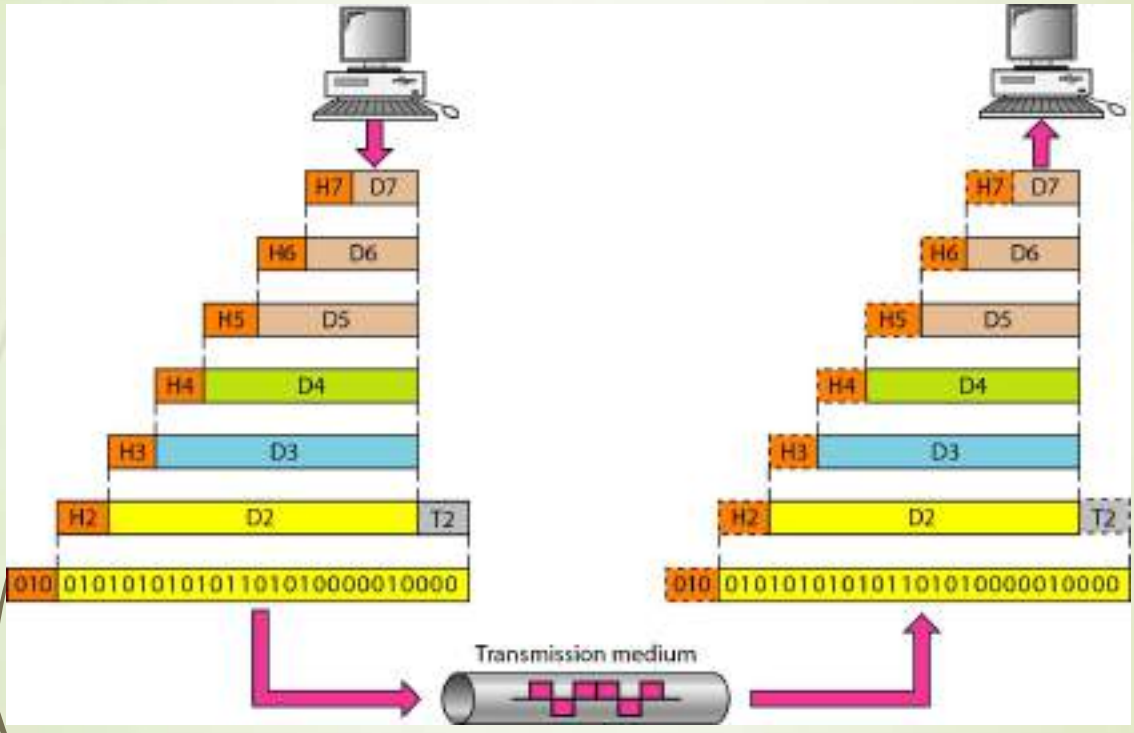


Figure 2.4 *An exchange using the OSI model*



2-3 LAYERS IN THE OSI MODEL

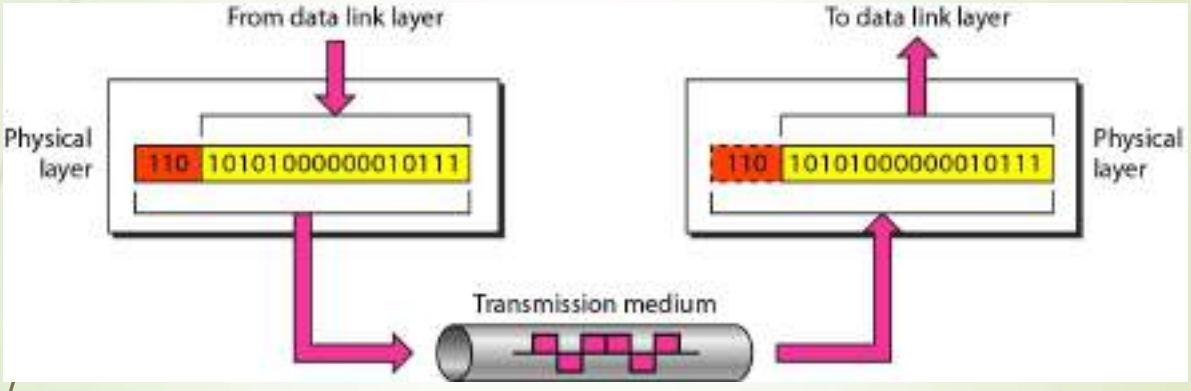
In this section we briefly describe the functions of each layer in the OSI model.

Topics discussed in this section:

- Physical Layer
- Data Link Layer
- Network Layer
- Transport Layer
- Session Layer
- Presentation Layer
- Application Layer

Figure 2.5 *Physical layer*

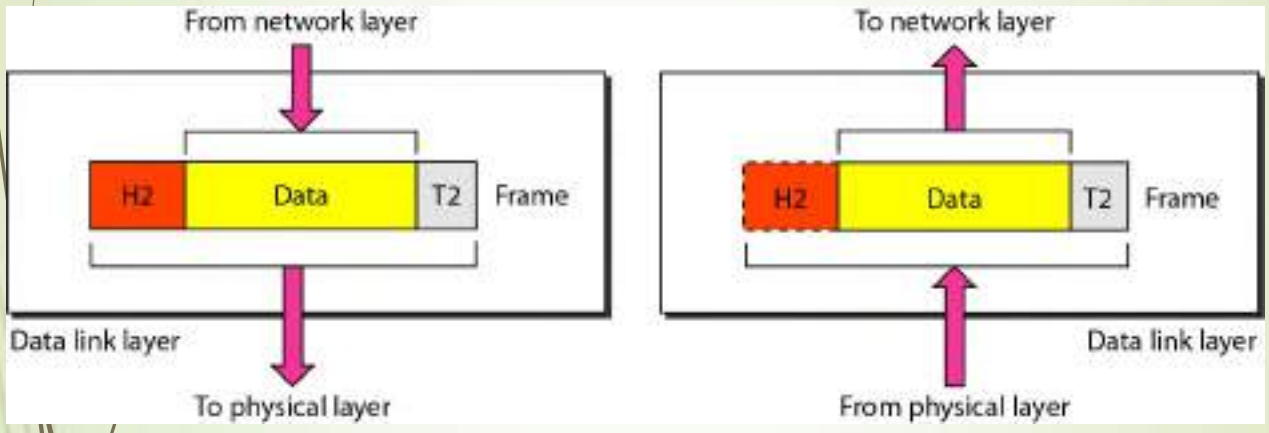
9



The physical layer is responsible for movements of individual bits from one hop (node) to the next.

Figure 2.6 *Data link layer*

10



The data link layer is responsible for moving frames from one hop (node) to the next.

Figure 2.7 *Hop-to-hop delivery*

11

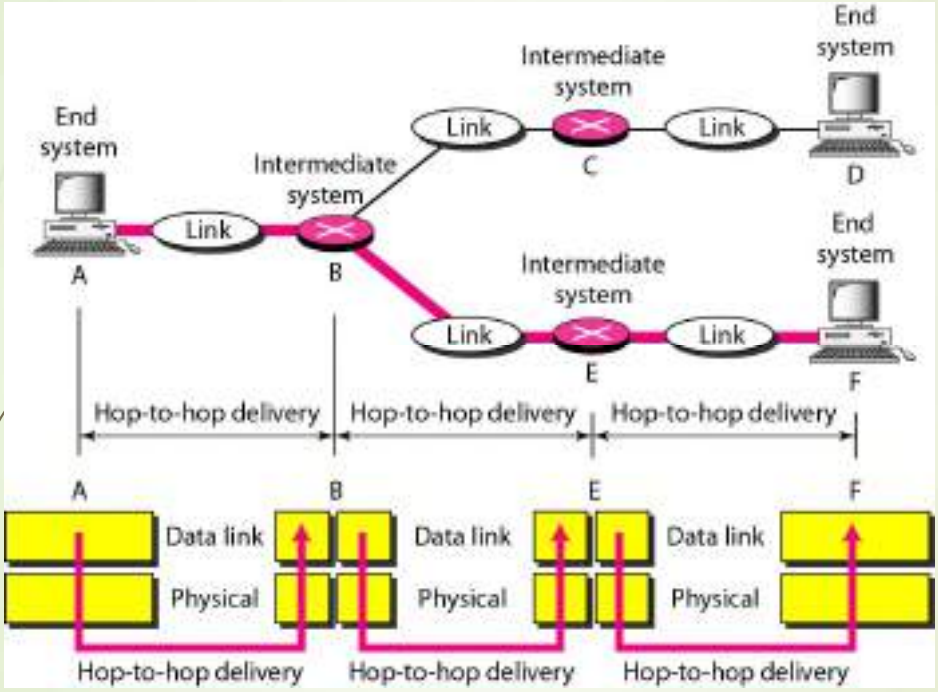
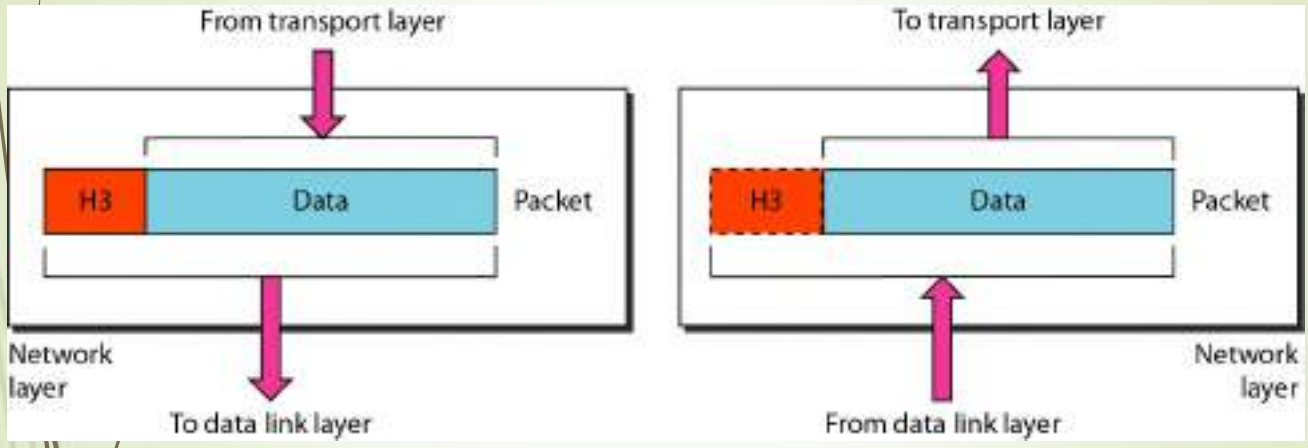


Figure 2.8 *Network layer*

12



The network layer is responsible for the delivery of individual packets from the source host to the destination host.

Figure 2.9 *Source-to-destination delivery*

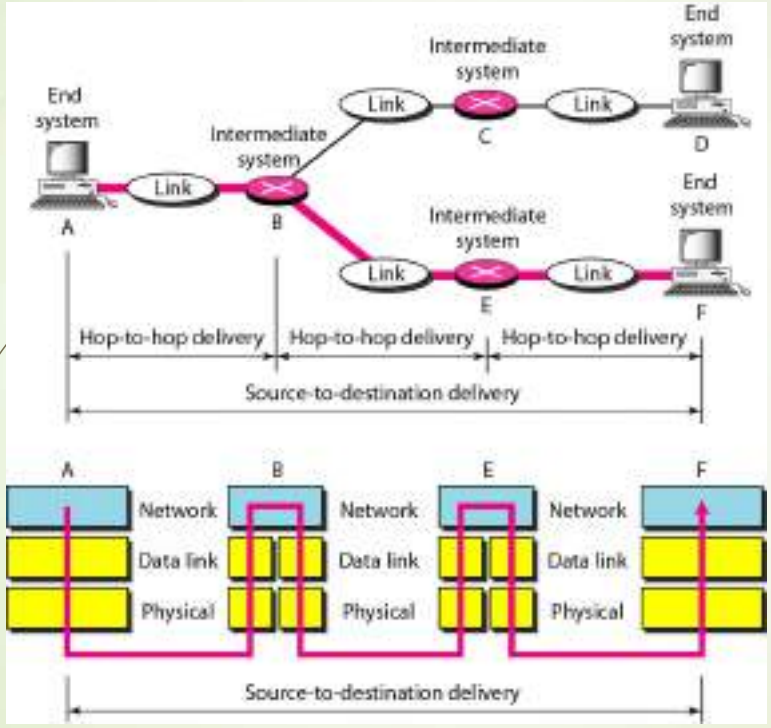
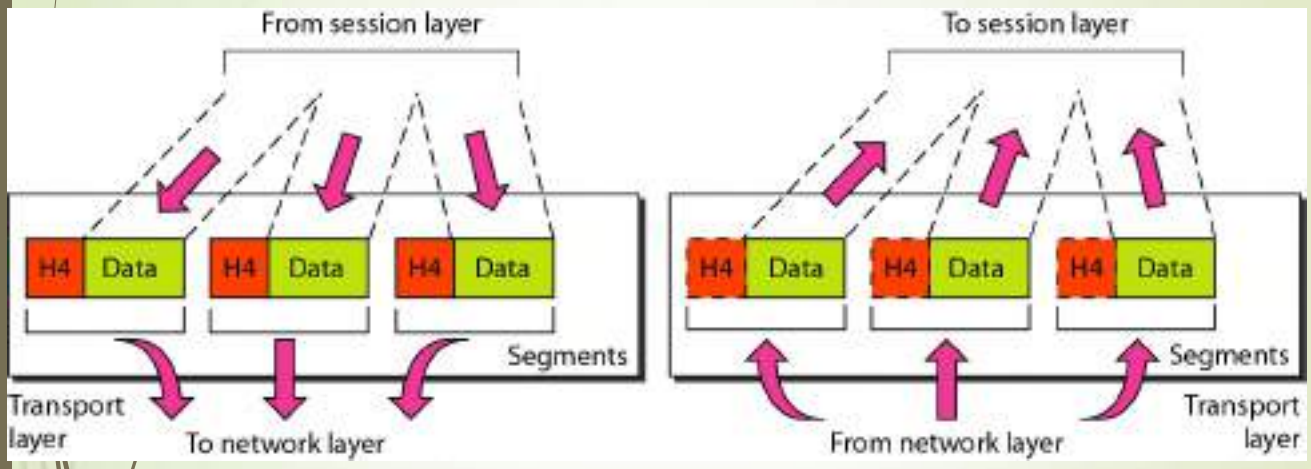


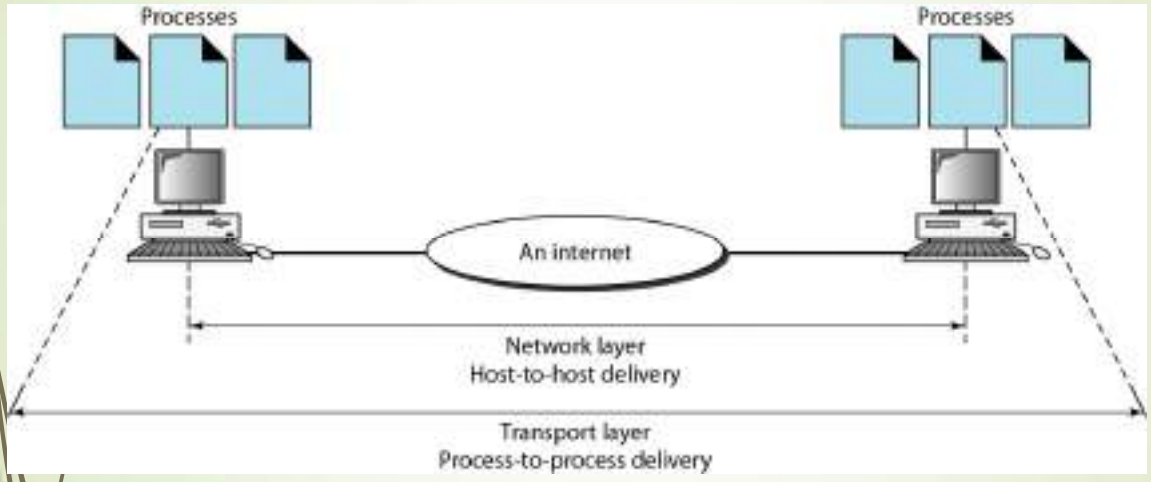
Figure 2.10 *Transport layer*



The transport layer is responsible for the delivery of a message from one process to another.

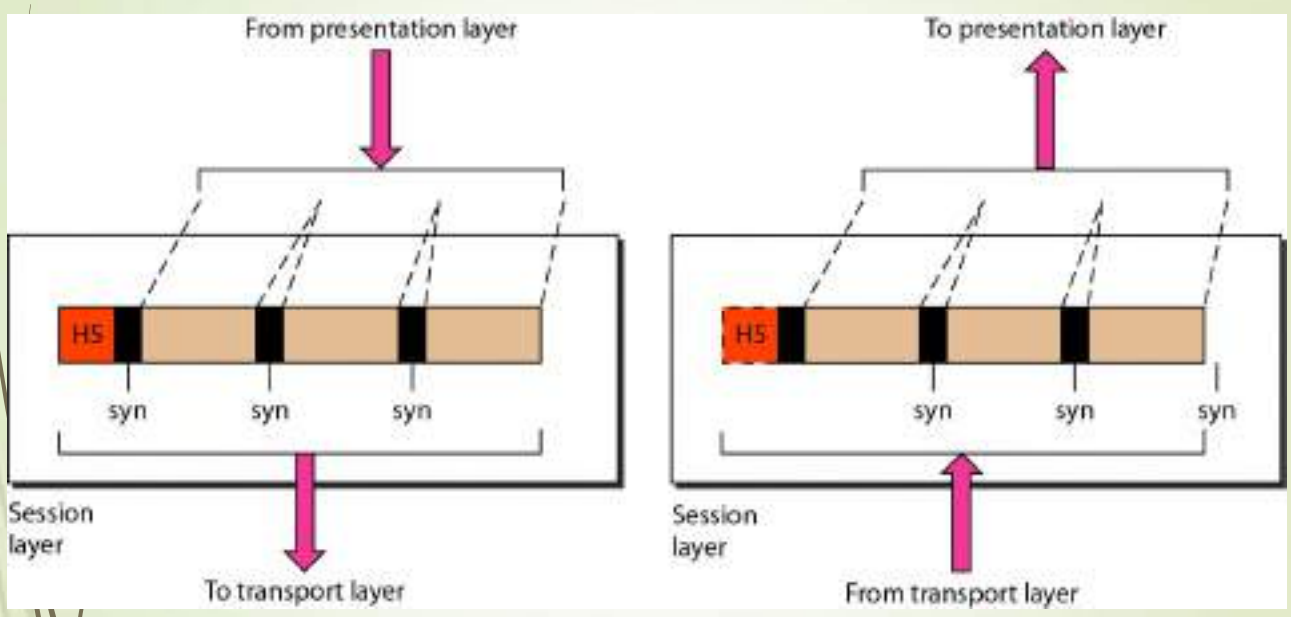
Figure 2.11 *Reliable process-to-process delivery of a message*

15



16

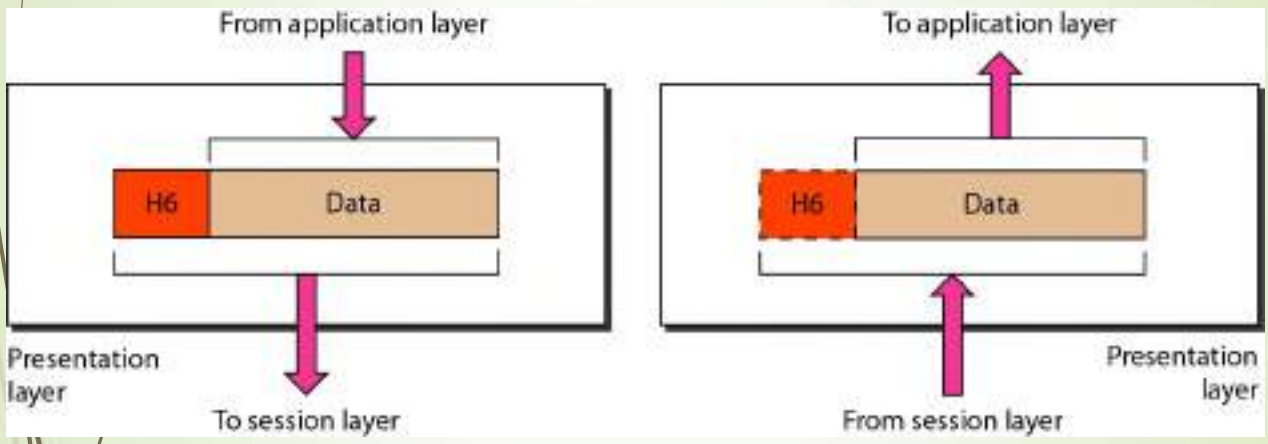
Figure 2.12 *Session layer*



The session layer is responsible for dialog control and synchronization.

Figure 2.13 *Presentation layer*

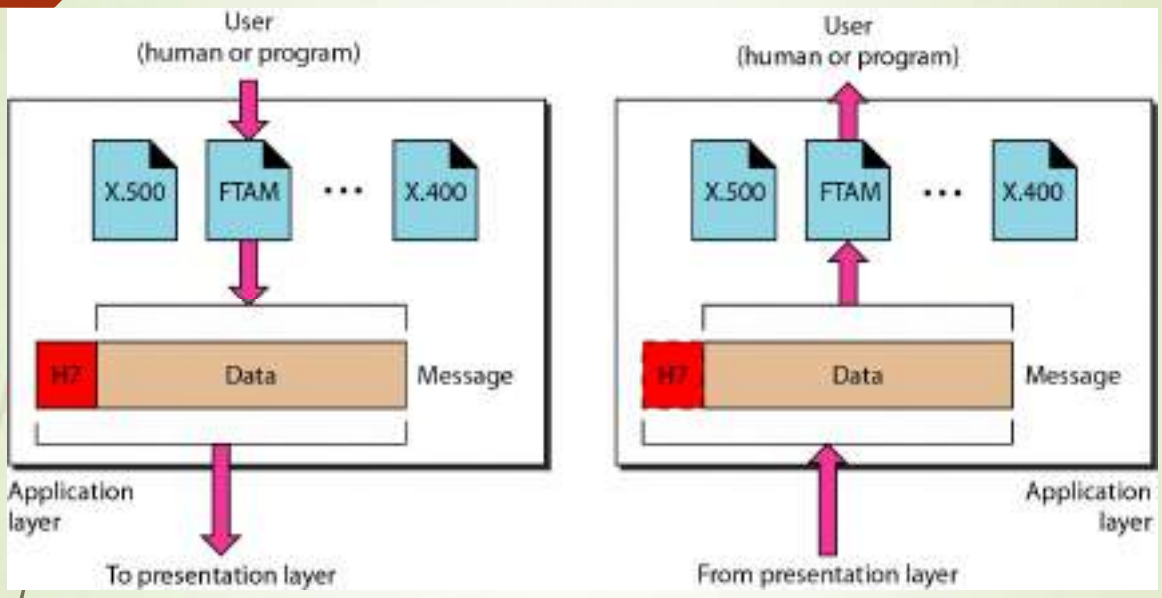
17



The presentation layer is responsible for translation, compression, and encryption.

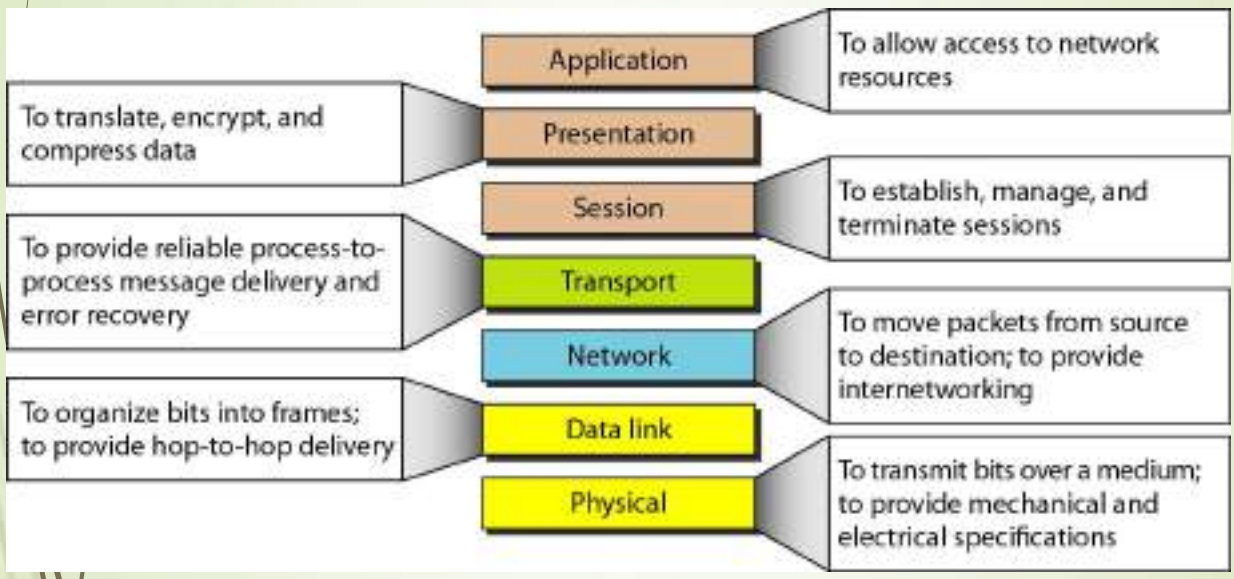
Figure 2.14 *Application layer*

18



The application layer is responsible for providing services to the user.

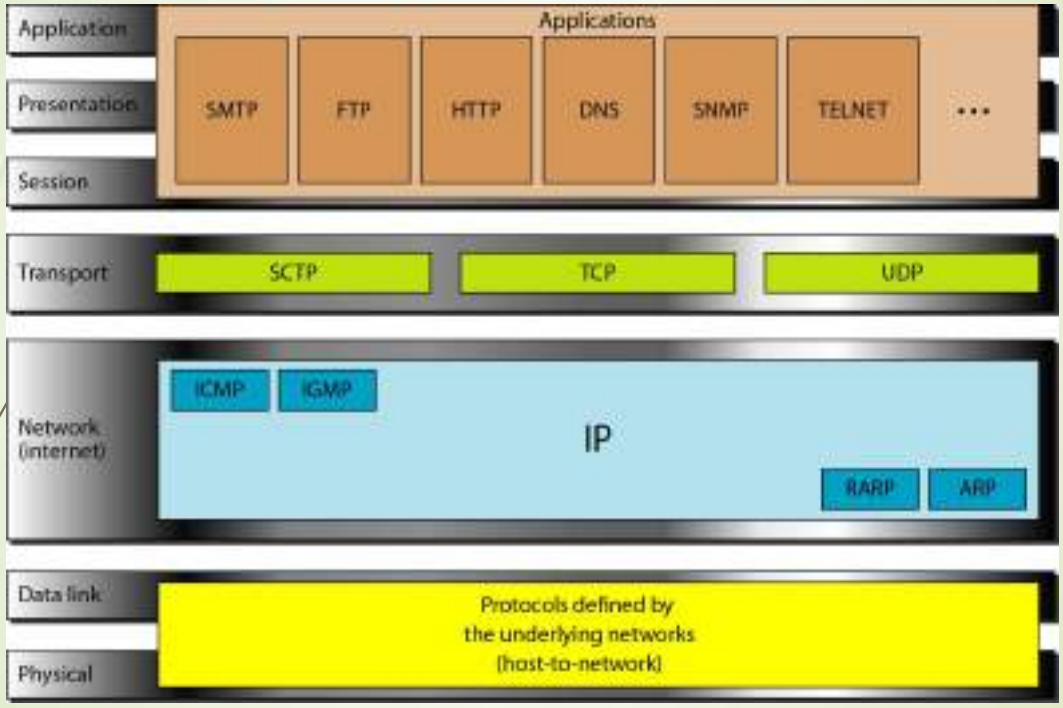
Figure 2.15 *Summary of layers*



2-4 TCP/IP PROTOCOL SUITE

The layers in the TCP/IP protocol suite do not exactly match those in the OSI model. The original TCP/IP protocol suite was defined as having four layers: host-to-network, internet, transport, and application. However, when TCP/IP is compared to OSI, we can say that the TCP/IP protocol suite is made of five layers: physical, data link, network, transport, and application.

Figure 2.16 *TCP/IP and OSI model*



2-5 ADDRESSING

Four levels of addresses are used in an internet employing the TCP/IP protocols: physical, logical, port, and specific.

Figure 2.17 *Addresses in TCP/IP*

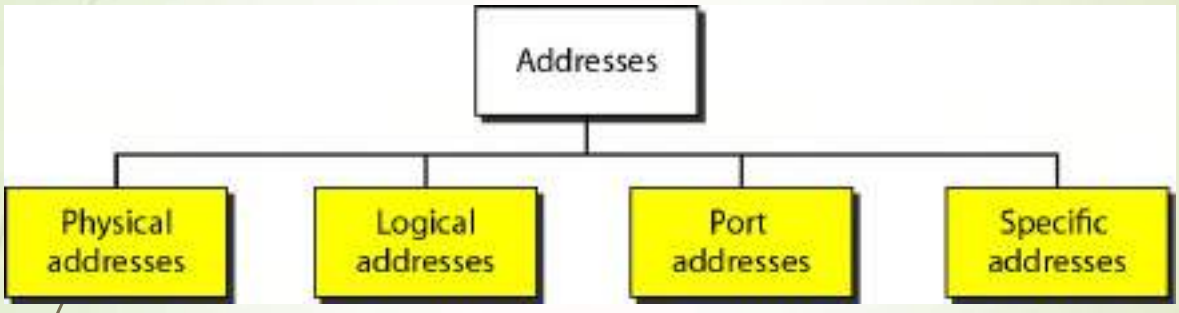
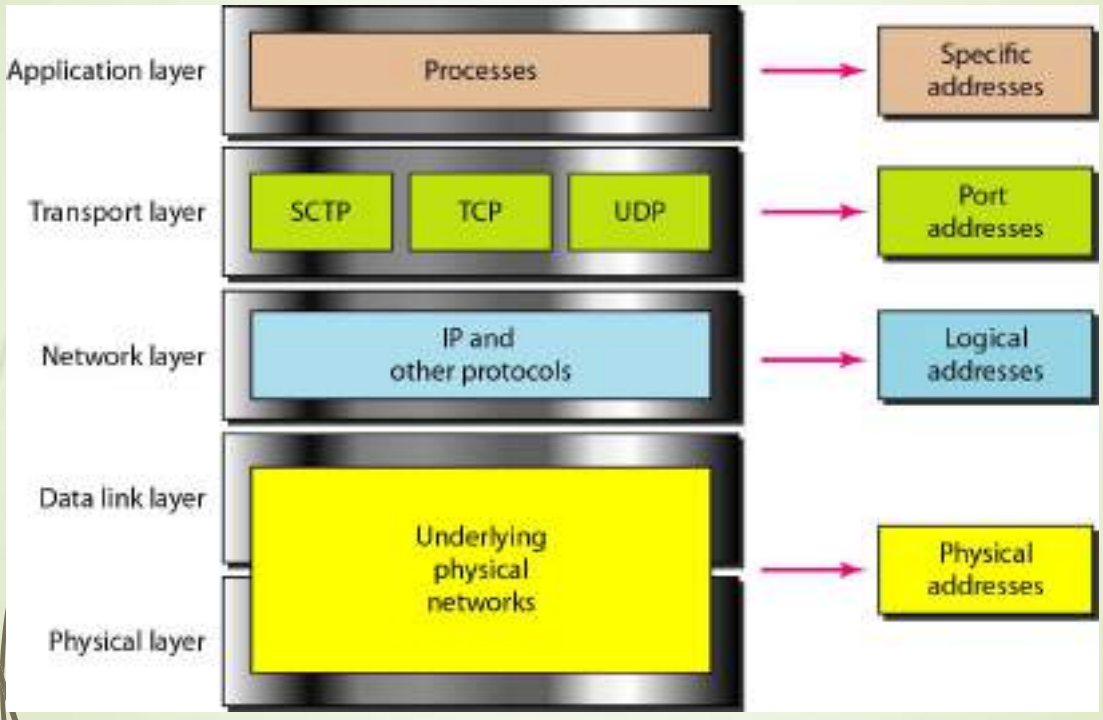


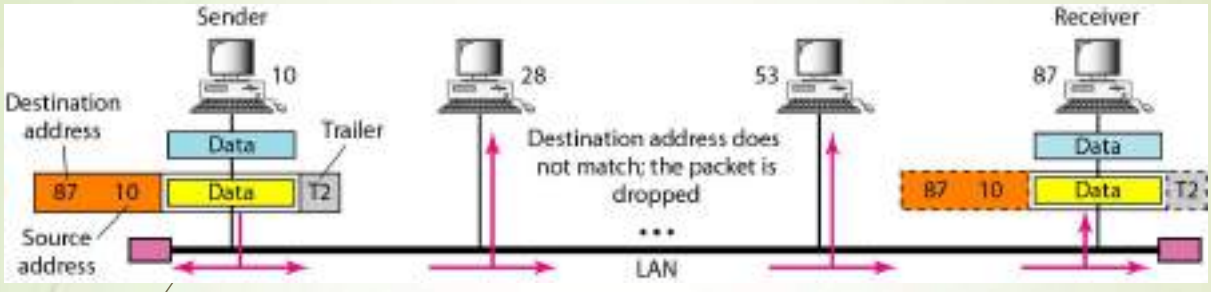
Figure 2.18 *Relationship of layers and addresses in TCP/IP*



Example 2.1

In Figure 2.19 a node with physical address 10 sends a frame to a node with physical address 87. The two nodes are connected by a link (bus topology LAN). As the figure shows, the computer with physical address 10 is the sender, and the computer with physical address 87 is the receiver.

Figure 2.19 *Physical addresses*



Example 2.2

most local-area networks use a 48-bit (6-byte) physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon, as shown below:

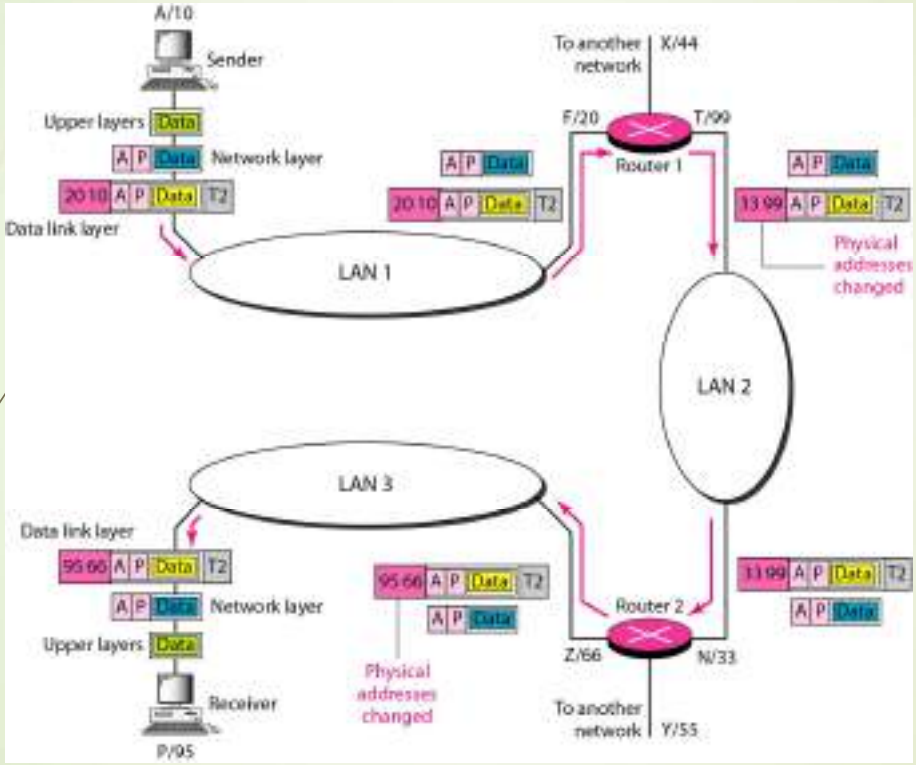
07:01:02:01:2C:4B

A 6-byte (12 hexadecimal digits) physical address.

Example 2.3

Figure 2.20 shows a part of an internet with two routers connecting three LANs. Each device (computer or router) has a pair of addresses (logical and physical) for each connection. In this case, each computer is connected to only one link and therefore has only one pair of addresses. Each router, however, is connected to three networks (only two are shown in the figure). So each router has three pairs of addresses, one for each connection.

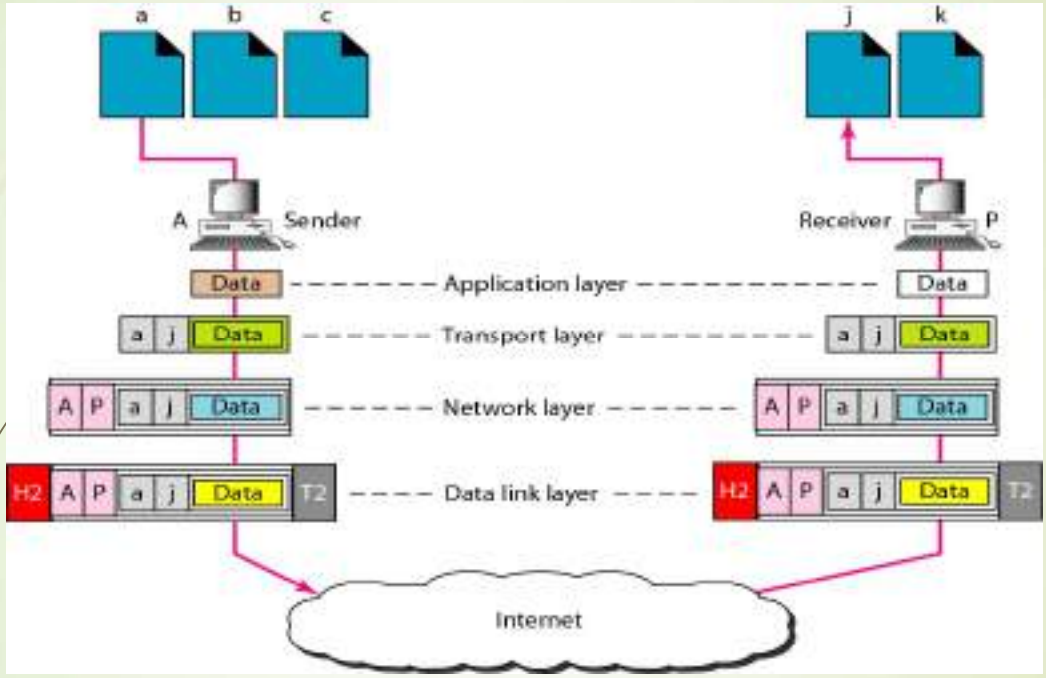
Figure 2.20 *IP addresses*



Example 2.4

Figure 2.21 shows two computers communicating via the Internet. The sending computer is running three processes at this time with port addresses a, b, and c. The receiving computer is running two processes at this time with port addresses j and k. Process a in the sending computer needs to communicate with process j in the receiving computer. Note that although physical addresses change from hop to hop, logical and port addresses remain the same from the source to destination.

Figure 2.21 *Port addresses*



The physical addresses will change from hop to hop, but the logical addresses usually remain the same.

Note

The physical addresses change from hop to hop, but the logical and port addresses usually remain the same.

3. Computer networks

Basics of network Devices

1. NIC

- Network Interface Card.
- A *hardware component* that *connects your computer* to a local data network or the *Internet.*
- A device that takes a signal from a network and converts it to a signal that a computer can understand and
- Translates computer data into electrical signals it sends through the network.
- Provides an interface onto a network (usually a LAN) for a computer system.
- A NIC is also known as a *network interface controller (NIC)*, *network interface controller card*, *expansion card*, *computer circuit board*, *network card*, *LAN card*, *network adapter* or *network adapter card* (NAC).

35

NIC

- It's a plastic circuit board about the size of a playing card.
- It has several computer chips that process signals from the network and the PC.
- The card slides into the PC's framework with a connector on the motherboard.
- A steel bracket holds the card in place.
- The bracket may have a network cable jack or an antenna.
- The bracket also has light-emitting diodes that indicate network status and activity.
- Need drivers (software code that helps to run NIC).

36

NIC Functioning

- Middleman/ Interface between your computer and the data network.
- For example,
  - when you log in to a website, the PC passes the site information to the network card, which converts the address into electrical impulses.
  - Network cables carry these impulses to a Web server somewhere on the Internet, which responds by sending a Web page back to you, once again in the form of electronic signals.
  - The card receives these signals and turns them into data that your PC displays.

NIC Types

- Work with Wi-Fi wireless networks: these cards have an antenna to send data signals via radio waves.
- Wired Ethernet connections: these cables have a rectangular plug which mates with a jack on the network card's bracket.

2. Repeater

- A repeater operates at the *physical layer.*
- Its job is to *regenerate the signal* over the same network before the signal becomes too weak or corrupted.
- It extend the length to which the signal can be transmitted over the same network.
- An important point to be noted about repeaters is that they *do not amplify the signal*.
- When the signal becomes weak, they *copy the signal bit by bit* and regenerate it at the original strength.

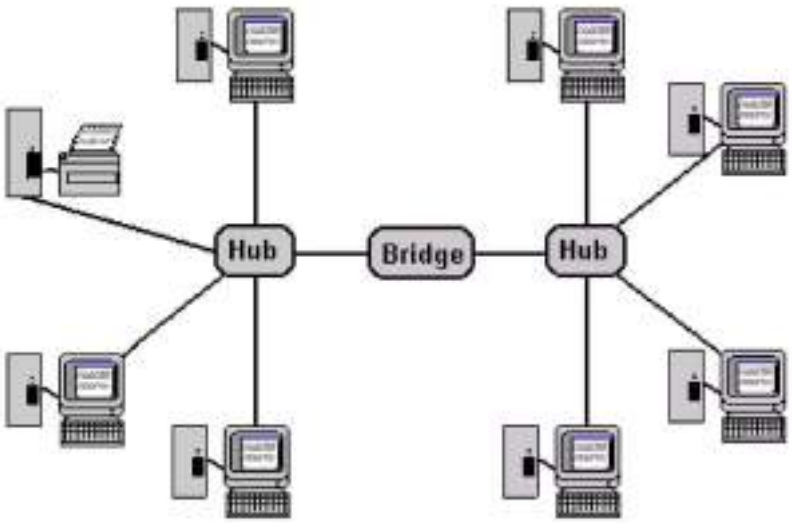
39

3. BRIDGE

- Hardware device, works at *data link layer.*
- It provides interconnection with other bridge networks that use the same protocol.
- *Connecting two different networks* together and providing communication between them.
- Bridges are similar to repeaters and hubs in that they broadcast data to every node.
- Bridges *maintain the media access control (MAC) address table* as soon as they discover new segments, so subsequent transmissions are sent to only to the desired recipient.

40

BRIDGE



41

How BRIDGEs Work

- A bridge uses a database to discover where to pass, transmit or discard the data frame.
- If the frame received by the bridge is meant for a segment that resides on the same host network, it will pass the frame to that node and the receiving bridge will then discard it.
- If the bridge receives a frame whose node MAC address is of the connected network, it will forward the frame toward it.

42

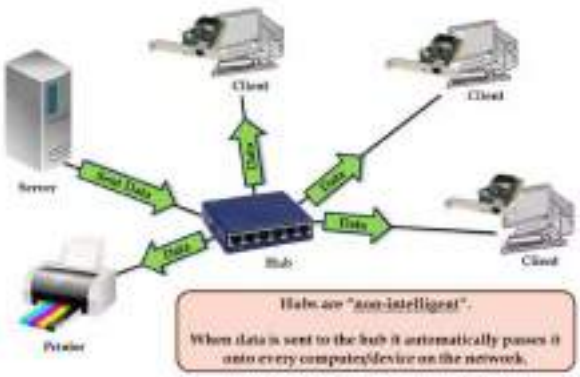
4. HUB

- A *Hardware device, physical layer device.*
- A *common connection point* for devices in a network – *connects multiple computers* or other network devices together.
- Hubs connect segments of a LAN.
- Work as repeaters.
- It has *no routing tables* or intelligence on where to send information – *broadcasts* all network data across each connection.

HUB

- *In the past*, network hubs were popular because *they were cheaper* than a switch or router.
- Today, switches do not cost much more than a hub and are a much better solution for any network.

HUB



45

How Network HUB Works

- A hub is an inexpensive way to connect devices on a network.
- Data travels around a network in 'packets' and a hub forwards these data packets out to all the devices connected to its ports.
- As a hub distributes packets to every device on the network, when a packet is destined for only one device, every other device connected to the hub receives that packet.
- Sometimes slow down the network because of traffic collision.

46

5. SWICTH

- *Hardware device*.
- A *high-speed device* that receives incoming data packets and redirects them to their destination on a local area network (LAN).
- A LAN switch operates at the *data link layer* or the *network layer* of the OSI Model.
- A switch, however, keeps a *record* of the *MAC addresses* of all the devices connected to it.
- Switches also run in *full duplex mode*.

47

SWICTH

48

How SWICTH Work

- Reads incoming TCP/IP data packets/frames containing destination information as they pass into one or more input ports.
- The destination information in the packets is used to determine which output ports will be used to send the data on to its intended destination.
- Node-to-node communication in the same network.

Similarities between Switches and HUBs

- Switches are similar to hubs, but smarter than hubs.
- A hub simply connects all the nodes on the network – communication in broadcast – resulting in many collisions.
- A switch, on the other hand, creates an electronic tunnel between source and destination ports – no other traffic can enter.
- This results in communication without collisions.

6. Routers

- Network Layer device
- A router is a device like a switch that routes data packets based on their IP addresses.
- Routers normally connect LANs and WANs together or a LAN and its ISP's network –for example, your PC and your service provider.
- Have a dynamically updating routing table based on which they make decisions on routing the data packets.

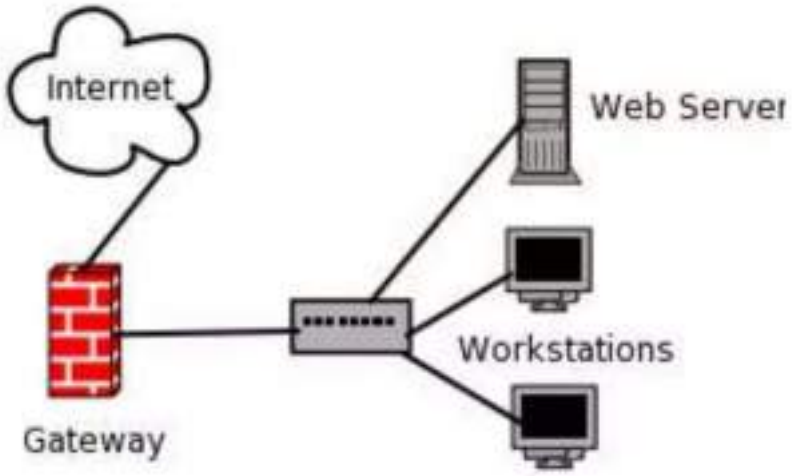
Routers

- Routers are located at gateways, the places where two or more networks connect.
- Routers use packet headers and forwarding tables to determine the best path for forwarding the packets.
- And they use protocols to communicate with each other and configure the best route between any two hosts.

7. Gateway

- A gateway, as the name suggests, is a passage to connect two networks together that may work upon different networking models.
- They basically works as the messenger agents that take data from one system, interpret it, and transfer it to another system.
- Gateways are also called protocol converters and can operate at any network layer.
- Gateways are generally more complex than switch or router.

Gateway



Gateway

- Gateways serve as the entry and exit point of a network; all data routed inward or outward must first pass through and communicate with the gateway in order to use routing paths.
- Generally, a router is configured to work as a gateway device in computer networks.
- The gateway (or default gateway) is implemented at the boundary of a network to manage all the data communication that is routed internally or externally from that network.
- Besides routing packets, gateways also possess information about the host network's internal paths.

8. CSU/ DSU

- Channel Service Unit/Data Service Unit.
- Is a hardware device about the size of an external modem.
- It converts a digital data frame from the communications technology used on a local area network (LAN) into a frame appropriate to a wide-area network (WAN) and vice versa.

CSU / DSU

- For Example: If have leased a digital line to a phone company or a gateway at an Internet service provider, you have a CSU/DSU at your end and the phone company or gateway host has a CSU/DSU at its end.
- The Channel Service Unit (CSU) receives and transmits signals from and to the WAN line and provides a barrier for electrical interference from either side of the unit.

57

9. MODEM

- Modem is short for *Modulator / Demodulator*.
- It is a hardware component that allows a computer or other device, such as a router or switch, to connect to the Internet.
- It *converts or modulates* an *analog signal* from a telephone or cable wire to a *digital signal* that a computer can recognize.
- Similarly, it converts outgoing digital data from a computer or other device to an analog signal.

58

MODEM

- The first modems were *dial-up* meaning they had to dial a phone number to connect to an *ISP*.
- These modems operated over standard analog phone lines and used the same frequencies as telephone calls, which limited their maximum data transfer rate to 56 Kbps.
- Dial-up modems also required full use of the local telephone line, meaning voice calls would interrupt the Internet connection.

MODEM

- Modern modems are typically DSL or cable modems, which are considered *broadband* devices.
- DSL modems operate over standard telephone lines, but use a wider frequency range.
- This allows for higher data transfer rates than dial-up modems and enables them to not interfere with phone calls.

1

IP Addresses

Objectives

Upon completion you will be able to:

- *Understand IPv4 addresses and classes*
- *Identify the class of an IP address*
- *Find the network address given an IP address*
- *Understand masks and how to use them*
- *Understand subnets and supernets*

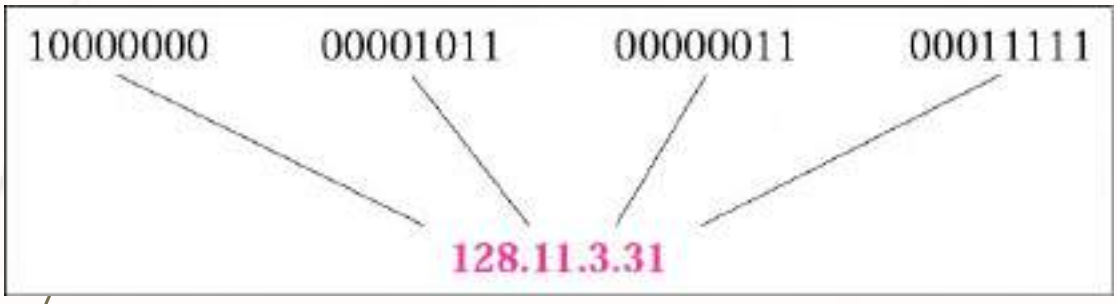
2

INTRODUCTION

The identifier used in the IP layer of the TCP/IP protocol suite to identify each device connected to the Internet is called the Internet address or IP address. An IP address is a 32-bit address that uniquely and universally defines the connection of a host or a router to the Internet. IP addresses are unique. They are unique in the sense that each address defines one, and only one, connection to the Internet. Two devices on the Internet can never have the same address.

The address space of IPv4 is
2^{32} or 4,294,967,296.

Figure 4.1 *Dotted-decimal notation*



EXAMPLE 1

Change the following IP addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111
b. 11000001 10000011 00011011 11111111
c. 11100111 11011011 10001011 01101111
d. 11111001 10011011 11111011 00001111

Solution

We replace each group of 8 bits with its equivalent decimal number (see Appendix B) and add dots for separation:

a. 129.11.11.239 *b. 193.131.27.255*
c. 231.219.139.111 *d. 249.155.251.15*

EXAMPLE 2

Change the following IP addresses from dotted-decimal notation to binary notation.

a. 111.56.45.78
b. 221.34.7.82
c. 241.8.56.12
d. 75.45.34.78

Solution

We replace each decimal number with its binary equivalent:

a. 01101111 00111000 00101101 01001110
b. 11011101 00100010 00000111 01010010
c. 11110001 00001000 00111000 00001100
d. 01001011 00101101 00100010 01001110

EXAMPLE 3

Find the error, if any, in the following IP addresses:

a. 111.56.045.78
b. 221.34.7.8.20
c. 75.45.301.14
d. 11100010.23.14.67

Solution

a. There are no leading zeroes in dotted-decimal notation (045).

b. We may not have more than four numbers in an IP address.

c. In dotted-decimal notation, each number is less than or equal to 255; 301 is outside this range.

d. A mixture of binary notation and dotted-decimal notation is not allowed.

7

EXAMPLE 4

Change the following IP addresses from binary notation to hexadecimal notation.

a. 10000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

Solution

We replace each group of 4 bits with its hexadecimal equivalent (see Appendix B). Note that hexadecimal notation normally has no added spaces or dots; however, 0X (or 0x) is added at the beginning or the subscript 16 at the end to show that the number is in hexadecimal.

a. 0X810B0BEF or $810B0BEF_{16}$
b. 0XC1831BFF or $C1831BFF_{16}$

8

CLASSFUL ADDRESSING

IP addresses, when started a few decades ago, used the concept of classes. This architecture is called classful addressing. In the mid-1990s, a new architecture, called classless addressing, was introduced and will eventually supersede the original architecture. However, part of the Internet is still using classful addressing, but the migration is very fast.

The topics discussed in this section include:

Recognizing Classes
Netid and Hostid
Classes and Blocks
Network Addresses
Sufficient Information
Mask
CIDR Notation
Address Depletion

Figure 4.2 *Occupation of the address space*

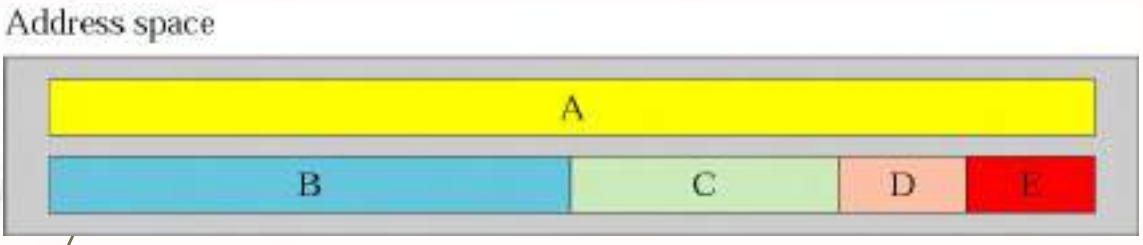


Table 4.1 Addresses per class

Class	*Number of Addresses*	*Percentage*
A	$2^{31} = 2,147,483,648$	50%
B	$2^{30} = 1,073,741,824$	25%
C	$2^{29} = 536,870,912$	12.5%
D	$2^{28} = 268,435,456$	6.25%
E	$2^{28} = 268,435,456$	6.25%

Figure 4.3 *Finding the class in binary notation*

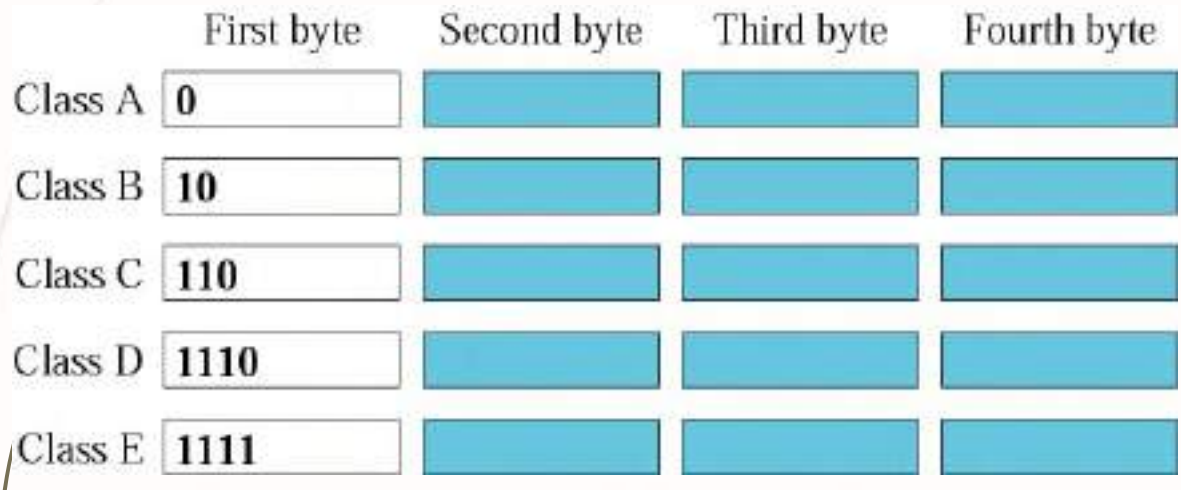
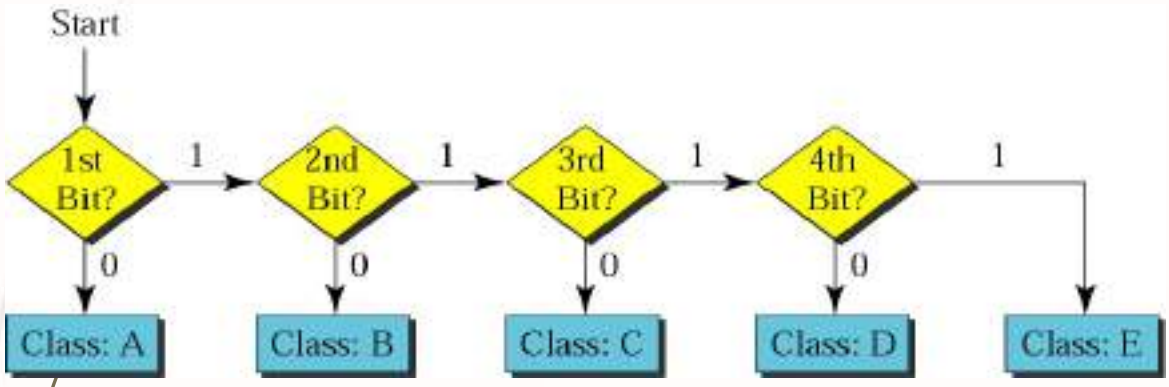


Figure 4.4 *Finding the address class*



13

EXAMPLE 5

How can we prove that we have 2,147,483,648 addresses in class A?

Solution

In class A, only 1 bit defines the class. The remaining 31 bits are available for the address. With 31 bits, we can have 2^{31} or 2,147,483,648 addresses.

14

EXAMPLE 6

Find the class of each address:

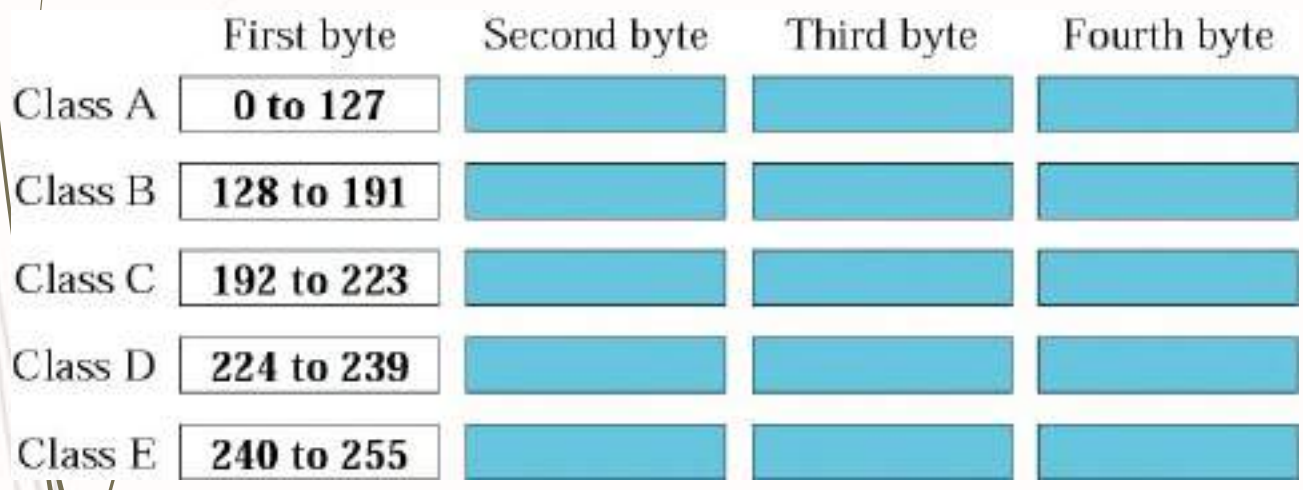
a. 00000001 00001011 00001011 11101111
b. 11000001 10000011 00011011 11111111
c. 10100111 11011011 10001011 01101111
d. 11110011 10011011 11111011 00001111

Solution

See the procedure in Figure 4.4.

a. The first bit is 0. This is a class A address.
b. The first 2 bits are 1; the third bit is 0. This is a class C address.
c. The first bit is 0; the second bit is 1. This is a class B address.
d. The first 4 bits are 1s. This is a class E address..

Figure 4.5 *Finding the class in decimal notation*



EXAMPLE 7

Find the class of each address:

a. 227.12.14.87 *b.193.14.56.22* *c.14.23.120.8*
d. 252.5.15.111 *e.134.11.78.56*

Solution

a. The first byte is 227 (between 224 and 239); the class is D.
b. The first byte is 193 (between 192 and 223); the class is C.
c. The first byte is 14 (between 0 and 127); the class is A.
d. The first byte is 252 (between 240 and 255); the class is E.
e. The first byte is 134 (between 128 and 191); the class is B.

EXAMPLE 8

In Example 5 we showed that class A has 2^{31} (2,147,483,648) addresses. How can we prove this same fact using dotted-decimal notation?

Solution

The addresses in class A range from 0.0.0.0 to 127.255.255.255. We need to show that the difference between these two numbers is 2,147,483,648. This is a good exercise because it shows us how to define the range of addresses between two addresses. We notice that we are dealing with base 256 numbers here. Each byte in the notation has a weight. The weights are as follows:

See Next Slide

EXAMPLE 8 (CONTINUED)

$$256^3, 256^2, 256^1, 256^0$$

Now to find the integer value of each number, we multiply each byte by its weight:

$$\text{Last address: } 127 \times 256^3 + 255 \times 256^2 + 255 \times 256^1 + 255 \times 256^0 = 2{,}147{,}483{,}647$$

$$\text{First address: } = 0$$

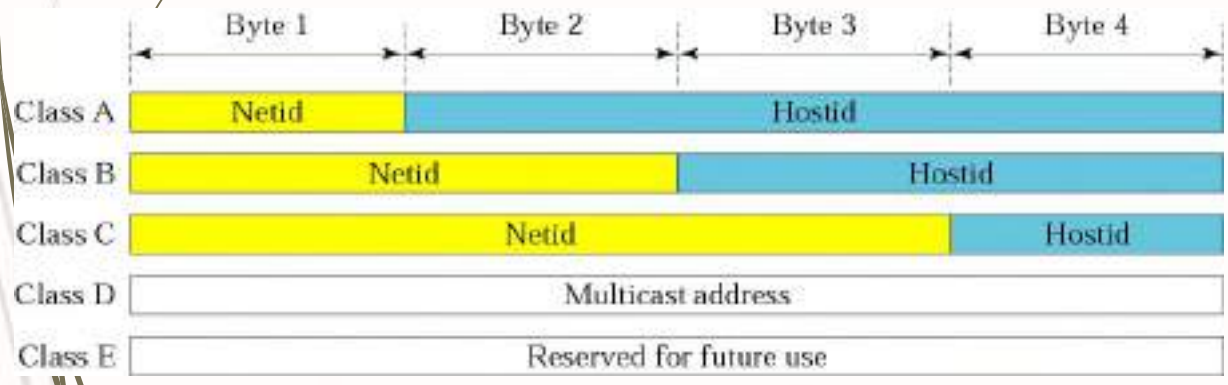
If we subtract the first from the last and add 1 to the result (remember we always add 1 to get the range), we get 2,147,483,648 or 2^{31}.

19

Netid and Hostid

In classful addressing, an IP address in classes A, B, and C is divided into netid and hostid. These parts are of varying lengths, depending on the class of the address

Figure 4.6 *Netid and hostid*



20

Classes and Blocks

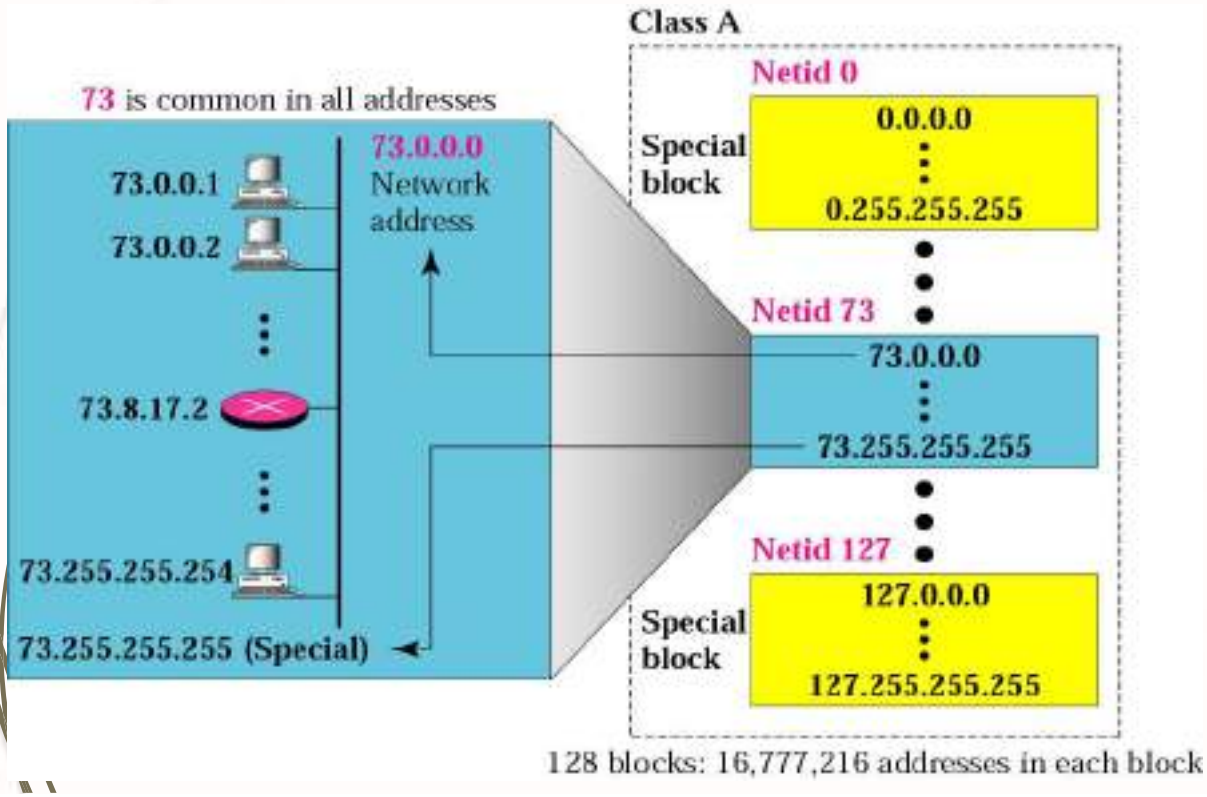
Class A

Since only 1 byte in class A defines the netid and the leftmost bit should be 0, the next 7 bits can be changed to find the number of blocks in this class. Therefore, class A is divided into 27 = 128 blocks that can be assigned to 128 organizations (the number is less because some blocks were reserved as special blocks). However, each block in this class contains 16,777,216 addresses, which means the organization should be a really large one to use all these addresses. Many addresses are wasted in this class. Figure shows the block in class A.

Note:

Millions of class A addresses are wasted.

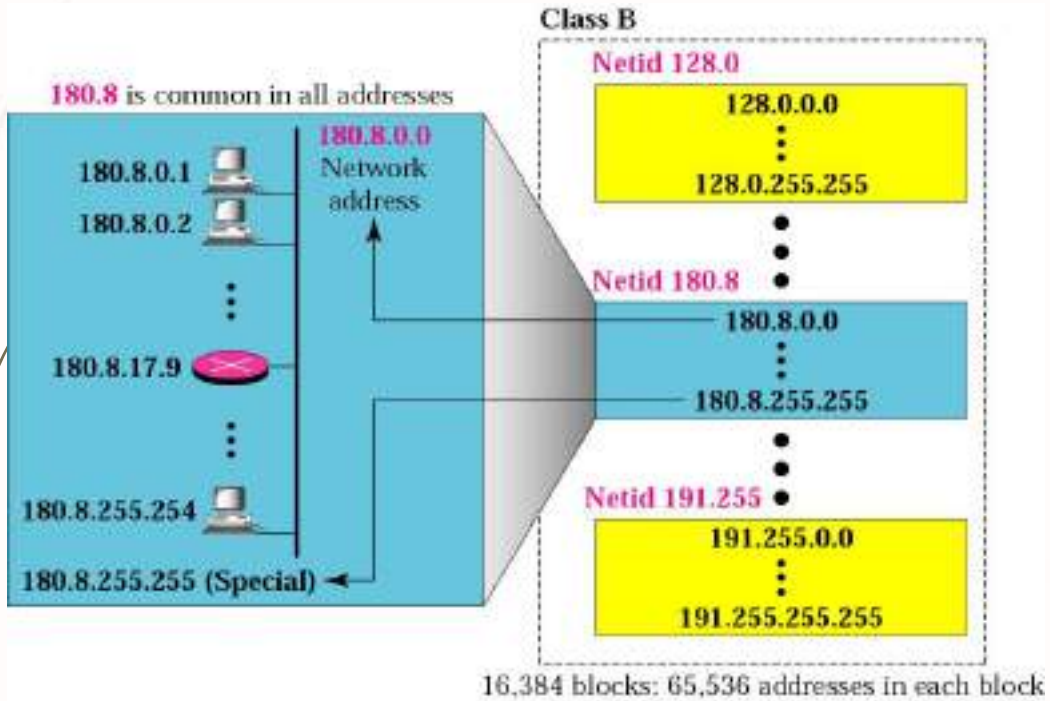
Figure 4.7 *Blocks in class A*



Class B

Since 2 bytes in class B define the class and the two leftmost bit should be 10 (fixed), the next 14 bits can be changed to find the number of blocks in this class. Therefore, class B is divided into 214 = 16,384 blocks that can be assigned to 16,384 organizations (the number is less because some blocks were reserved as special blocks). However, each block in this class contains 65,536 addresses. Not so many organizations can use so many addresses. Many addresses are wasted in this class. Figure shows the blocks in class B.

Figure 4.8 *Blocks in class B*

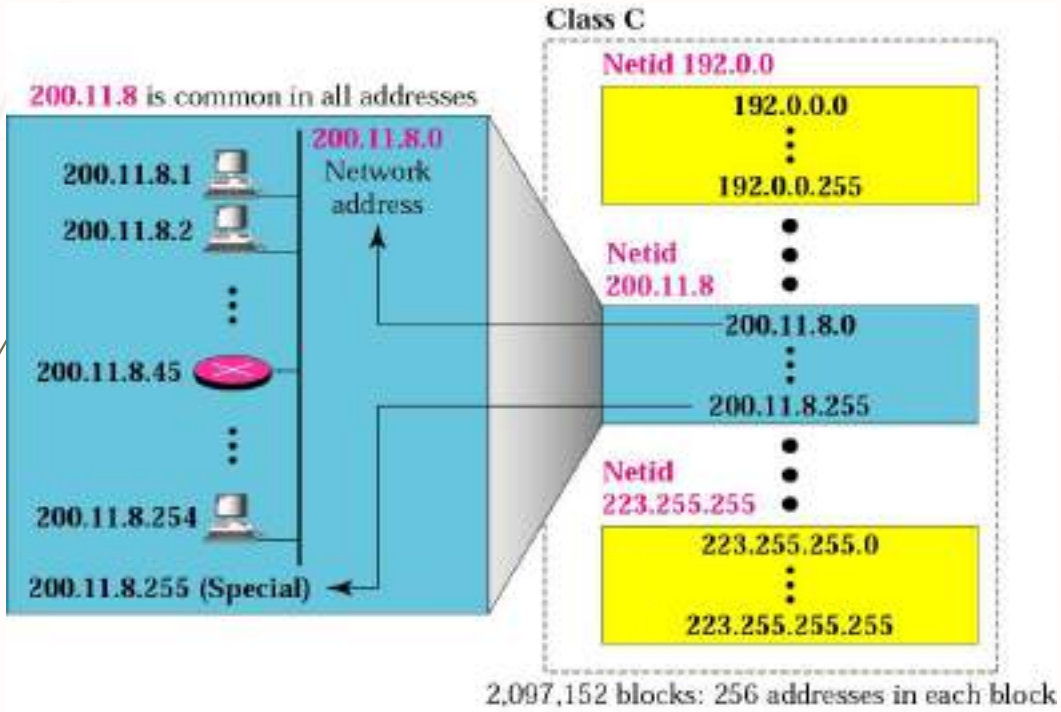


Many class B addresses are wasted.

Class C

Since 3 bytes in class C define the class and the three leftmost bits should be 110 (fixed), the next 21 bits can be changed to find the number of blocks in this class. Therefore, class C is divided into 221 = 2,097,152 blocks, in which each block contains 256 addresses, that can be assigned to 2,097,152 organizations (the number is less because some blocks were reserved as special blocks). Each block contains 256 addresses. However, not so many organizations were so small as to be satisfied with a class C block. Figure shows the blocks in class C.

Figure 4.9 *Blocks in class C*



Note:

The number of addresses in class C is smaller than the needs of most organizations.

Note:

Class D addresses are used for multicasting; there is only one block in this class.

Note:

Class E addresses are reserved for future purposes; most of the block is wasted.

Note:

In classful addressing, the network address (the first address in the block) is the one that is assigned to the organization. The range of addresses can automatically be inferred from the network address.

EXAMPLE 9

Given the network address 17.0.0.0, find the class, the block, and the range of the addresses.

Solution

The class is A because the first byte is between 0 and 127. The block has a netid of 17. The addresses range from 17.0.0.0 to 17.255.255.255.

EXAMPLE 10

Given the network address 132.21.0.0, find the class, the block, and the range of the addresses.

Solution

The class is B because the first byte is between 128 and 191. The block has a netid of 132.21. The addresses range from 132.21.0.0 to 132.21.255.255.

EXAMPLE 11

Given the network address 220.34.76.0, find the class, the block, and the range of the addresses.

Solution

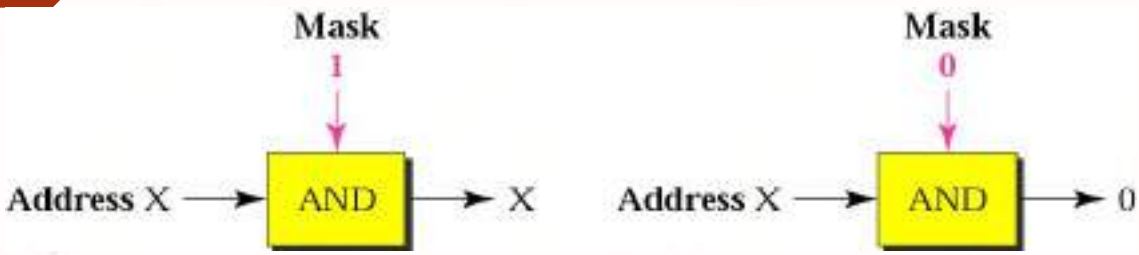
The class is C because the first byte is between 192 and 223. The block has a netid of 220.34.76. The addresses range from 220.34.76.0 to 220.34.76.255.

35

Masking concept

36

Figure 4.11 AND operation



Class	*Mask in binary*	*Mask in dotted-decimal*
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0

Note:

The network address is the beginning address of each block. It can be found by applying the default mask to any of the addresses in the block (including itself). It retains the netid of the block and sets the hostid to zero.

EXAMPLE 12

Given the address 23.56.7.91, find the beginning address (network address).

Solution

The default mask is 255.0.0.0, which means that only the first byte is preserved and the other 3 bytes are set to 0s. The network address is 23.0.0.0.

39

EXAMPLE 13

Given the address 132.6.17.85, find the beginning address (network address).

Solution

The default mask is 255.255.0.0, which means that the first 2 bytes are preserved and the other 2 bytes are set to 0s. The network address is 132.6.0.0.

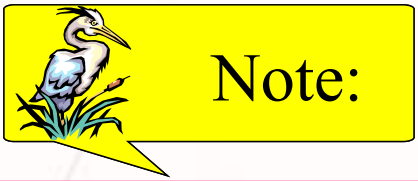
40

EXAMPLE 14

Given the address 201.180.56.5, find the beginning address (network address).

Solution

The default mask is 255.255.255.0, which means that the first 3 bytes are preserved and the last byte is set to 0. The network address is 201.180.56.0.



IP addresses are designed with two levels of hierarchy.

Figure 4.20 A network with two levels of hierarchy (not subnetted)

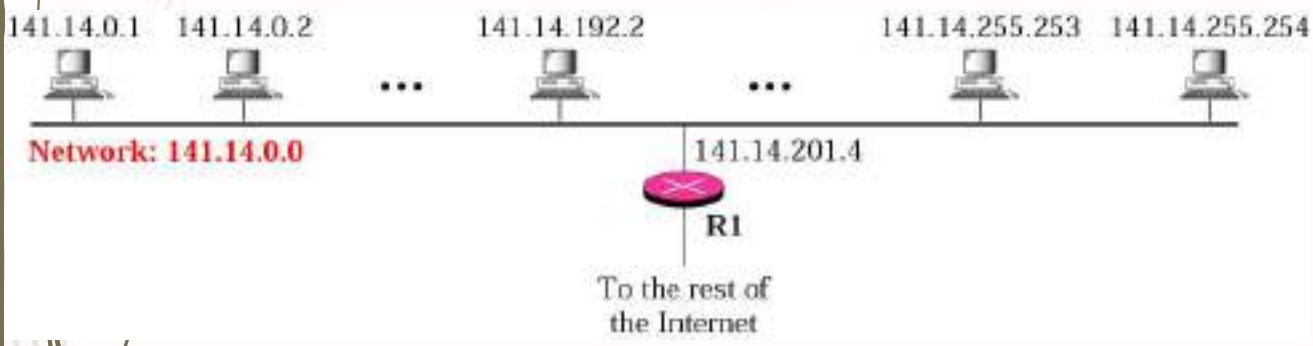


Figure 4.21 *A network with three levels of hierarchy (subnetted)*

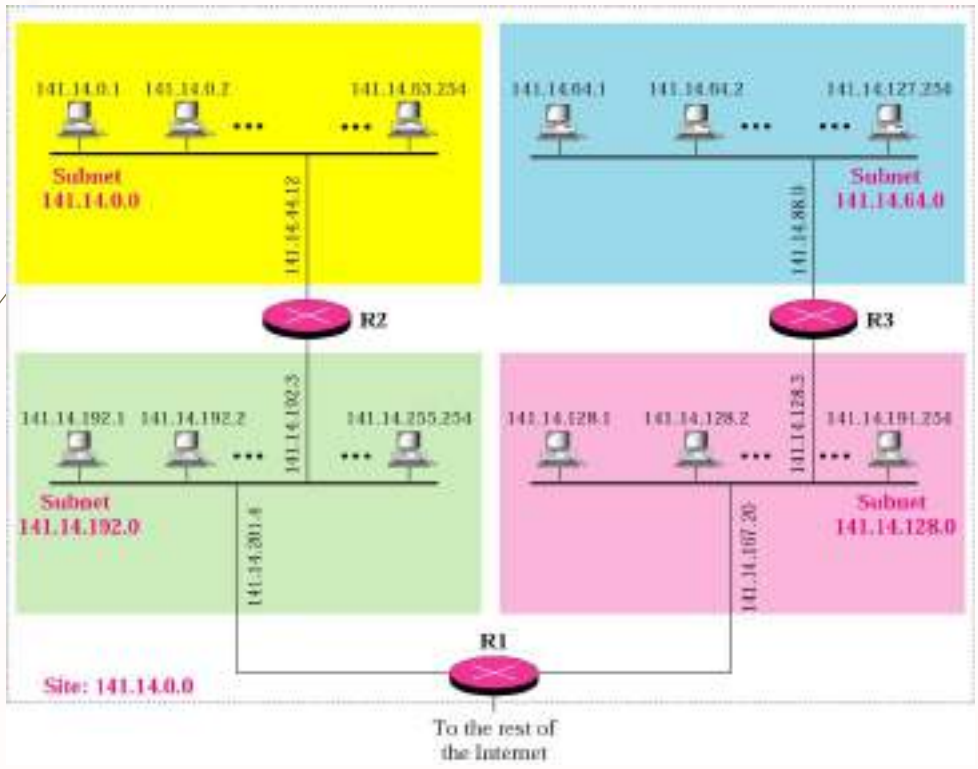


Figure 4.22 *Addresses in a network with and without subnetting*

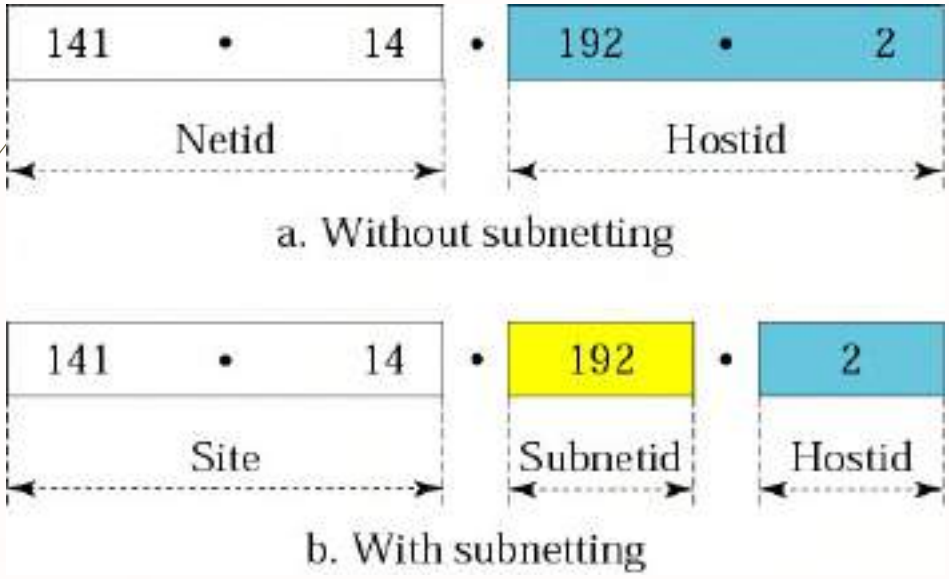
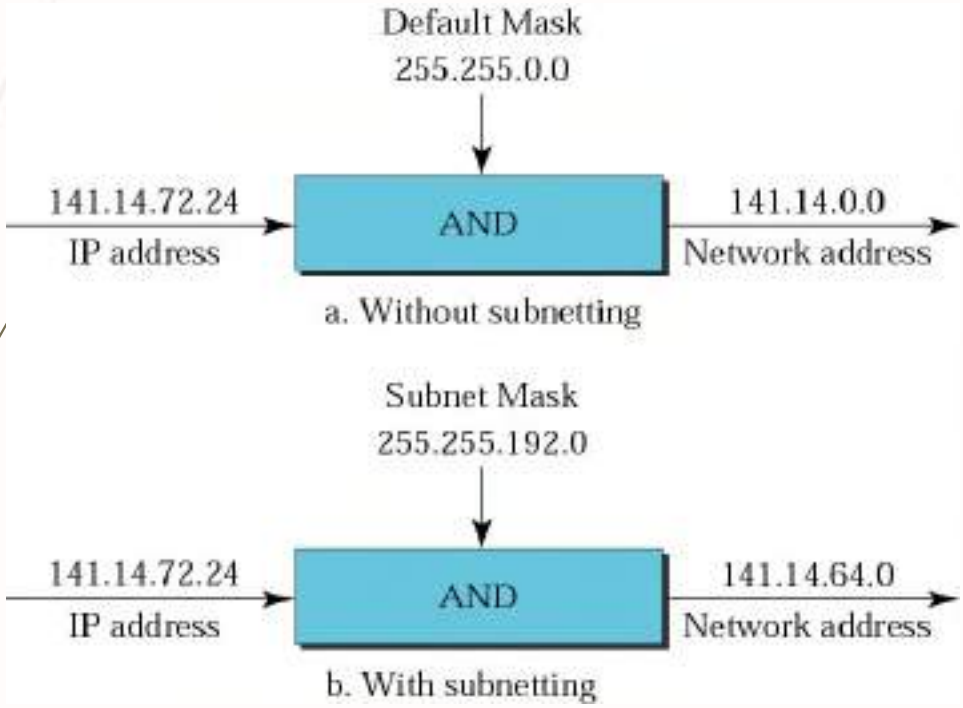


Figure 4.24 *Default mask and subnet mask*



EXAMPLE 15

What is the subnetwork address if the destination address is 200.45.34.56 and the subnet mask is 255.255.240.0?

Solution

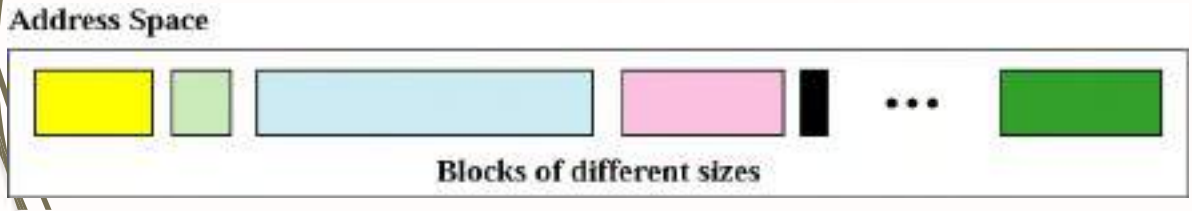
We apply the AND operation on the address and the subnet mask.

Address	→	11001000 00101101 00100010 00111000
Subnet Mask	→	11111111 11111111 11110000 00000000
Subnetwork Address	→	11001000 00101101 00100000 00000000.

IP Addresses: Classless Addressing

47

Subnetting in classful addressing did not really solve the address depletion problem and made the distribution of addresses and the routing process more difficult. With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution. The larger address space, however, requires that the length of IP addresses to be increased, which means the format of the IP packets needs to be changed. Although the long-range solution has already been devised and is called IPv6, a short-term solution was also devised to use the same address space but to change the distribution of addresses to provide a fair share to each organization. The short-term solution still uses IPv4 addresses, but it is called classless addressing. In other words, the class privilege was removed from the distribution to compensate for the address depletion.



48

EXAMPLE 1

Which of the following can be the beginning address of a block that contains 16 addresses?

a. 205.16.37.32 *b.190.16.42.44*
c. 17.17.33.80 *d.123.45.24.52*

Solution

Only two are eligible (a and c). The address 205.16.37.32 is eligible because 32 is divisible by 16. The address 17.17.33.80 is eligible because 80 is divisible by 16.

EXAMPLE 2

Which of the following can be the beginning address of a block that contains 256 addresses?

a.205.16.37.32 *b.190.16.42.0*
c.17.17.32.0 *d.123.45.24.52*

Solution

In this case, the right-most byte must be 0. the IP addresses use base 256 arithmetic. When the right-most byte is 0, the total address is divisible by 256. Only two addresses are eligible (b and c).

EXAMPLE 3

Which of the following can be the beginning address of a block that contains 1024 addresses?

a. 205.16.37.32 *b.190.16.42.0*
c. 17.17.32.0 *d.123.45.24.52*

Solution

In this case, we need to check two bytes because 1024 = 4 × 256. The right-most byte must be divisible by 256. The second byte (from the right) must be divisible by 4. Only one address is eligible (c).

Figure 5.2 *Format of classless addressing address*

Table 5.1 *Prefix lengths*

/n	Mask	/n	Mask	/n	Mask	/n	Mask
/1	128.0.0.0	/9	255.128.0.0	/17	255.255.128.0	/25	255.255.255.128
/2	192.0.0.0	/10	255.192.0.0	/18	255.255.192.0	/26	255.255.255.192
/3	224.0.0.0	/11	255.224.0.0	/19	255.255.224.0	/27	255.255.255.224
/4	240.0.0.0	/12	255.240.0.0	/20	255.255.240.0	/28	255.255.255.240
/5	248.0.0.0	/13	255.248.0.0	/21	255.255.248.0	/29	255.255.255.248
/6	252.0.0.0	/14	255.252.0.0	/22	255.255.252.0	/30	255.255.255.252
/7	254.0.0.0	/15	255.254.0.0	/23	255.255.254.0	/31	255.255.255.254
/8	255.0.0.0	/16	255.255.0.0	/24	255.255.255.0	/32	255.255.255.255

The slash notation is formally referred to as classless interdomain routing or CIDR notation.

EXAMPLE 4

What is the first address in the block if one of the addresses is 167.199.170.82/27?

Solution

The prefix length is 27, which means that we must keep the first 27 bits as is and change the remaining bits (5) to 0s. The following shows the process:

Address in binary:	10100111 11000111 10101010 01010010
Keep the left 27 bits:	10100111 11000111 10101010 01000000
Result in CIDR notation:	167.199.170.64/27

EXAMPLE 5

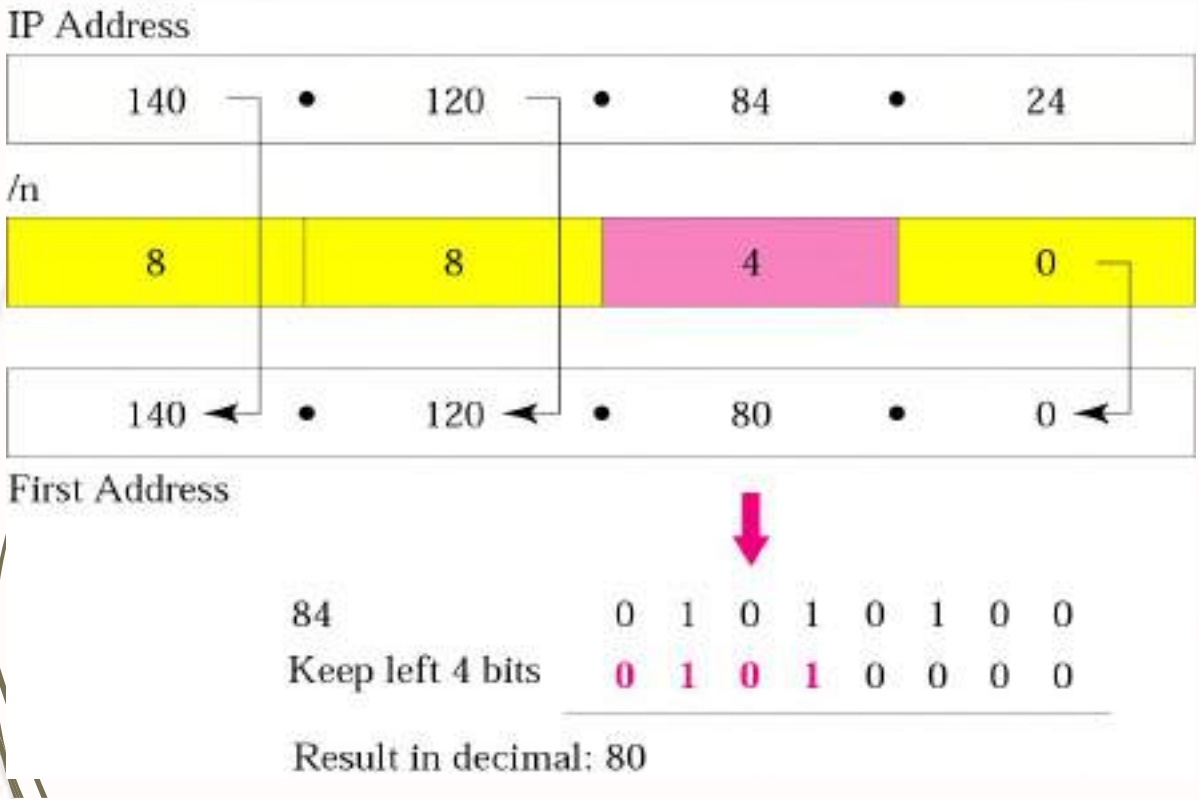
What is the first address in the block if one of the addresses is 140.120.84.24/20?

Solution

Figure 5.3 shows the solution. The first, second, and fourth bytes are easy; for the third byte we keep the bits corresponding to the number of 1s in that group. The first address is 140.120.80.0/20.

See Next Slide

Figure 5.3 *Example 5*



SUBNETTING

When an organization is granted a block of addresses, it can create subnets to meet its needs. The prefix length increases to define the subnet prefix length.

The topics discussed in this section include:

Finding the Subnet Mask
Finding the Subnet Addresses
Variable-Length Subnets

In fixed-length subnetting, the number of subnets is a power of 2.

EXAMPLE 12

An organization is granted the block 130.34.12.64/26. The organization needs 4 subnets. What is the subnet prefix length?

Solution

We need 4 subnets, which means we need to add two more 1s ($\log_2 4 = 2$) to the site prefix. The subnet prefix is then /28.

EXAMPLE 13

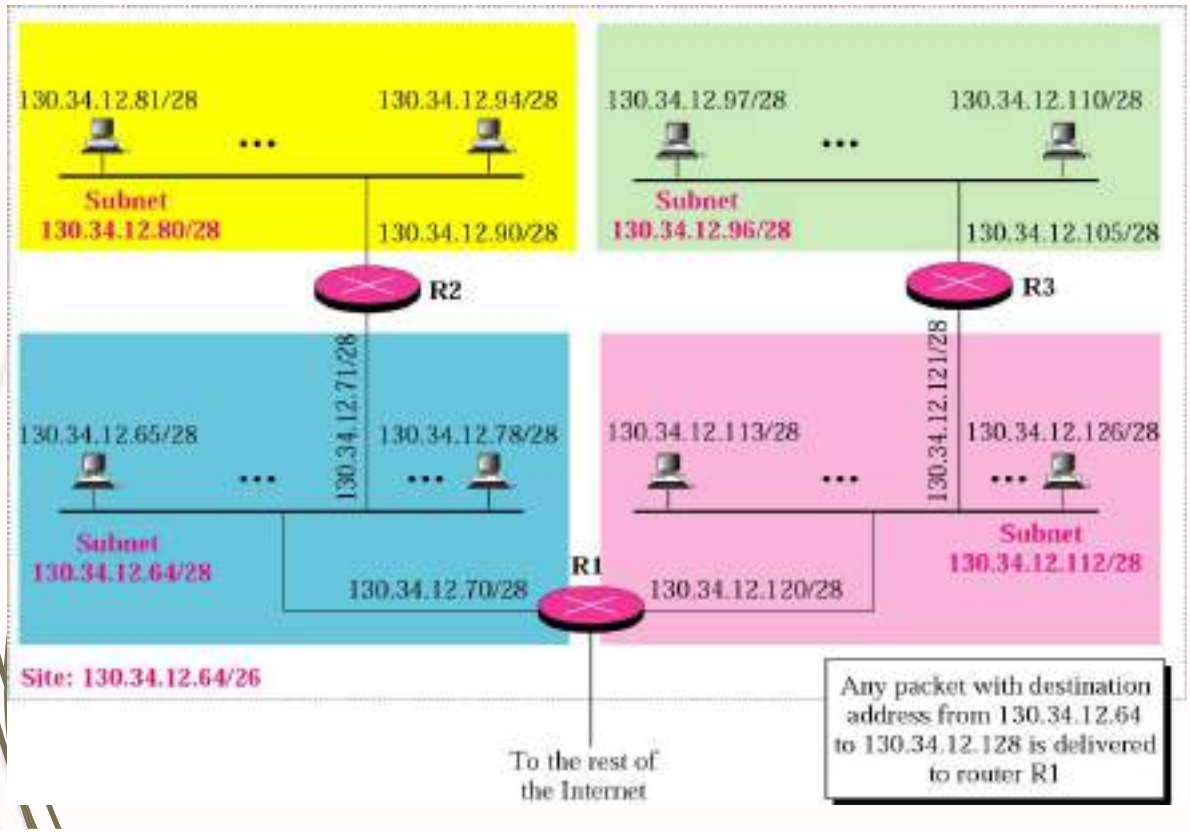
What are the subnet addresses and the range of addresses for each subnet in the previous example?

Solution

Figure 5.6 shows one configuration.

See Next Slide

Figure 5.6 *Example 13*



61

ADDRESS ALLOCATION

Address allocation is the responsibility of a global authority called the Internet Corporation for Assigned Names and Addresses (ICANN). It usually assigns a large block of addresses to an ISP to be distributed to its Internet users.

62

EXAMPLE 16

An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:

a. The first group has 64 customers; each needs 256 addresses.
b. The second group has 128 customers; each needs 128 addresses
c. The third group has 128 customers; each needs 64 addresses.

See Next Slide

EXAMPLE 16 (CONTINUED)

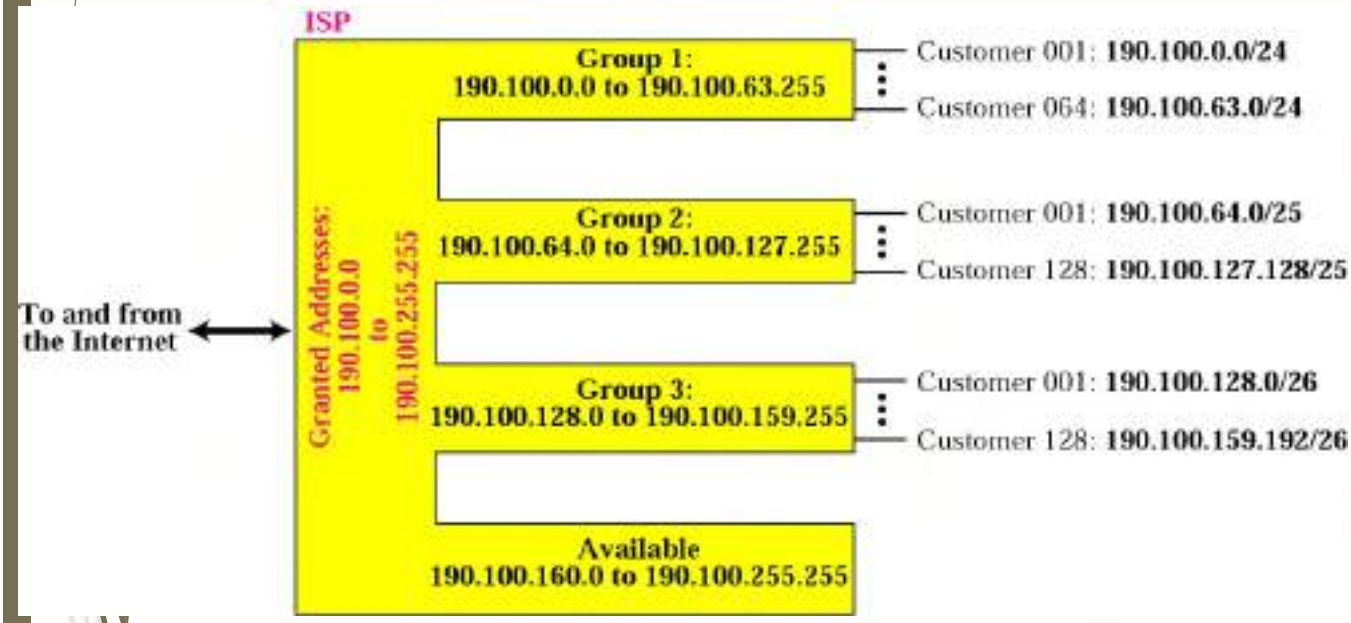
Design the subblocks and find out how many addresses are still available after these allocations.

Solution

Figure 5.9 shows the situation.

See Next Slide

Figure 5.9 *Example 16*



EXAMPLE 16 (CONTINUED)

Group 1

For this group, each customer needs 256 addresses. This means the suffix length is 8 (2^8 =256). The prefix length is then 32 − 8 = 24. The addresses are:

1st Customer	190.100.0.0/24	190.100.0.255/24
2nd Customer	190.100.1.0/24	190.100.1.255/24
. . .		
64th Customer	190.100.63.0/24	190.100.63.255/24

Total = 64 × 256 = 16,384

See Next Slide

EXAMPLE 16 (CONTINUED)

Group 2

For this group, each customer needs 128 addresses. This means the suffix length is 7 (2^7 =128). The prefix length is then 32 − 7 = 25. The addresses are:

1st Customer	190.100.64.0/25	190.100.64.127/25
2nd Customer	190.100.64.128/25	190.100.64.255/25
. . .		
128th Customer	190.100.127.128/25	190.100.127.255/25

Total = 128 × 128 = 16,384

See Next Slide

EXAMPLE 16 (CONTINUED)

Group 3

For this group, each customer needs 64 addresses. This means the suffix length is 6 (26 = 64). The prefix length is then 32 − 6 = 26. The addresses are:

1st Customer	190.100.128.0/26	190.100.128.63/26
2nd Customer	190.100.128.64/26	190.100.128.127/26
. . .		
128th Customer	190.100.159.192/26	190.100.159.255/26

Total = 128 × 64 = 8,192

See Next Slide

EXAMPLE 16 (CONTINUED)

Number of granted addresses to the ISP: 65,536

Number of allocated addresses by the ISP: 40,960

Number of available addresses: 24,576

Wired LANs: Ethernet

we mentioned that the TCP/IP protocol suite does not define any protocol for the data-link or the physical layer. In other words, TCP/IP accepts any protocol at these two layers that can provide services to the network layer. The data-link layer and the physical layer are actually the territory of the local and wide area networks.

a local area network (LAN) is a computer network that is designed for a limited geographic area such as a building or a campus. Although a LAN can be used as an isolated network to connect computers in an organization for the sole purpose of sharing resources, most LANs today are also linked to a wide area network (WAN) or the Internet.

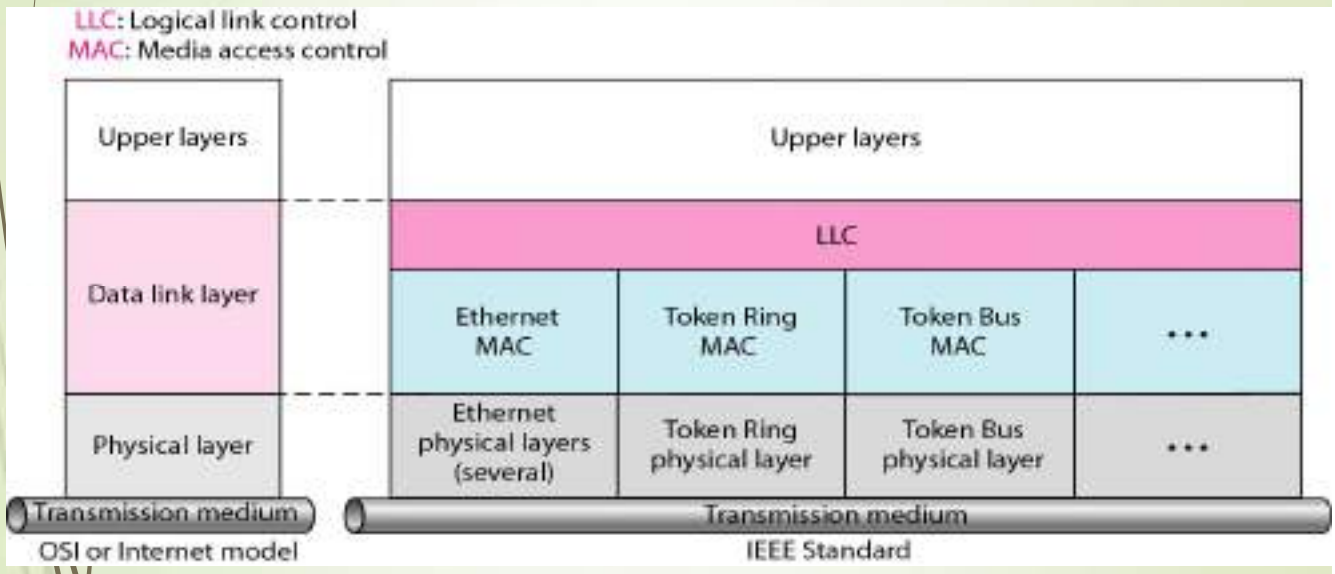
3

IEEE Project 802

IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers. Project 802 does not seek to replace any part of the OSI model or TCP/IP protocol suite. Instead, it is a way of specifying functions of the physical layer and the data-link layer of major LAN protocols.

4

The IEEE has subdivided the data-link layer into two sublayers



5

Logical Link Control (LLC)
data link control handles framing, flow control, and error control. In IEEE Project 802, flow control, error control, and part of the framing duties are collected into one sublayer called the logical link control (LLC). Framing is handled in both the LLC sublayer and the MAC sublayer

Media Access Control (MAC)
IEEE Project 802 has created a sublayer called media access control that defines the specific access method for each LAN. For example, it defines CSMA/CD as the media access method for Ethernet LANs and defines the token-passing method for Token Ring and Token Bus LANs.

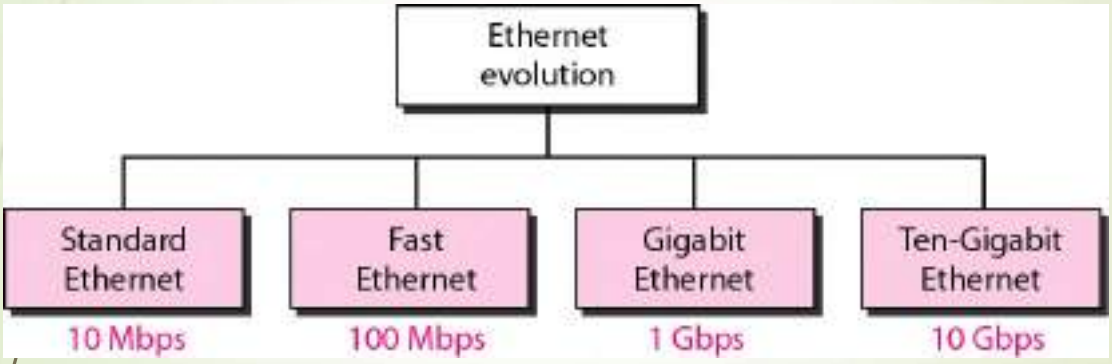
6

STANDARD ETHERNET

The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC). Since then, it has gone through four generations. We briefly discuss the Standard (or traditional) Ethernet in this section.

7

Ethernet Evolution



8

STANDARD ETHERNET

We refer to the original Ethernet technology with the data rate of 10 Mbps as the Standard Ethernet

Characteristics

Let us first discuss some characteristics of the Standard Ethernet.

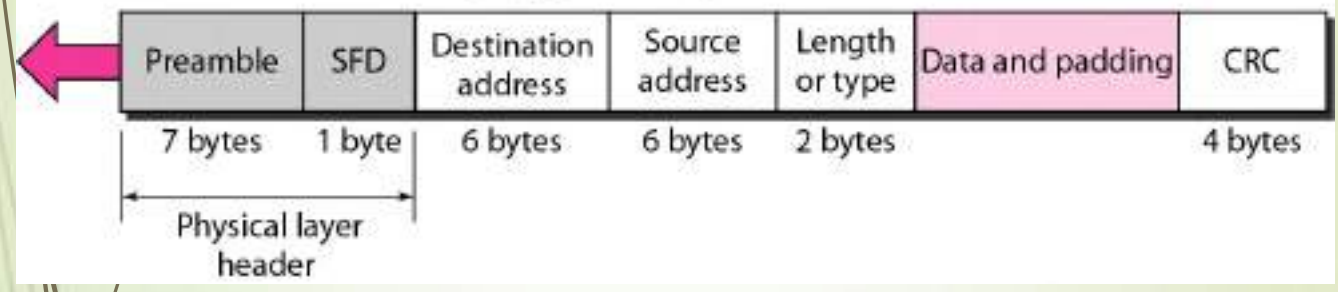
1. Connectionless and Unreliable Service

9

2. Frame Format

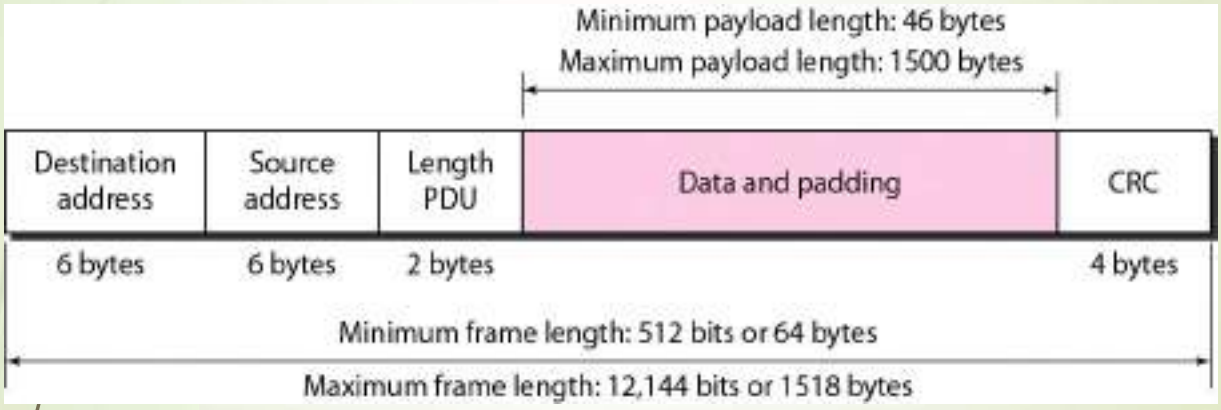
Preamble: 56 bits of alternating 1s and 0s.

SFD: Start frame delimiter, flag (10101011)

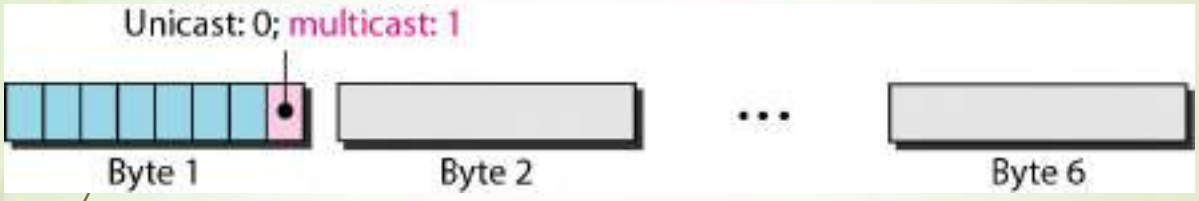


10

3. Frame Length



4. Unicast and multicast addresses



Note

The least significant bit of the first byte defines the type of address. If the bit is 0, the address is unicast; otherwise, it is multicast.

The broadcast destination address is a special case of the multicast address in which all bits are 1s.

Example

Define the type of the following destination addresses:

a. 4A:30:10:21:10:1A b. 47:20:1B:2E:08:EE

c. FF:FF:FF:FF:FF:FF

Solution

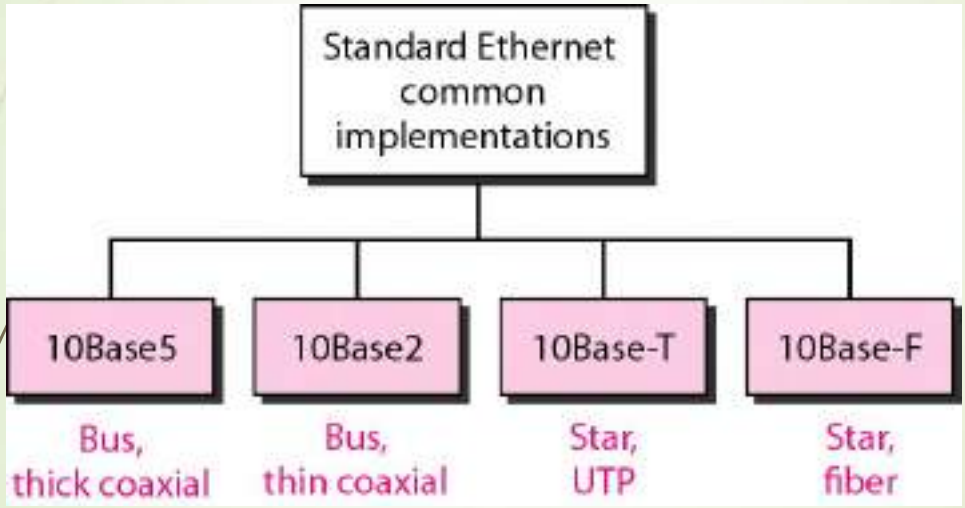
To find the type of the address, we need to look at the second hexadecimal digit from the left. If it is even, the address is unicast. If it is odd, the address is multicast. If all digits are F's, the address is broadcast. Therefore, we have the following:

a. This is a unicast address because A in binary is 1010.

b. This is a multicast address because 7 in binary is 0111.

c. This is a broadcast address because all digits are F's.

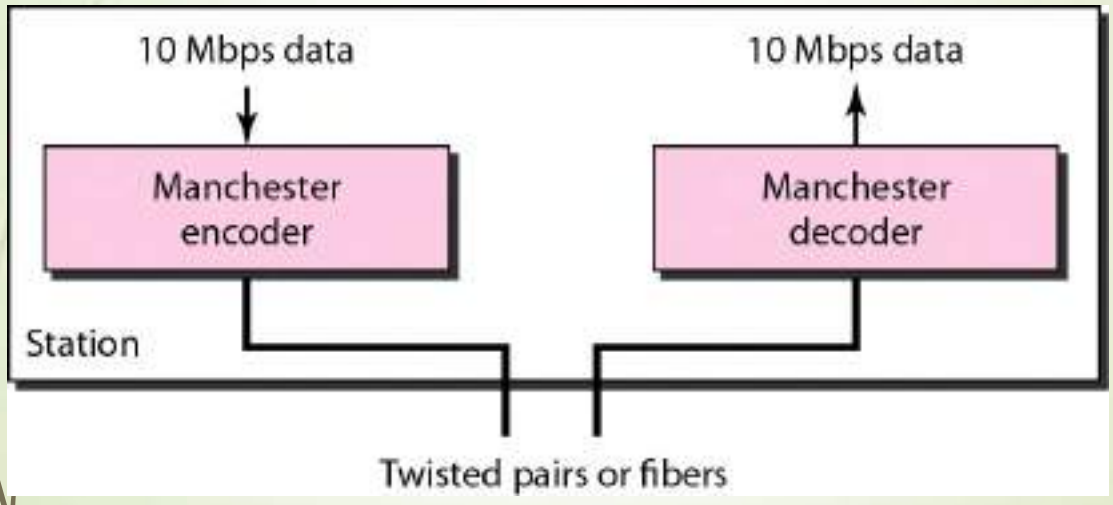
5. Categories of Standard Ethernet



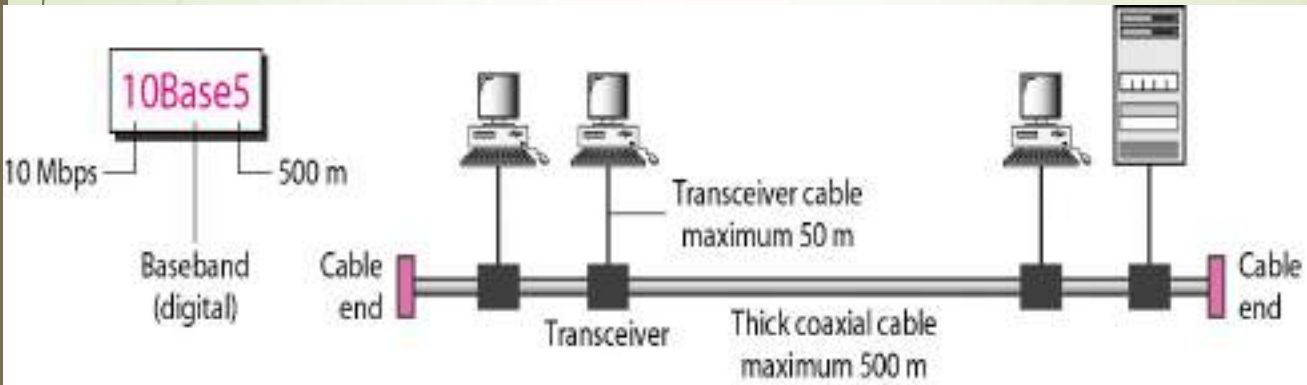
Summary of Standard Ethernet implementations

Implementation	*Medium*	*Medium Length*	*Encoding*
10Base5	Thick coax	500 m	Manchester
10Base2	Thin coax	185 m	Manchester
10Base-T	2 UTP	100 m	Manchester
10Base-F	2 Fiber	2000 m	Manchester

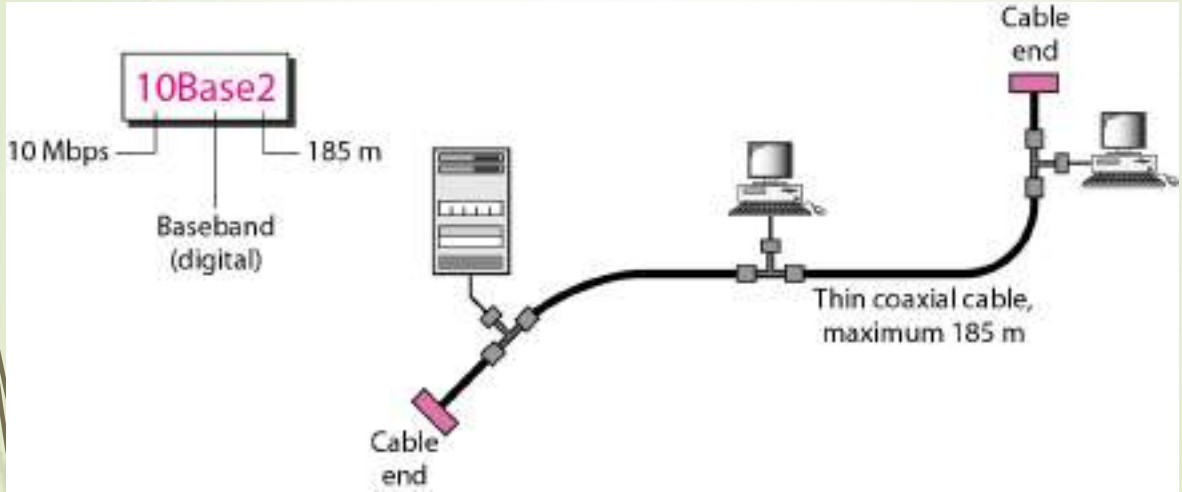
6. Encoding in a Standard Ethernet implementation



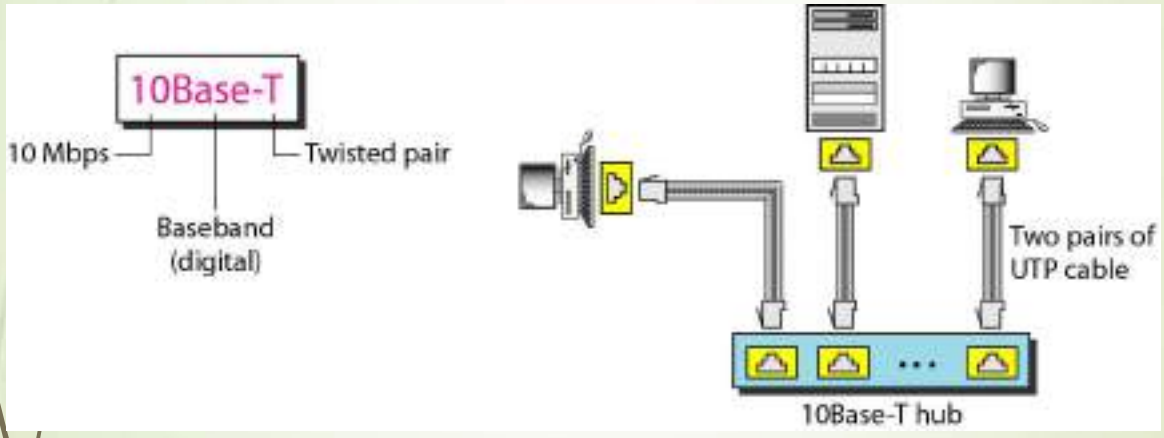
17 ***10Base5 implementation***



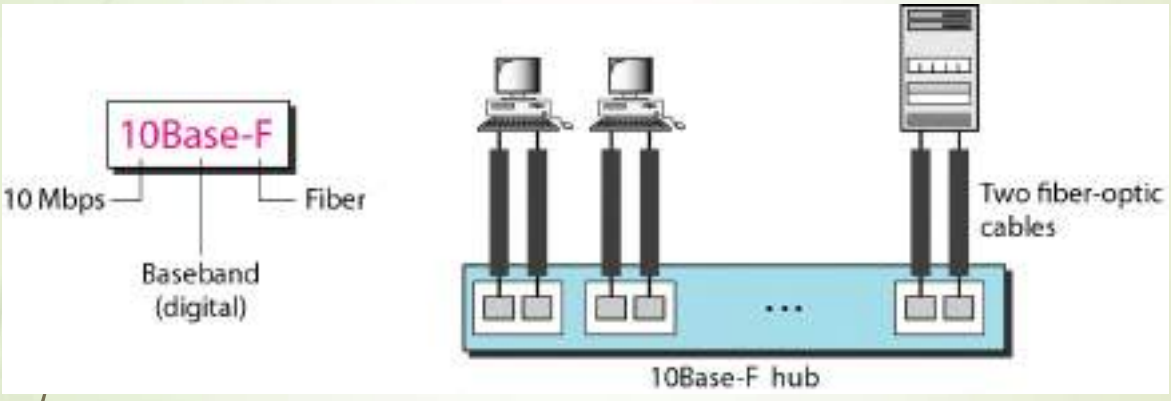
18 ***10Base2 implementation***



19 ***10Base-T implementation***



20 ***10Base-F implementation***



Wireless Networks

1

2

Wireless?

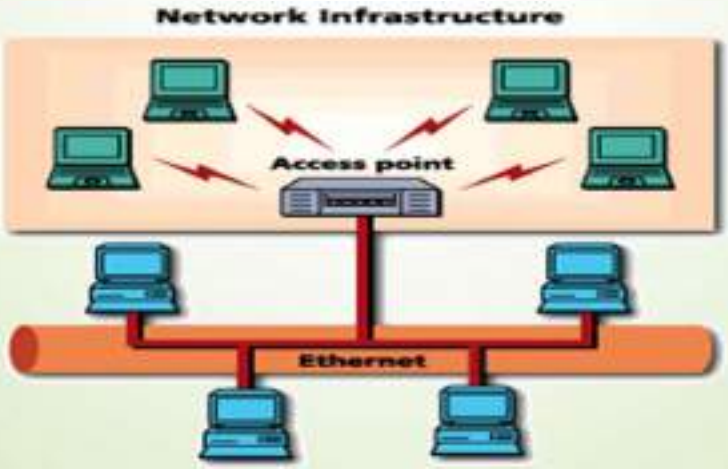
- A wireless LAN or WLAN is a wireless local area network that uses radio waves as its carrier.
- The last link with the users is wireless, to give a network connection to all users in a building or campus.
- The backbone network usually uses cables

3

Common Topologies

The wireless LAN connects to a wired LAN

- There is a need of an access point that bridges wireless LAN traffic into the wired LAN.
- The access point (AP) can also act as a repeater for wireless nodes, effectively doubling the maximum possible distance between nodes.



4

Common Topologies

Complete Wireless Networks

- The physical size of the network is determined by the maximum reliable propagation range of the radio signals.
- Referred to as **ad hoc** networks
- Are self-organizing networks without any centralized control
- Suited for temporary situations such as meetings and conferences.

5

How do wireless LANs work?

- Wireless LANs operate in almost the same way as wired LANs, using the same networking protocols and supporting the most of the same applications.

6

How are WLANs Different?

- They use specialized **physical and data link** protocols
- They integrate into existing networks through **access points** which provide a bridging function
- They let you stay connected as you **roam** from one coverage area to another
- They have unique **security** considerations
- They have specific **interoperability** requirements
- They require **different hardware**
- They offer **performance** that differs from wired LANs.

7

Physical and Data Link Layers

Physical Layer:

- The wireless **NIC** takes **frames** of data from the link layer, scrambles the data in a predetermined way, then uses the modified data stream to modulate a **radio carrier signal**.

Data Link Layer:

- Uses **C**arriers-**S**ense-**M**ultiple-**A**ccess with **C**ollision **A**voidance (CSMA/CA).

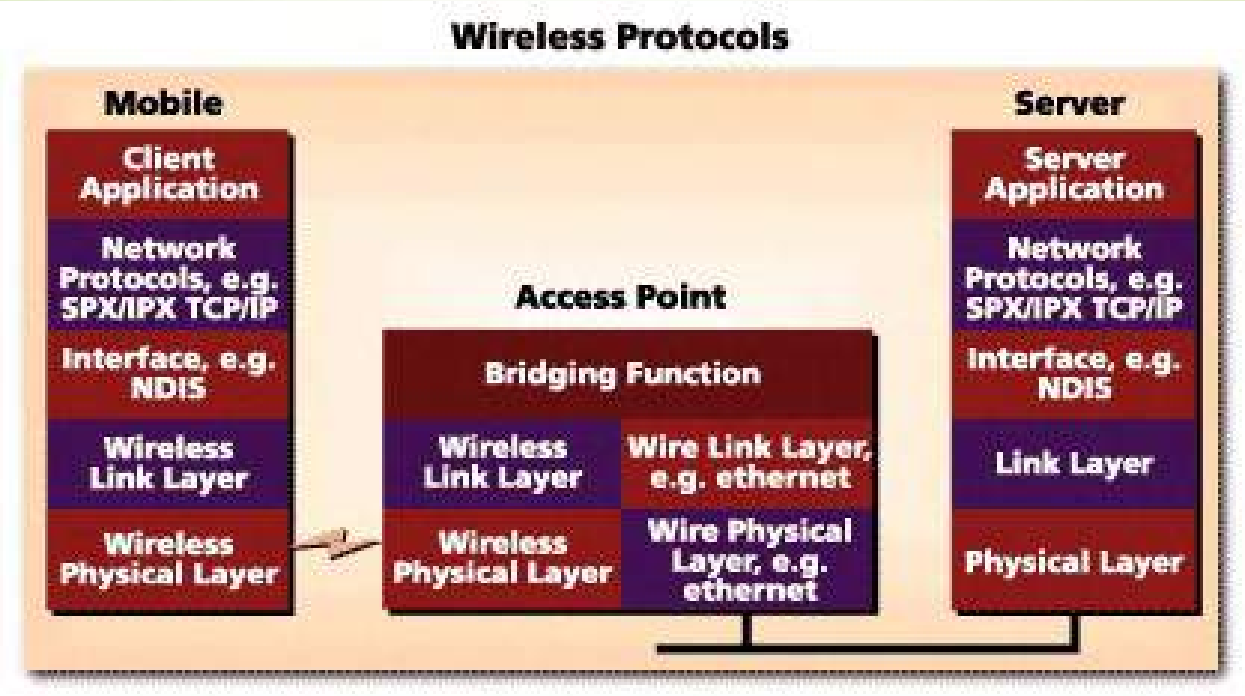
8

Integration With Existing Networks

- Wireless Access Points (APs) - a small device that bridges wireless traffic to your network.
- Most access points bridge wireless LANs into Ethernet networks, but Token-Ring options are available as well.

9

Integration With Existing Networks



10

Roaming

- Users maintain a continuous connection as they roam from one physical area to another
- Mobile nodes automatically register with the new access point.
- Methods: DHCP, Mobile IP
- IEEE 802.11 standard does not address roaming, you may need to purchase equipment from one vendor if your users need to roam from one access point to another.

11

Security

- In theory, spread spectrum radio signals are inherently difficult to decipher without knowing the exact hopping sequences or direct sequence codes used
- The IEEE 802.11 standard specifies optional security called "**W**ired **E**quivalent **P**rivacy" whose goal is that a wireless LAN offer privacy equivalent to that offered by a wired LAN. The standard also specifies optional authentication measures.

12

Interoperability

- Before the IEEE 802.11 interoperability was based on cooperation between vendors.
- IEEE 802.11 only standardizes the physical and medium access control layers.
- Vendors must still work with each other to ensure their IEEE 802.11 implementations interoperate
- Wireless Ethernet Compatibility Alliance (WECA) introduces the Wi-Fi Certification to ensure cross-vendor interoperability of 802.11b solutions

13

Hardware

- PC Card, either with integral antenna or with external antenna/RF module.
- ISA Card with external antenna connected by cable.
- Handheld terminals
- Access points

14

Hardware

CISCO Aironet 350 series

Wireless Handheld Terminal

Semi Parabolic Antenna

BreezeCOM AP

15

Performance

- **802.11a** offers speeds with a theoretically maximum rate of 54Mbps in the 5 GHz band
- **802.11b** offers speeds with a theoretically maximum rate of 11Mbps at in the 2.4 GHz spectrum band
- **802.11g** is a new standard for data rates of up to a theoretical maximum of 54 Mbps at 2.4 GHz.

16

What is 802.11?

- A family of wireless LAN (WLAN) specifications developed by a working group at the Institute of Electrical and Electronic Engineers (IEEE)
- Defines standard for WLANs using the following four technologies
  - Frequency Hopping Spread Spectrum (FHSS)
  - Direct Sequence Spread Spectrum (DSSS)
  - Infrared (IR)
  - Orthogonal Frequency Division Multiplexing (OFDM)
- Versions: 802.11a, 802.11b, 802.11g, 802.11e, 802.11f, 802.11i

17

802.11 - Transmission

- Most wireless LAN products operate in unlicensed radio bands
  - 2.4 GHz is most popular
  - Available in most parts of the world
  - No need for user licensing
- Most wireless LANs use spread-spectrum radio
  - Resistant to interference, secure
  - Two popular methods
    - Frequency Hopping (FH)
    - Direct Sequence (DS)

18

Frequency Hopping Vs. Direct Sequence

- FH systems use a radio carrier that "hops" from frequency to frequency in a pattern known to both transmitter and receiver
  - Easy to implement
  - Resistance to noise
  - Limited throughput (2-3 Mbps @ 2.4 GHz)
- DS systems use a carrier that remains fixed to a specific frequency band. The data signal is spread onto a much larger range of frequencies (at a much lower power level) using a specific encoding scheme.
  - Much higher throughput than FH (11 Mbps)
  - Better range
  - Less resistant to noise (made up for by redundancy – it transmits at least 10 fully redundant copies of the original signal at the same time)

19

802.11a

- Employs Orthogonal Frequency Division Multiplexing (OFDM)
  - Offers higher bandwidth than that of 802.11b, DSSS (Direct Sequence Spread Spectrum)
  - 802.11a MAC (Media Access Control) is same as 802.11b
- Operates in the 5 GHz range

20

802.11a Advantages

- Ultra-high spectrum efficiency
  - 5 GHz band is 300 MHz (vs. 83.5 MHz @ 2.4 GHz)
  - More data can travel over a smaller amount of bandwidth
- High speed
  - Up to 54 Mbps
- Less interference
  - Fewer products using the frequency
    - 2.4 GHz band shared by cordless phones, microwave ovens, Bluetooth, and WLANs

21

802.11a Disadvantages

- Standards and Interoperability
  - Standard not accepted worldwide
  - No interoperability certification available for 802.11a products
  - Not compatible or interoperable with 802.11b
- Legal issues
  - License-free spectrum in 5 GHz band not available worldwide
- Market
  - Beyond LAN-LAN bridging, there is limited interest for 5 GHz adoption

22

802.11a Disadvantages

- Cost
  - 2.4 GHz will still has >40% cost advantage
- Range
  - At equivalent power, 5 GHz range will be ~50% of 2.4 GHz
- Power consumption
  - Higher data rates and increased signal require more power
    - OFDM is less power-efficient then DSSS

23

802.11a Applications

- Building-to-building connections
- Video, audio conferencing/streaming video, and audio
- Large file transfers, such as engineering CAD drawings
- Faster Web access and browsing
- High worker density or high throughput scenarios
  - Numerous PCs running graphics-intensive applications

24

802.11a Vs. 802.11b

802.11a vs. 802.11b	802.11a	802.11b
Raw data rates	Up to 54 Mbps (54, 48, 36, 24,18, 12 and 6 Mbps)	Up to 11 Mbps (11, 5.5, 2, and 1 Mbps)
Range	50 Meters	100 Meters
Bandwidth	UNII and ISM (5 GHz range)	ISM (2.4000—2.4835 GHz range)
Modulation	OFDM technology	DSSS technology

25

802.11g

- 802.11g is a high-speed extension to 802.11b
  - Compatible with 802.11b
  - High speed up to 54 Mbps
  - 2.4 GHz (vs. 802.11a, 5 GHz)
  - Using ODFM for backward compatibility
  - Adaptive Rate Shifting

26

802.11g Advantages

- Provides higher speeds and higher capacity requirements for applications
  - Wireless Public Access
- Compatible with existing 802.11b standard
- Leverages Worldwide spectrum availability in 2.4 GHz
- Likely to be less costly than 5 GHz alternatives
- Provides easy migration for current users of 802.11b WLANs
  - Delivers backward support for existing 802.11b products
- Provides path to even higher speeds in the future

27

802.11b Security Features

- Wired Equivalent Privacy (**WEP**) – A protocol to protect link-level data during wireless transmission between clients and access points.
- Services:
  - **Authentication**: provides access control to the network by denying access to client stations that fail to authenticate properly.
  - **Confidentiality**: intends to prevent information compromise from casual eavesdropping
  - **Integrity**: prevents messages from being modified while in transit between the wireless client and the access point.

28

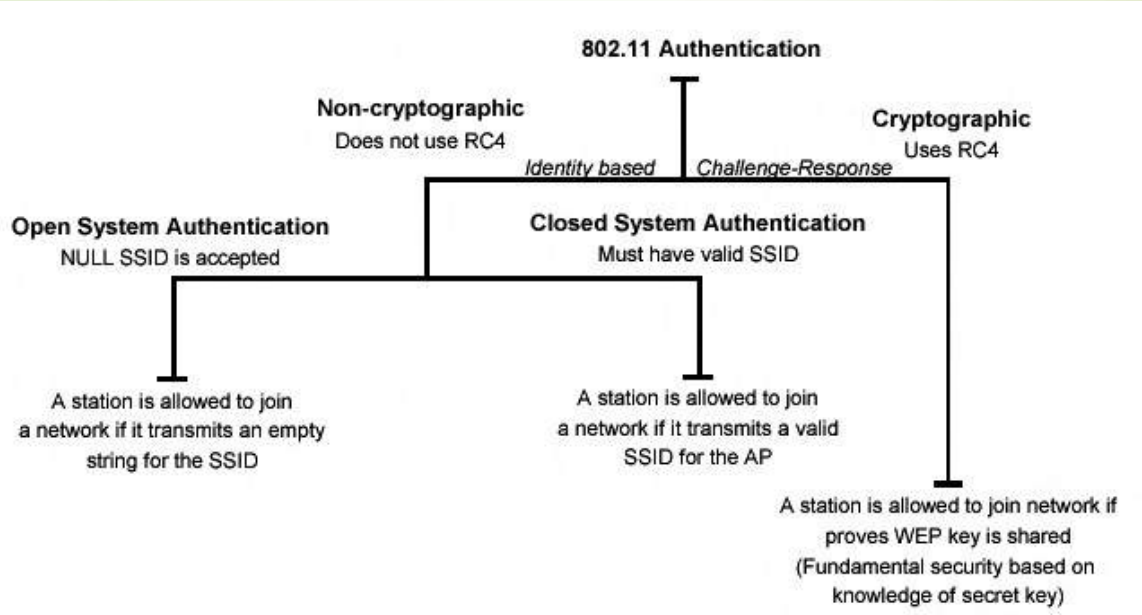
Authentication

Means:

- Based on cryptography
- Non-cryptographic
- Both are identity-based verification mechanisms (devices request access based on the SSID – Service Set Identifier of the wireless network).

29

Authentication



30

Privacy

- Cryptographic techniques
- WEP Uses RC4 symmetric key, stream cipher algorithm to generate a pseudo random data sequence. The stream is XORed with the data to be transmitted
- Key sizes: 40bits to 128bits
- Unfortunately, recent attacks have shown that the WEP approach for privacy is vulnerable to certain attack regardless of key size

31

Data Integrity

- Data integrity is ensured by a simple encrypted version of CRC (Cyclic Redundant Check)
- Also vulnerable to some attacks

32

Security Problems

- Security features in Wireless products are frequently not enabled.
- Use of static WEP keys (keys are in use for a very long time). WEP does not provide key management.
- Cryptographic keys are short.
- No user authentication occurs – only devices are authenticated. A stolen device can access the network.
- Identity based systems are vulnerable.
- Packet integrity is poor.

33

Other WLAN Security Mechanisms

- 3Com Dynamic Security Link
- CISCO LEAP - Lightweight Extensible Authentication Protocol
- IEEE 802.1x – Port-Based Network Access Control
- RADIUS Authentication Support
- EAP-MD5
- EAP-TLS
- EAP-TTLS
- PEAP - Protected EAP
- TKIP - Temporal Key Integrity Protocol
- IEEE 802.11i

34

Choose the right technology

- Usually IEEE 802.11b or 802.11a
- 802.11b offers interoperability (WECA Wi-Fi Certification Program)
- 802.11a offers higher data rates (up to 54 mbps) -> higher throughput per user. Limited interoperability.

35

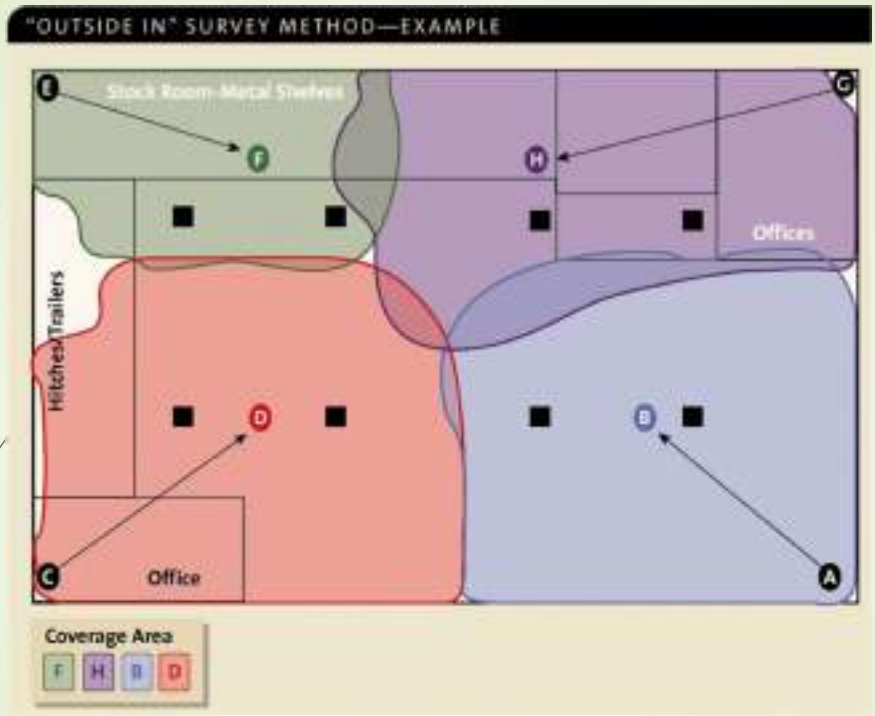
Data rates

- Data rates affect range
- 802.11b 1 to 11 Mbps in 4 increments
- 802.11a 6 to 54 Mbps in 7 increments
- The minimum data rate must be determined at design time
- Selecting only the highest data rate will require a greater number of APs to cover a specific area
- Compromise between data rates and overall system cost

36

The Site Survey

- Helps define the coverage areas, data rates, the precise placement of access point.
- Gather information: diagramming the coverage area and measuring the signal strength, SNR (signal to noise ratio), RF interference levels


"OUTSIDE IN" SURVEY METHOD—EXAMPLE
E
Stock Room-Metal Shelves
F
H
G
Offices
Hitches/Trailers
D
D
C
Office
A
Coverage Area
F
H
B
D

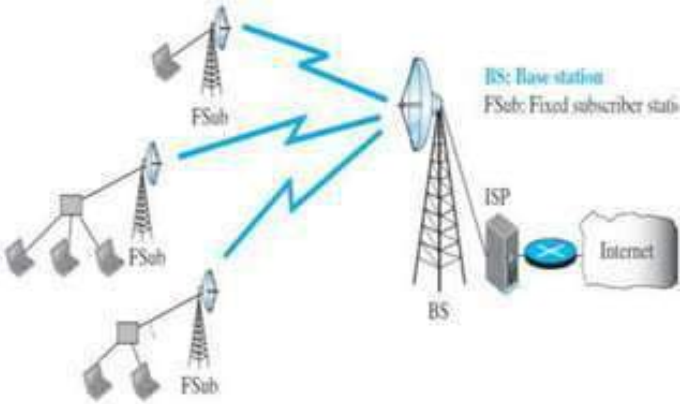
Other Wireless Networks

WiMAX

- **WiMAX** stands for **Worldwide Interoperability for Microwave Access**
- It provides the "last mile" broadband wireless access.
- WiMAX provides two types of services to subscribers:
  - Fixed
  - Mobile

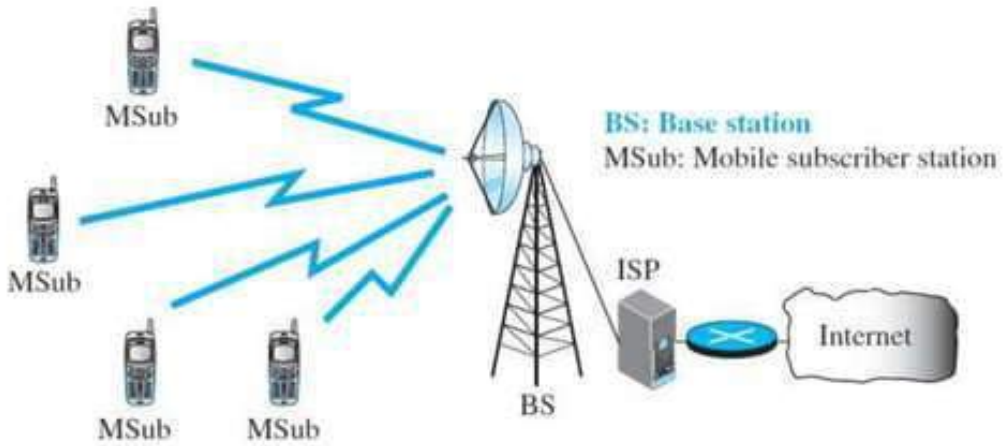
[redacted] Fixed WiMAX

- A base station can use different types of antenna (omnidirectional, sector, or panel) to optimize the performance.
- WiMAX uses a beamsteering **adaptive antenna system (AAS).**
- While transmitting, it can focus its energy in the direction of the subscriber;
- While receiving, it can focus in the direction of the subscriber station to receive maximum energy sent by the subscriber.
- The fixed service can be compared with the service provided by the telephone and the network companies using wired connections.
- WiMAX also uses a MIMO antenna system, which can provide simultaneous transmitting and receiving.



[redacted] Mobile WiMAX

- It is the same as fixed service except the subscribers are mobile stations that move from one place to another.
- The same issues involved in the cellular telephone system, such as roaming, are present here.



[redacted] IEEE Project 802.16

- **WiMAX** is the result of the IEEE 802.16 project, which was an effort to standardize the proprietary **broadband wireless system** in 2002.
- The standard is referred to as ***wireless local loop***
- A later revision of IEEE 802.16 created two new standards called
  - IEEE 802.16d, which concentrates on the fixed WiMAX
  - IEEE 802.16e, which defines the mobile WiMAX.
- The two new standards do not change the main idea behind the original 802.16, but concentrate on the nature of two services

Difference between 802.16 and 802.11

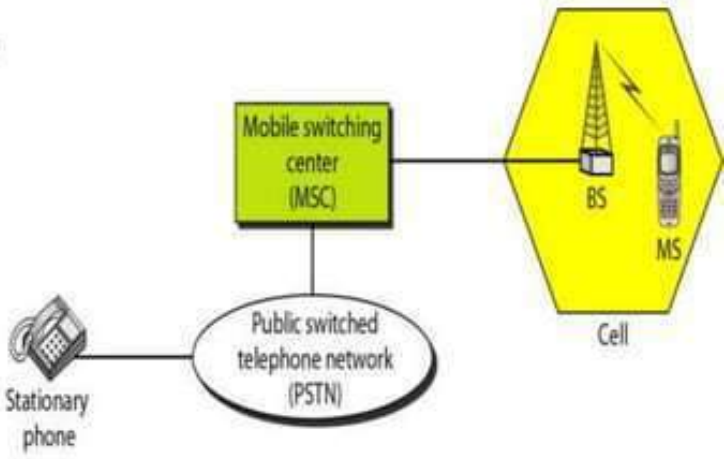
IEEE 802.11	IEEE 802.16
802.11 is a standard for a wireless LAN	802.16 is a standard for a wireless WAN (or MAN).
Range upto **100m**	Range **upto 40km**
802.11 defines a connectionless communication	802.16 defines a connection oriented service.
Indoor Coverage	Outdoor coverage
Hundred of users	Thousand of users
54Mbps max speed	10-100Mbps max speed

[redacted] CELLULAR TELEPHONY

- **Cellular telephony** is designed to provide communications between
  - Two moving units, called ***mobile stations*** *(MSs)*, or
  - between one **mobile unit** and one **stationary unit**, often called a ***land unit***.
- A service provider must be able to **locate and track a caller**, assign a channel to the call, and transfer the channel from base station to base station as the caller moves out of range.
- To make this tracking possible, each cellular service area is divided into small regions called ***cells***.

[redacted] CELLULAR TELEPHONY contd..

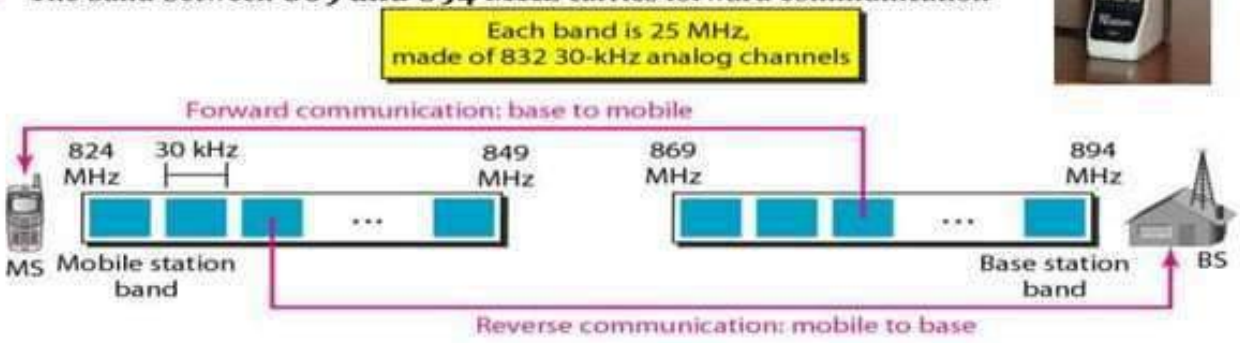
- Each cell contains an antenna and is controlled by a solar- or Ac powered network station, called the ***base station*** **(BS)**.
- Each base station, in turn, is controlled by a switching office, called a ***mobile switching center (MSC).***
- The MSC coordinates communication between all the base stations and the telephone central office.
- It is a computerized center that is responsible for connecting calls, recording call information, and billing
- Cell size is **not fixed** and can be increased or decreased
- The typical radius of a cell is 1 to 12 mi



First Generation (1G)

- Cellular telephony is now in its fourth generation.
- The first generation was designed for voice communication using **analog signals**.
- **AMPS(**Advanced Mobile Phone Service) **is an analog cellular phone system using FDMA**

- AMPS operates in the ISM 800-MHz band.
- The system uses two separate analog channels
  - one for forward (base station to mobile station) communication
  - one for reverse (mobile station to base station) communication
- The band between **824 and 849 MHz** carries reverse communication
- The band between **869 and 894 MHz** carries forward communication



Second Generation (2G)

- The second generation of wireless networks designed to improve on **analog with digital circuit-switched solutions.**
- It was commercially launched in 1991 as **GSM** standard in Finland.
- As with 1G phones, 2G phones didn't have any worldwide standardizations.
- 2G systems were also known as **personal communications services (PCM)**.

Advantages of 2G over 1G

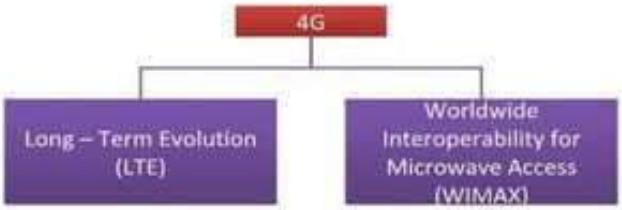
- It allows **voice signals to be digitized and compressed**. So, they are more efficient on frequency spectrum than 1G.
- They introduced data services for mobile in form of **SMS** text messaging.
- **Data and voice signals are digitally encrypted**. So, security against eavesdropping and fraud increased manifold.
- Digital signals **consume less battery power**. And so mobile sets are much more energy efficient than their 1G counterparts.

[redacted] Third Generation (3G)

- The third generation of cellular telephony refers to a combination of technologies that provide both digital data and voice communication.
- 3G systems comply with the International Mobile Telecommunications-2000 (IMT-2000) specifications by the International Telecommunication Union (ITU).
- The first 3G services were available in 1998.
- It provides high speed transmission having data transfer rate more than 0.2Mbps.
- Global roaming services are available for both voice and data.
- It offers advanced **multimedia access** like playing music, viewing videos, television services etc.
- It provides access to all **advanced Internet services**, for example surfing webpages with **audio and video**.

[redacted] Fourth Generation (4G)

- Fourth Generation (4G) mobile phones provides **broadband cellular network** services and is successor to 3G mobile networks.
- It provides an all IP packet switched network for transmission of **voice, data, signals and multimedia.**
- It aims to provide **high quality uninterrupted services** to any location at any time.
- As laid down in IMT-Advanced specifications, 4G networks should have peak data rates of 100Mbps for highly mobile stations like train, car etc., and 1Gbps for low mobility stations like residence etc.
- It also lays down that 4G networks should make it possible for 1 Gbps downlink over less than 67 MHz bandwidth.
- They provide have smooth handoffs across heterogeneous network areas.



Categories

- **Long – Term Evolution (LTE)**
  - Long – term evolution or LTE is an extension of the 3G technology.
  - It is a standard for high-speed mobile communication, based upon GSM/EDGE and UMTS/HSPA technologies.
  - The peak data rate for download is 100 Mbps and upload is 50 Mbps.
  - The LTE Advanced meets the specifications of IMT-Advanced standard for 4G technology.
  - Its peak data rates are 1000 Mbps for downlink and 500 Mbps for uplink.

- **Worldwide Interoperability for Microwave Access (WIMAX)**
  - WiMAX is a mobile wireless broadband access (MWBA) standard is sometimes branded 4G.
  - It offers peak data rates of 128 Mbps for downlink and 56 Mbps for uplink over 20 MHz wide channels.
  - The latest version of WIMAX is not compatible to the earlier versions and instead is compatible with LTE.

Application Areas of 4G

- Advanced mobile web access
- IP telephony
- High-resolution high-speed gaming services
- Streamed multimedia and data
- High-definition mobile TV
- Video conferencing
- 3D television

SATELLITE NETWORKS

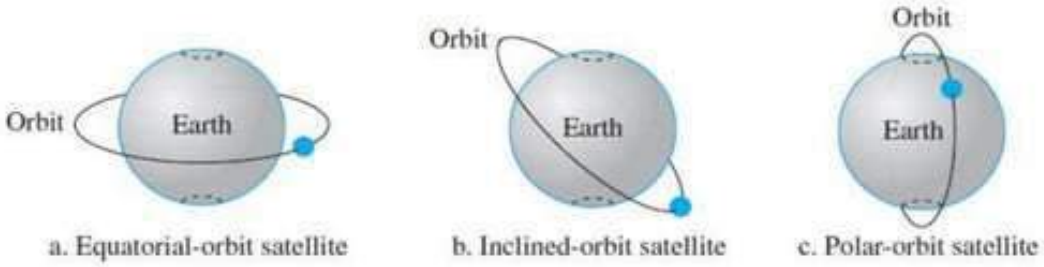
- A *satellite network* is a combination of nodes, some of which are satellites, that provides communication from one point on the Earth to another.
- A node in the network can be a satellite, an Earth station, or an end-user terminal or telephone.
- Satellite networks are like cellular networks in that they divide the planet into cells.
- Satellites can provide transmission capability to and from any location on Earth,

Operation

- Let us first discuss some general issues related to the operation of satellites.
  - ***Orbits***
  - ***Footprint***
  - ***Frequency Bands for Satellite Communication***

Orbits

- An artificial satellite needs to have an *orbit,* the path in which it travels around the Earth.
- The orbit can be equatorial, inclined, or polar.
- The period of a satellite, the time required for a satellite to make a complete trip around the Earth, is determined by Kepler's law,
- which defines the period as a function of the distance of the satellite from the center of the Earth.



a. Equatorial-orbit satellite b. Inclined-orbit satellite c. Polar-orbit satellite

Footprint

- Satellites process microwaves with bidirectional antennas (line-of-sight).
- Therefore, the signal from a satellite is normally aimed at a specific area called the ***footprint.***
- The signal power at the center of the footprint is maximum.
- The power decreases as we move out from the footprint center.
- The boundary of the footprint is the location where the power level is at a predefined threshold.

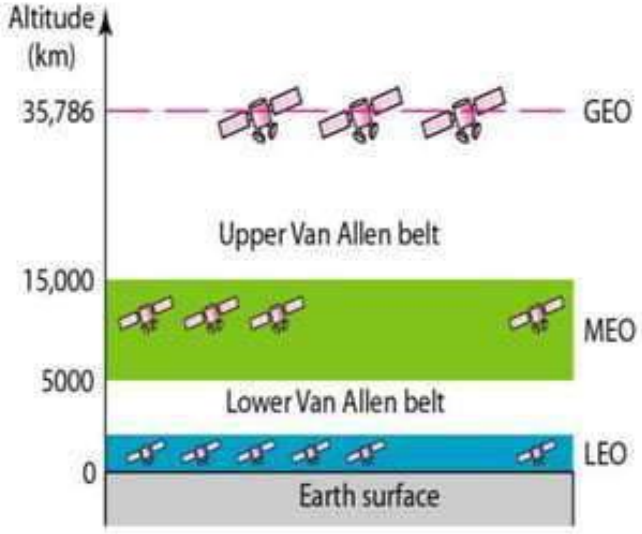
Frequency Bands for Satellite Communication

- The frequencies reserved for satellite microwave communication are in the gigahertz (GHz) range.
- Each satellite sends and receives over two different bands.
- Transmission from the Earth to the satellite is called the ***uplink***.
- Transmission from the satellite to the Earth is called the ***downlink***.
- Below table gives the band names and frequencies for each range.

Band	*Downlink, GHz*	*Uplink, GHz*	*Bandwidth, MHz*
L	1.5	1.6	15
S	1.9	2.2	70
C	4.0	6.0	500
Ku	11.0	14.0	500
Ka	20.0	30.0	3500

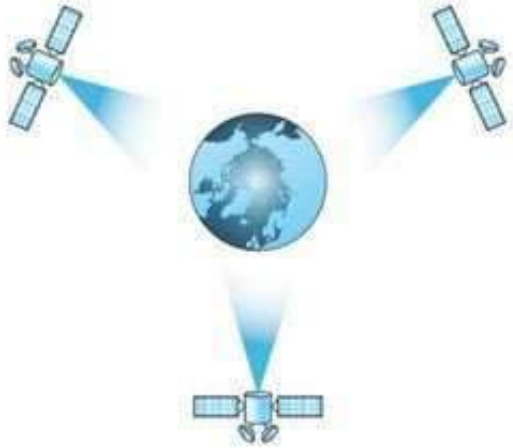
Three Categories of Satellites

- **Geostationary Earth orbit (GEO), low-Earth-orbit (LEO),** and **medium-Earth-orbit (MEO).**
- Figure shows the satellite altitudes with respect to the surface of the Earth.
- One reason for having different orbits is the existence of two Van Allen belts.
- A Van Allen belt is a layer that contains charged particles.
- A satellite orbiting in one of these two belts would be totally destroyed by the energetic charged particles.
- The MEO orbits are located between these two belts.



[redacted] GEO Satellites

- Synchronous with respect to earth
- Footprint is covering almost 1/3rd of the Earth
  - 3-4 Satellites are enough to cover the earth
- Circular Orbit, Satellite visibility 24 hour
- Altitude : 36,000 km
- Inclination Angle : 0
- **Applications:**
  - TV and radio broadcast
  - Weather forecast
  - Backbones for the telephone networks
- **Issues**
  - Shading of the signals
  - High latency (270 ms)
  - Transferring a Satellite into GEO is very expensive
  - Cannot be used for small mobile phones (High transmit power needed)

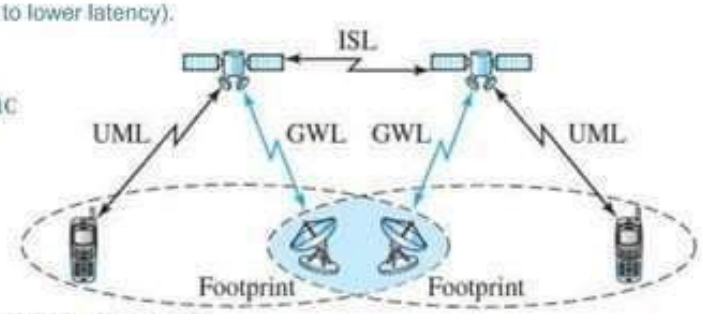


[redacted] MEO Satellites

- Altitude 10000km-20000km
- Orbital period 6-12 hour
- 10-15 satellites required
- Satellite visibility 2-4 hrs
- Propagation delay less
- Set-up cost is medium
- MEO can cover larger populations,so requiring fewer handovers than LEO
- Issues
  - Larger Delay: 70–80 ms
  - Need higher transmit power
  - Special antennas for smaller footprints
- Example:
  - ICO (Intermediate Circular Orbit, Inmarsat) start ca. 2000

[redacted] LEO Satellites

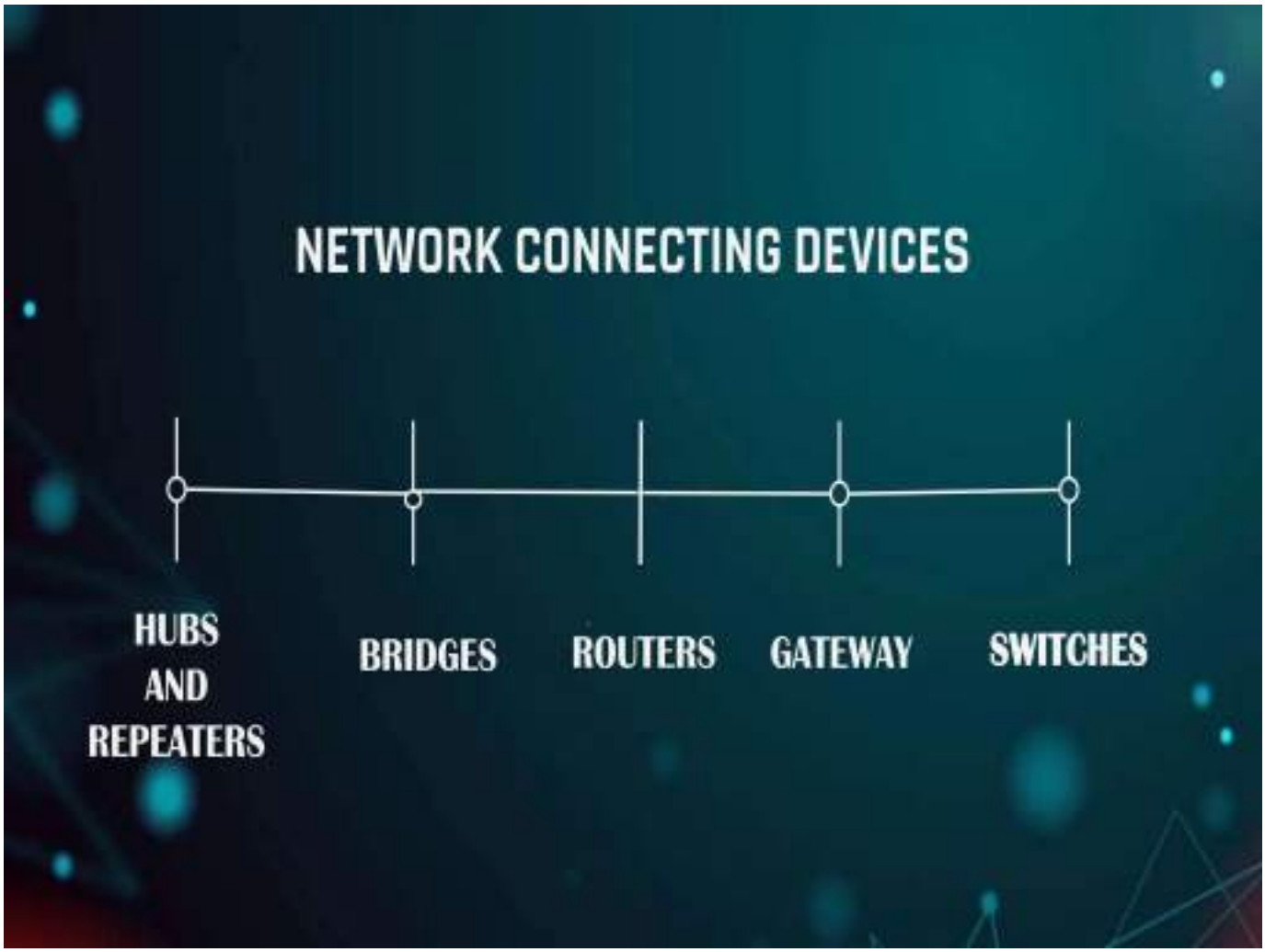
- Altitude 500-2000km
- Satellite visibility 10-20m, Orbital period 5-8 hour
- Delay : relatively low (approx 10 ms)
- Smaller footprints of LEOs allow for better frequency reuse, similar to the concepts used for cellular networks
- **Applications:**
  - Remote sensing and Mobile communication services (due to lower latency).
- **Issues:**
  - 48 and above satellites required to cover whc
  - Short life: 4-10 years
  - Larger Handoffs
- **Examples:**
  - Iridium (start 1998, 66 satellites)
    - Bankruptcy in 2000, deal with US DoD for free use
  - Globalstar (start 1999, 48 satellites)
    - Not many customers (2001: 44000)



Parameter	LEO	MEO	GEO
Satellite Height	700 to 1400 Km	10,000 to 15,000 Km	36,000 Km
Orbital Period	10-40 minutes	2-8 hours	24 hours
Number of Satellites per operator	40 +	10 to 15	3 to 4
Satellite Life	3 to 7 yrs	10 to 15 yrs	10 to 15 yrs
Space Segment Cost	High	Low	Medium
Terrestrial Gateway Cost	High	Medium	Low
Propagation Loss	Least	High	Highest

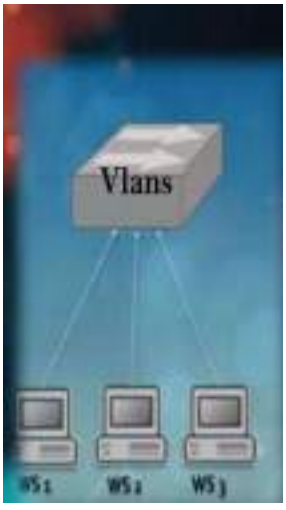

CONNECTING
DEVICES AND
VIRTUAL LANS


WHAT IS
CONNECTING
DEVICE ??
A connecting device,
In simple words, is
something that links or
joins different things
together. It's like a bridge
or a connector that allows
various devices or parts of
a system to communicate
or work together.



VIRTUAL LAN

- A Virtual LAN (VLAN) is a technology used in computer networking that allows you to segment a physical network into multiple logical networks. These logical networks operate as if they are separate and isolated, even though they share the same physical infrastructure.
- VLANs are typically used to improve network efficiency, security, and management by grouping devices into different broadcast domains regardless of their physical location.
- This segmentation is achieved through network switches and isa fundamental tool for network administrators to control and organize traffic in complex networks.



SEVERAL FEATURES AND BENEFITS

- Improved network security: VLANs can be used to separate network traffic andlimit access to specific network resources that improves security.
- Better network performance: By segregating network traffic into smaller logicalnetworks, VLANs can reduce the amount of broadcast traffic.
- Simplified network management: VLANs allow network administrators to groupdevices together logically, rather than physically, which can simplify networkmanagement tasks such as configuration, troubleshooting etc.
- Flexibility: VLANs can be configured dynamically, allowing network administratorsto quickly and easily adjust network configurations as needed.
- Cost savings: VLANs can help reduce hardware costs by allowing multiple virtualnetworks to share a single physical network .

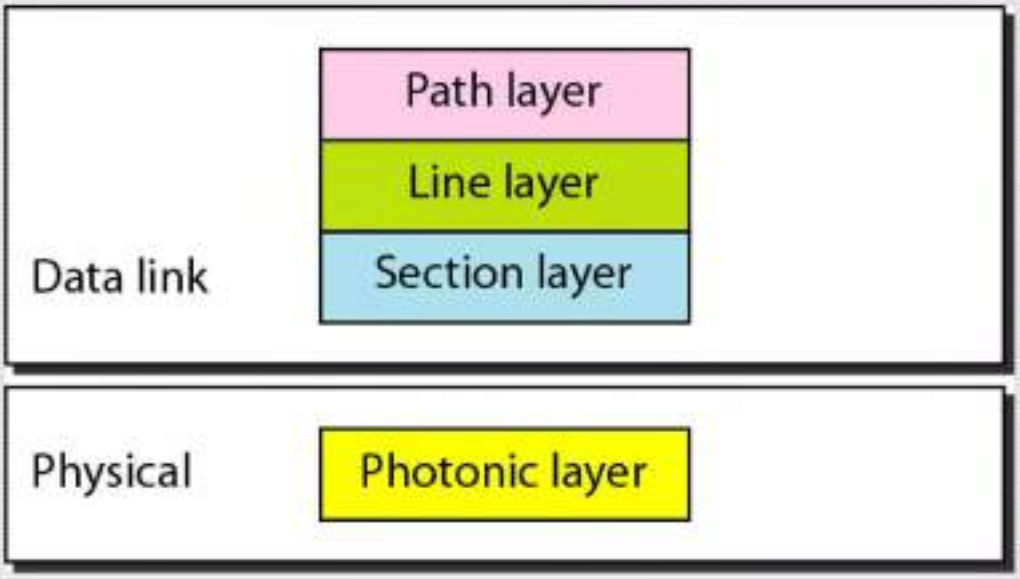
Introduction to SONET

- SONET stands for Synchronous Optical Network.
- Standard for optical telecommunications transport.
- It defines optical carrier (OC) levels.
- Synchronous transport signals (STSs) for the fiber-optic–based transmission hierarchy.

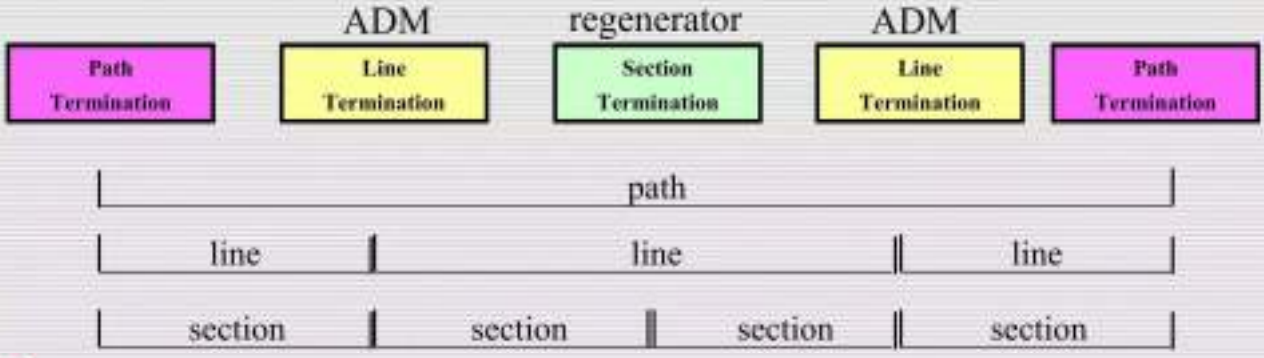
SONET/SDH Rates

STS	*OC*	*Rate (Mbps)*	*STM*
STS-1	OC-1	51.840	
STS-3	OC-3	155.520	**STM-1**
STS-9	OC-9	466.560	**STM-3**
STS-12	OC-12	622.080	**STM-4**
STS-18	OC-18	933.120	**STM-6**
STS-24	OC-24	1244.160	**STM-8**
STS-36	OC-36	1866.230	**STM-12**
STS-48	OC-48	2488.320	**STM-16**
STS-96	OC-96	4976.640	**STM-32**
STS-192	OC-192	9953.280	**STM-64**

SONET Layers Compared with OSI

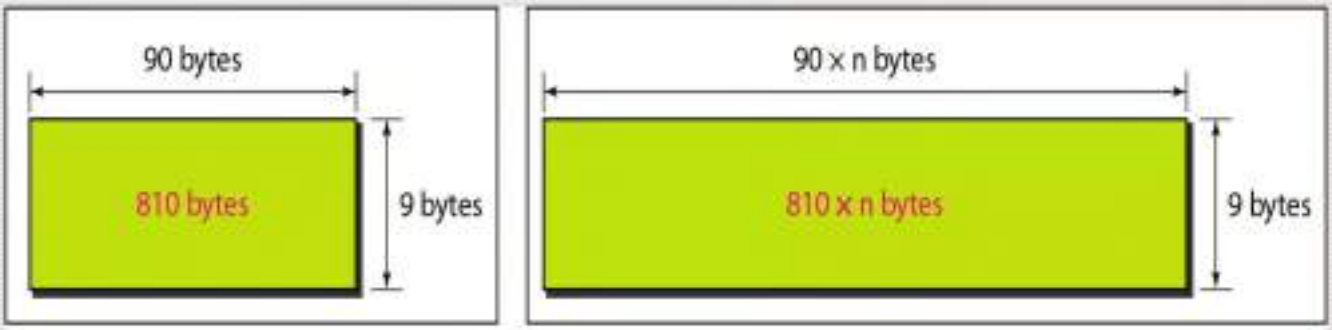


SONET Architecture



- **Signals:** **Electrical signaling** levels called STSs (Synchronous Transport Signals, (STMs)). **Optical signals** are called OCs (Optical Carriers)
- **Devices:** STS Multiplexer/ Demultiplexer, Regenerator, Add/Drop Multiplexer and Terminals
- **Connections:** SONET devices are connected using sections, lines, and paths

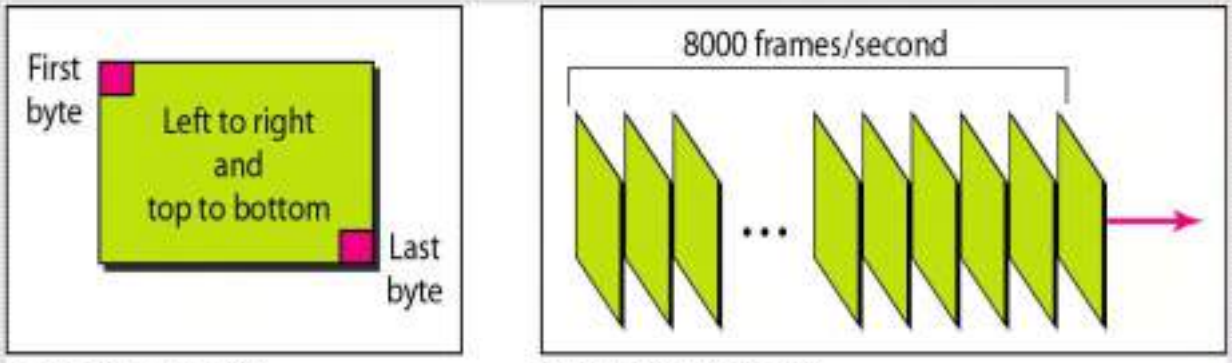
SONET Frames



a. STS-1 frame b. STS-n frame

- Each synchronous transfer signal STS-n is composed of 8000 frames.
- Each frame is a two-dimensional matrix of bytes with 9 rows by 90 × n columns.

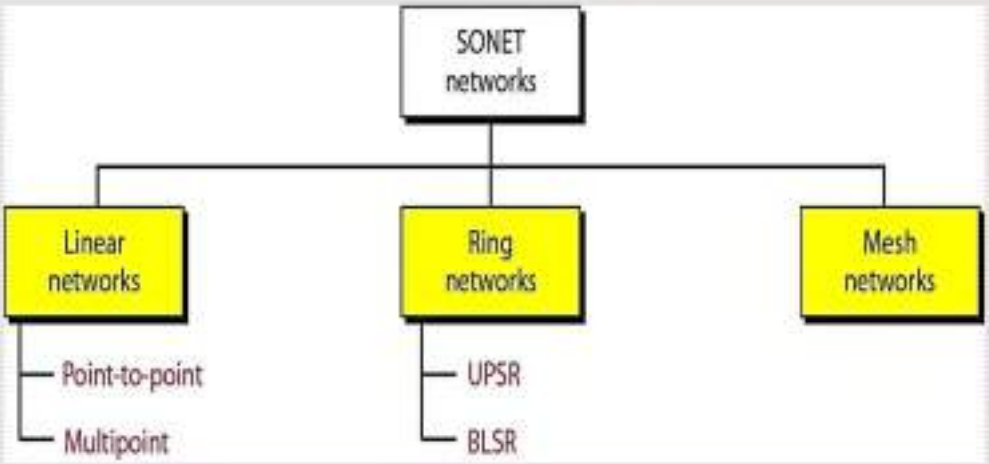
STS Frames in Transition



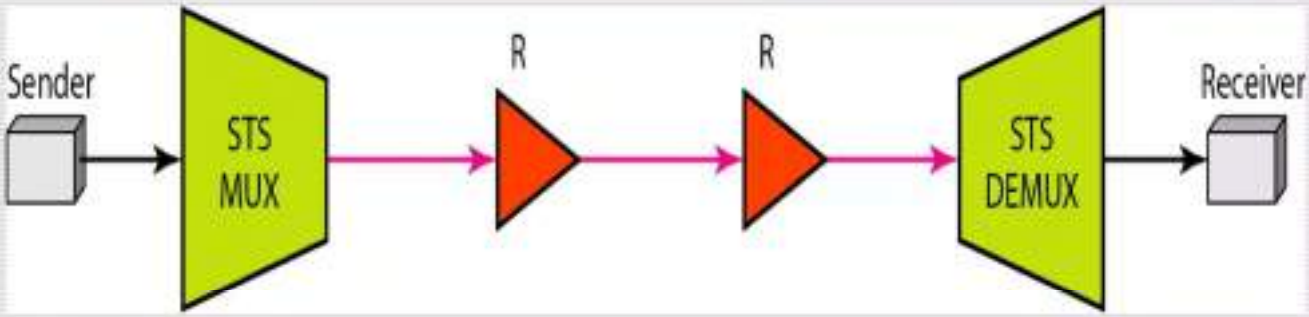
a. Byte transmission b. Frame transmission

- A SONET STS-n signal is transmitted at 8000 frames per second.
- Each byte in a SONET frame can carry a digitized voice channel.
- In SONET, the data rate of an STS-n signal is *n* times the data rate of an STS-1 signal.
- In SONET, the duration of any frame is 125 μs.

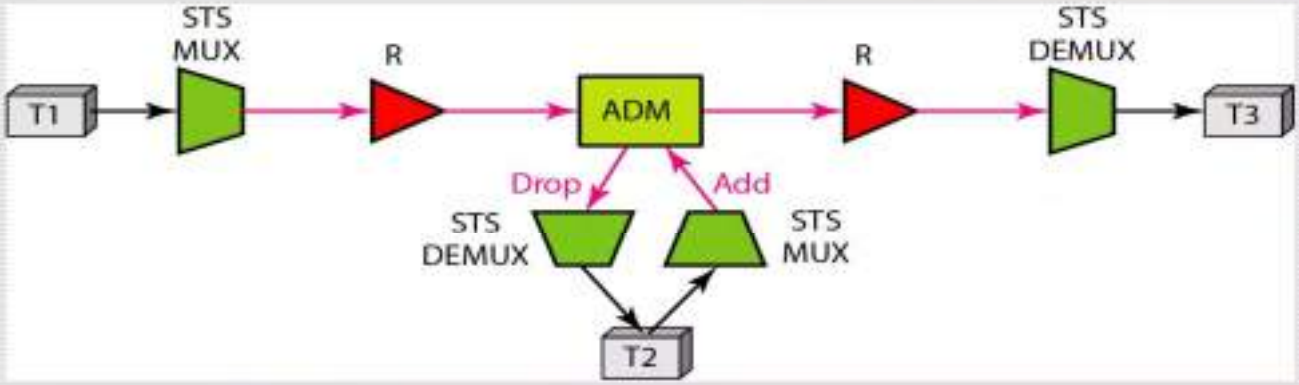
SONET Networks



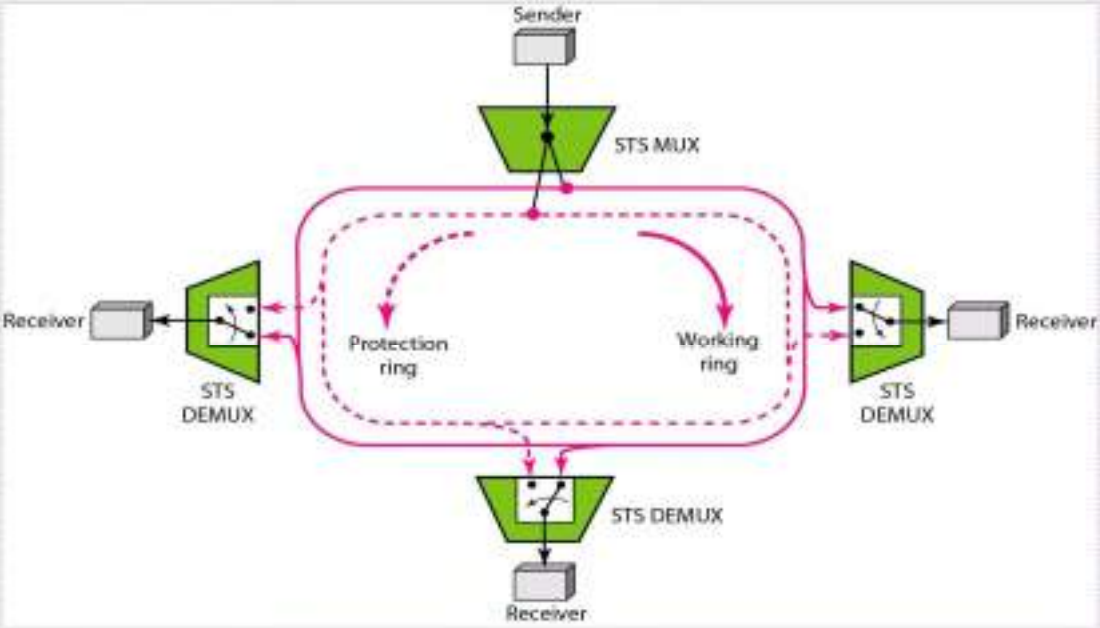
Point-to-Point Network



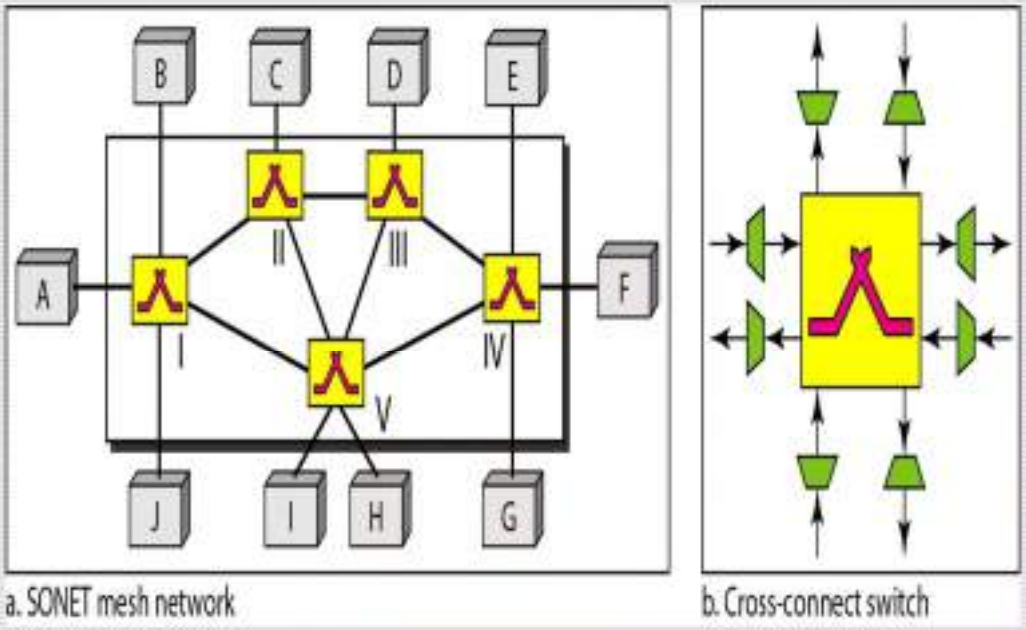
Multipoint Network



Ring Network:
Unidirectional Path Switching Ring



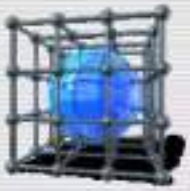
Mesh Network



a. SONET mesh network

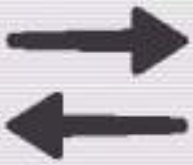
b. Cross-connect switch

Advantages



Reduced network complexity

Flexible Topologies



High data rate.

Efficient management of bandwidth

Protection Bandwidth

Disadvantages

- High cost.
- Complex equipment as compared to cheaper Ethernet
- Strict synchronization schemes required.