MODULE 1

SECTION :A,B,C,D

Topic	Portion
Introduction Chapter 1	Data Communications: Components, Representations, Data Flow, Networks: Physical Structures, Network Types: LAN, WAN, Switching, Internet
Network Models Chapter 2	Protocol Layering: Scenarios, Principles, Logical Connections.
	TCP/IP Protocol Suite: Layered Architecture, Layers in TCP/IP suite, Description of layers
	Encapsulation and Decapsulation, Addressing, Multiplexing and Demultiplexing, The OSI Model: OSI Versus TCP/IP
Data-Link Layer Chapter 11	Introduction: Nodes and Links, Services, Categories' of link, Sublayers, Link Layer addressing: Types of addresses, ARP.
	Data Link Control (DLC) services: Framing, Flow and Error Control,
	Data Link Layer Protocols: Simple Protocol, Stop and Wait protocol, Piggybacking

TEXT BOOK: Data Communications and Networking, B Forouzan**, 5**th** Ed, McGrawHill Education, 2016, ISBN: 1-25-906475-3**

MODULE 1

Data communications

When we communicate, we are sharing information. This sharing can be local or remote. Between individuals,

local communication usually occurs face to face, while remote communication takes place over distance.

The term **telecommunication**, which includes telephony, telegraphy, and television, means communication at a distance (tele is Greek for "far"). The word data refers to information presented in whatever form is agreed upon by the parties creating and using the data.

 Data communications are the exchange of data between two devices via some form of transmission medium such as a wire cable. For data communications to occur, the communicating devices must be part of a communication system made up of a combination of hardware (physical equipment) and software (programs). The effectiveness of a data communications system depends on four fundamental characteristics: delivery, accuracy, timeliness, and jitter.

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

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

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

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

Components

A data communications system has five components
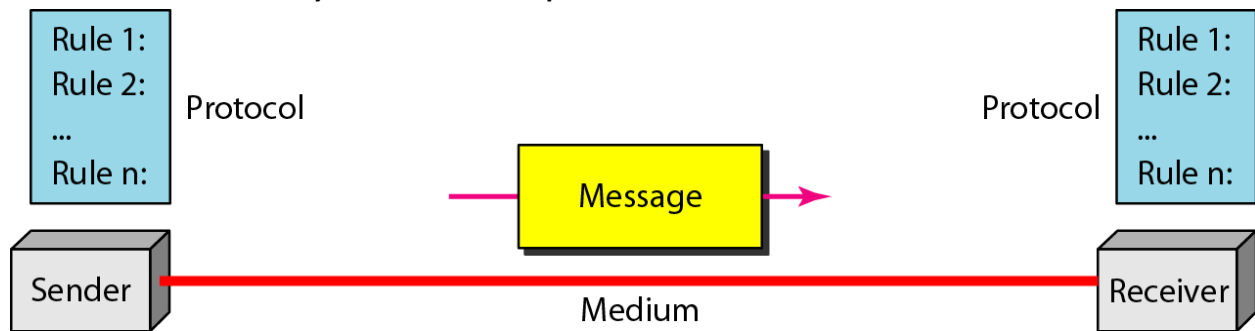
1.Message-. The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.

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

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

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

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

Data Representation

Information today comes in different forms such as text, numbers, images, audio, and video.

Text -In data communications, text is represented as a bit pattern, a sequence of bits (0s or 1s). Different sets of bit patterns have been designed to represent text symbols. Each set is called a code, and the process of representing symbols is called coding. Today, the prevalent coding system is called **Unicode**, which uses 32 bits to represent a symbol or character used in any language in the world. The American Standard Code for Information Interchange (ASCII), developed some decades ago in the United States, now constitutes the first 127 characters in Unicode and is also referred to as Basic Latin.

Numbers- are also represented by bit patterns. However, a code such as ASCII is not used to represent numbers; the number is directly converted to a binary number to simplify mathematical operations. Appendix B discusses several different numbering systems. Images

Images- are also represented by bit patterns. In its simplest form, an image is composed of a matrix of pixels (picture elements), where each pixel is a small dot. The size of the pixel depends on the resolution.

For example, an image can be divided into 1000 pixels or 10,000 pixels. In the second case, there is a better representation of the image (better resolution), but more memory is needed to store the image. After an image is divided into pixels, each pixel is assigned a bit pattern. The size and the value of the pattern depend on the image. For an image made of only black and-white dots (e.g., a chessboard), a 1-bit pattern is enough to represent a pixel. If an image is not made of pure white and pure black pixels, we can increase the size of the bit pattern to include gray scale. For example, to show four levels of gray scale, we can use 2-bit patterns. A black pixel can be represented by 00, a dark gray pixel by 01, a light gray pixel by 10, and a white pixel by 11. There are several methods to represent color images. One method is called RGB, so called because each color is made of a combination of three primary colors: red, green, and blue. The intensity of each color is measured, and a bit pattern is assigned toit. Another method is called YCM, in which a color is made of a combination of three other primary colors: yellow, cyan, and magenta.

Audio- Audio refers to the recording or broadcasting of sound or music. Audio is by nature different from text, numbers, or images. It is continuous, not discrete. Even when we use a microphone to change voice or music to an electric signal, we create a continuous signal.

Video- Video refers to the recording or broadcasting of a picture or movie. Video can either be produced as a continuous entity (e.g., by a TV camera), or it can be a combination of images, each a discrete entity, arranged to convey the idea of motion.

Data Flow

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

simplex mode- the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive (see Figure a).

example-Keyboards and traditional monitors are examples of simplex devices. The keyboard can only introduce input; the monitor can only accept output.

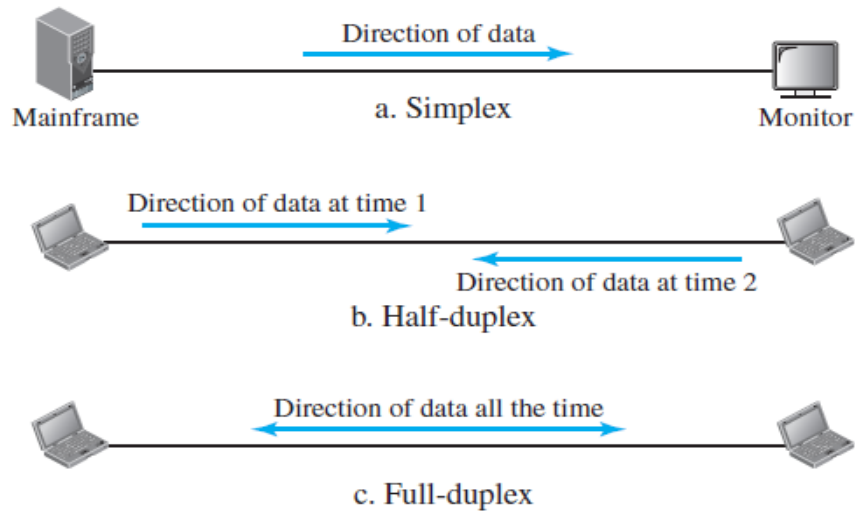simplex mode can use the entire capacity of the channel to send data in one direction

fig: data flow

Half-Duplex- In half-duplex mode, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa (see Figure b). The half-duplex mode is like a one-lane road with traffic allowed in both directions. When cars are traveling in one direction, cars going the other way must wait.

example-Walkie-talkies and CB (citizens band) radios are both half-duplex systems.

The half-duplex mode is used in cases where there is no need for communication in both directions at the same time; the entire capacity of the channel can be utilized for each direction.

 Full-Duplex- In full-duplex mode (also called duplex), both stations can transmit and receive simultaneously (see Figure c). The full-duplex mode is like a two-way street with traffic flowing in both directions at the same time. In full-duplex mode, signals going in one direction share the capacity of the link with signals going in the other direction.

This sharing can occur in two ways: Either the link must contain two physically separate transmission paths, one for sending and the other for receiving; or the capacity of the channel is divided between signals traveling in both directions.

NETWORKS

A network is the interconnection of a set of devices capable of communication. a device can be a host (or an end system as it is sometimes called) such as a large computer, desktop, laptop, workstation, cellular phone, or security system.

A device can also be a connecting device such as a router, which connects the network to other networks, a switch, which connects devices together, a modem (modulator-demodulator), which changes the form of data, and so on.

These devices in a network are connected using wired or wireless transmission media such as cable or air. When we connect two computers at home using a plug-and-play router, we have created a network, although very small.

Network Criteria

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

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

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

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

Physical Structures

Network attributes - Type of Connection and physical topology

Type of Connection

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

There are two possible types of connections:

Point-to-Point -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 Figure a).

example-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.

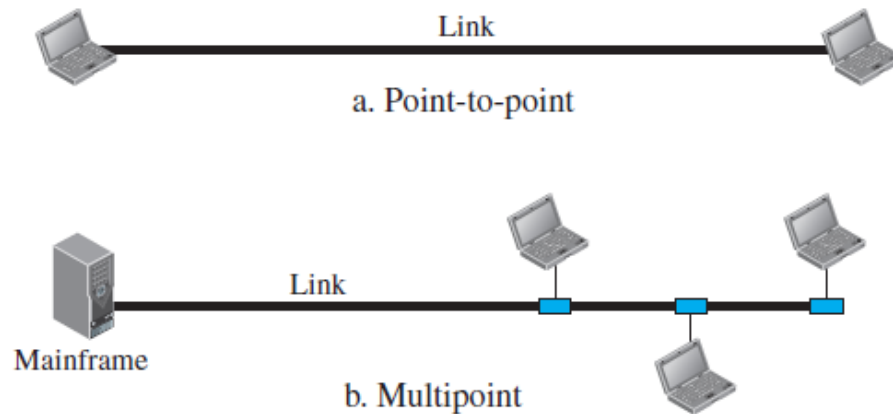Multipoint A multipoint (also called multidrop) connection is one in which more than two specific devices share a single link (see Figure b).

a. Point-to-point

b. Multipoint

fig: Type of connection

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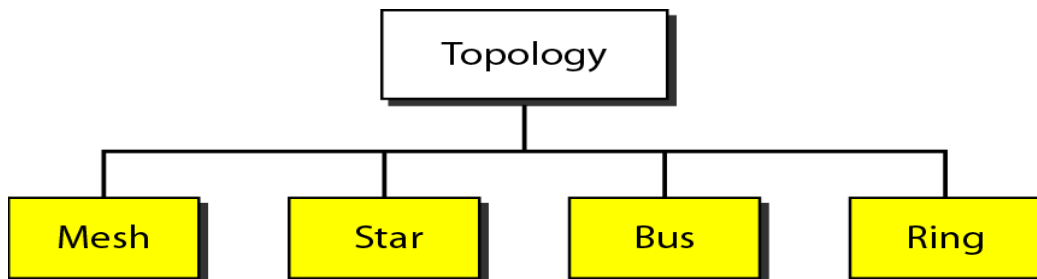
Mesh Topology- In a mesh topology, every device has a dedicated point-to-point link to every other device. The term dedicated means that the link carries traffic only between the two devices it connects.

We need n (n − 1) physical links in a fully connected mesh network with n nodes. if each physical link allows communication in both directions (duplex mode),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 Figure) to be connected to the other n − 1 stations.

Advantages-

 1.Use of dedicated links guarantees that each connection can carry its own data load, thus eliminating the traffic problems that can occur when links must be shared by multiple devices.

 2. A mesh topology is robust. If one link becomes unusable, it does not incapacitate the entire system.

3. Privacy or security. When every message travels along a dedicated line, only the intended recipient sees it. Physical boundaries prevent other users from gaining access to messages.

4. Point-to-point links make fault identification and fault isolation easy. Traffic can be routed to avoid links with suspected problems. This facility enables the network manager to discover the precise location of the fault and aids in finding its cause and solution.
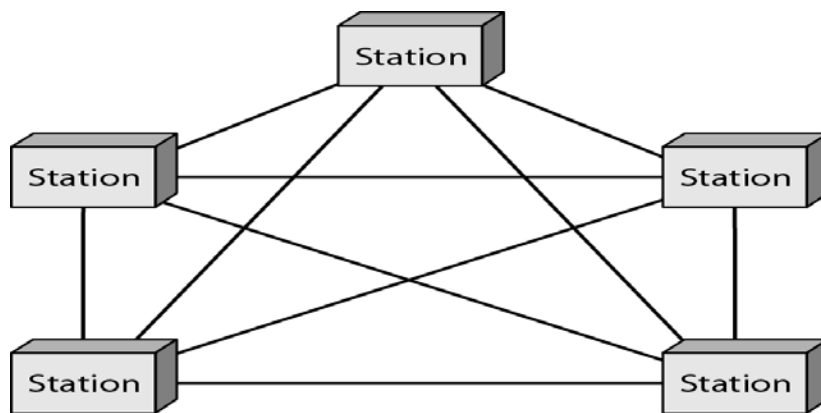


A fully connected mesh topology (five devices)

 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.

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.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
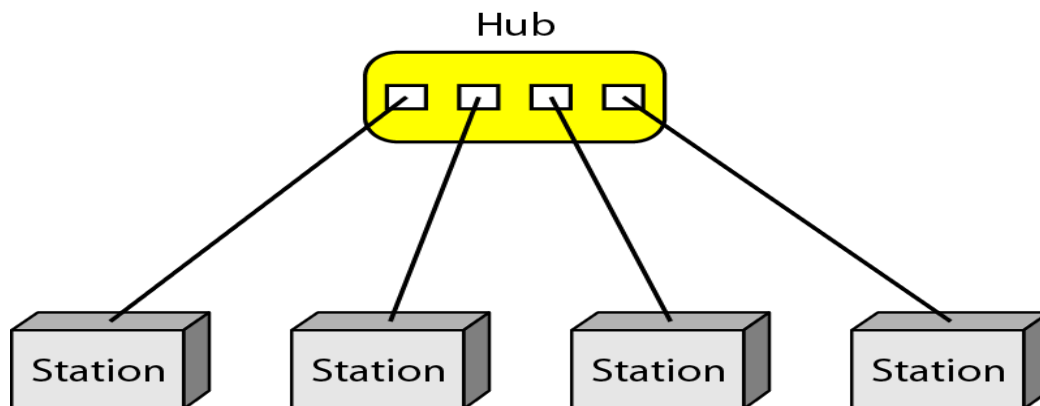
A star topology connecting four stations

Advantages

1. A star topology is less expensive than a mesh topology.

2. Each device needs only one link and one I/O port to connect . This factor makes it easy to install and reconfigure. Far less cabling needs to be housed, and additions, moves, and deletions involve only one connection: between that device and the hub.

3. robust. If one link fails, only that link is affected. All other links remain active. This factor also lends itself to easy fault identification and fault isolation. As long as the hub is working, it can be used to monitor link problems and bypass defective links.

Disadvantage

1. The dependency of the whole topology on one single point, the hub. If the hub goes down, the whole system is dead.

2. more cabling is required in a star than in some other topologies (such as ring or bus).

The star topology is used in local-area networks (LANs), High-speed LANs often use a star topology with a central hub.

Bus Topology

A bus topology, is multipoint. One long cable acts as a backbone to link all the devices in a network
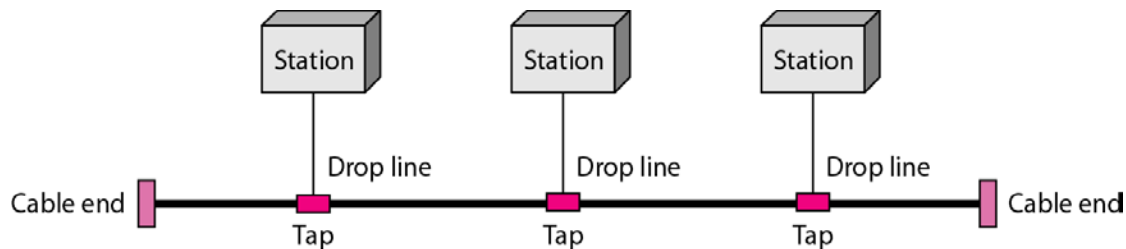
Nodes are connected to the bus cable by drop lines and taps. A drop line is a connection running between the device and the main cable.

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

1. Easy to install.

2. bus uses less cabling than mesh or star topologies. Only the backbone cable stretches through the entire facility. Each drop line has to reach only as far as the nearest point on the backbone.

Disadvantages

1. Difficult reconnection and fault isolation.

2. Difficult to add new devices.

3. Signal reflection at the taps can cause degradation in quality. This degradation can be controlled by limiting the number and spacing of devices connected to the given length of the cable

4. Adding new devices require modification or replacement of the backbone.

5. a fault or break in the bus cable stops all transmission. The damaged area reflects signals back in the direction of origin, creating noise in both directions.

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, but they are less popular now .

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
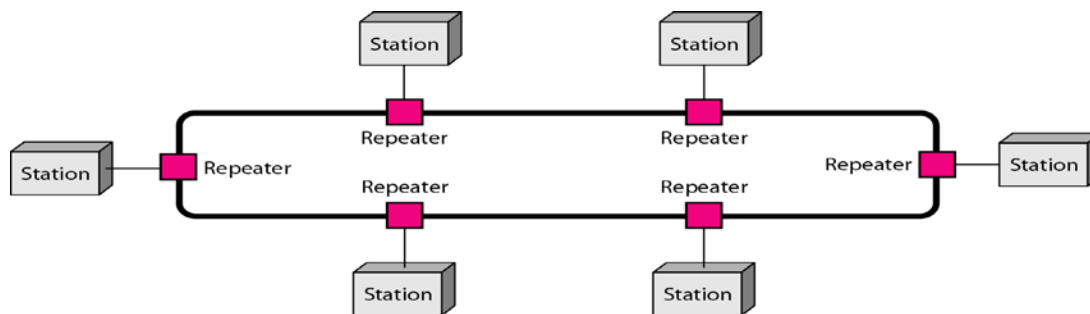

fig: A ring topology connecting six stations

Advantages

1. A ring is relatively easy to install and reconfigure. Each device is linked to only its immediate neighbors (either physically or logically). To add or delete a device requires changing only two connections.

2. Fault isolation is simplified.

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.

 Disadvantage

 In a simple ring, a break in the ring (such as a disabled station) can disable the entire network. This weakness can be solved by using a dual ring or a switch capable of closing off the break.

 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

Different types of networks

Local Area Network (LAN)-

a. LAN with a common cable (past)
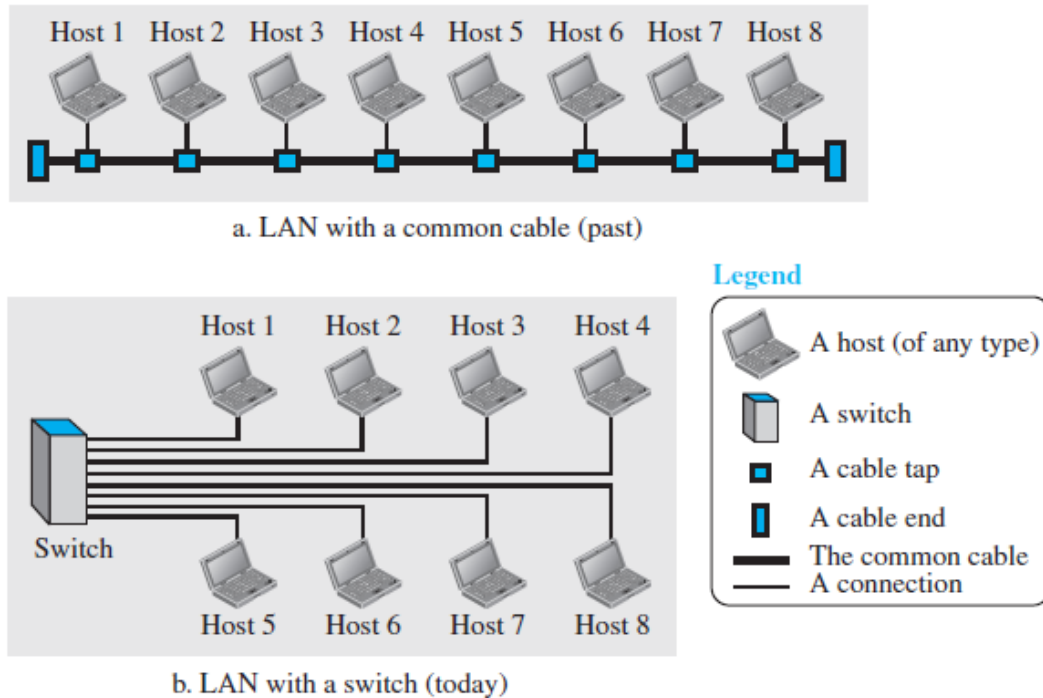
b. LAN with a switch (today)

fig: An isolated LAN in the past and today

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.

Wide Area Network

A wide area network (WAN) is also an interconnection of devices capable of communication. However, 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).
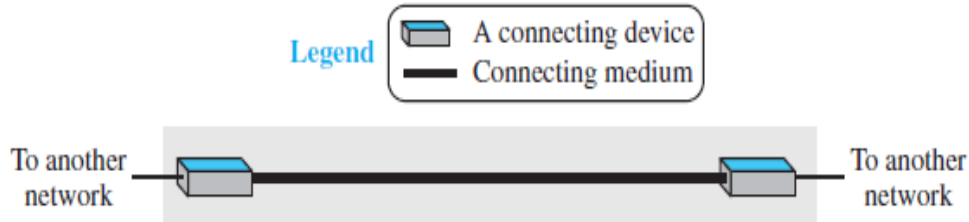

fig: Point-to-Point WAN

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..
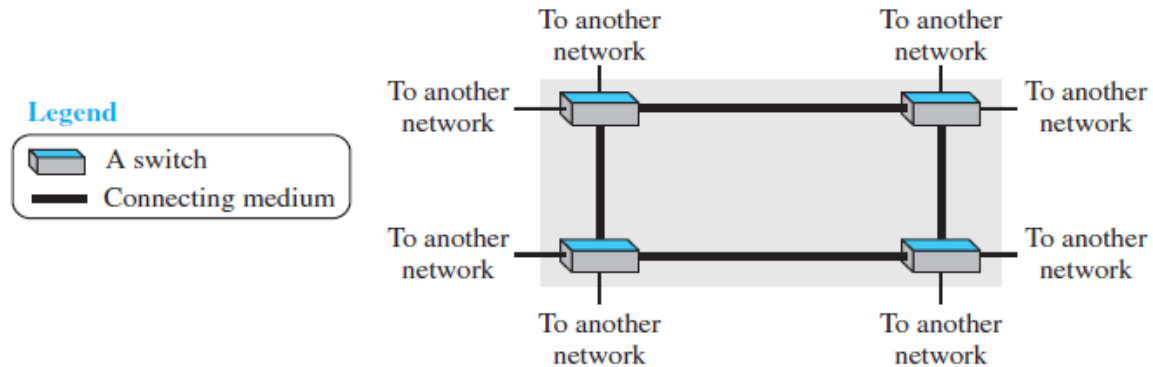
fig: A switched WAN

LAN VS WAN

1. A LAN is normally limited in size, spanning an office, a building, or a campus.	1. A WAN has a wider geographical span, spanning a town, a state, a country, or even the world
2. A LAN interconnects hosts	2. WAN interconnects connecting devices such as switches, routers, or modems
3. A LAN is normally privately owned by the organization that uses it	3. a WAN is normally created and run by communication companies and leased by an organization that uses it.

Internetwork

Today, it is very rare to see a LAN or a WAN in isolation; they are connected to one another. When two or more networks are connected, they make an internetwork, or internet.

 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 (with lowercase i). Communication between offices is now possible.
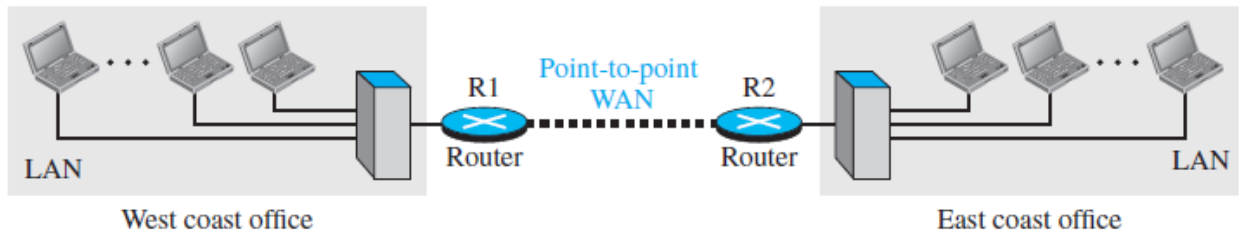
fig: An network made of two LAN and point-to-point dedicated WAN

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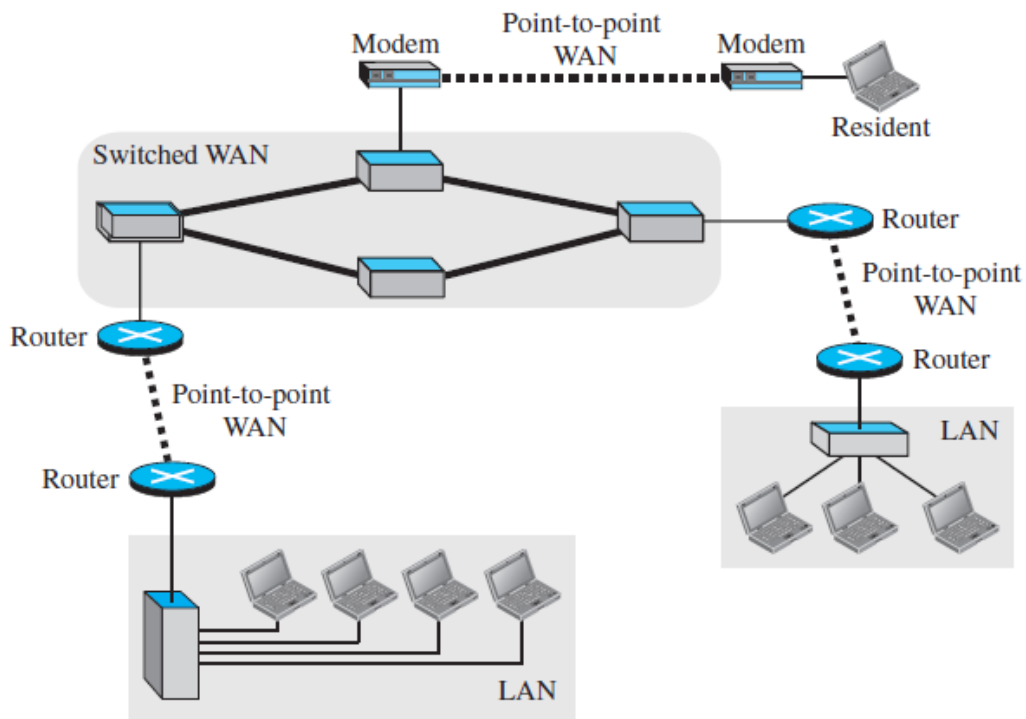

fig: A heterogeneous network made of four WANs and three LANs

Switching

An internet is a switched network in which a switch connects at least two links together. A switch needs to forward data from a network to another network when required. The two most common types of switched networks are circuit-switched and packet-switched networks.

Circuit-Switched Network

In a circuit-switched network, a dedicated connection, called a circuit, is always available between the two end systems; the switch can only make it active or inactive(continuous communication between two telephone). FIG shows a very simple switched network that connects four telephones to each end. We have used telephone sets instead of computers as an end system because circuit switching was very common in telephone networks in the past,
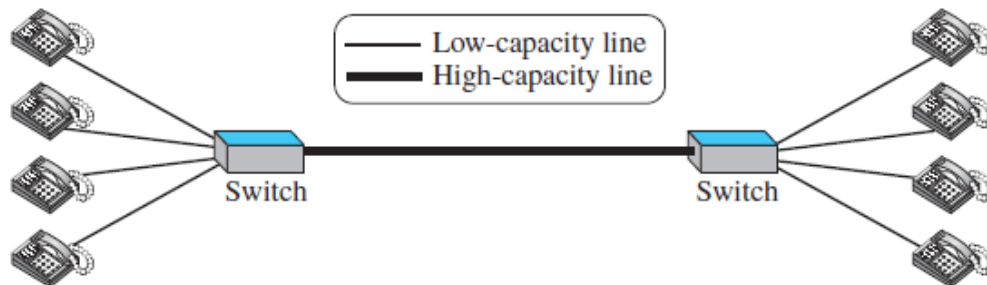

fig : Circuit-Switched Network

The thick line connecting two switches is a high-capacity communication line that can handle four voice communications at the same time; the capacity can be shared between all pairs of telephone sets. The switches used in this example have forwarding tasks but no storing capability.

Let us look at two cases.

In the first case, all telephone sets are busy; four people at one site are talking with four people at the other site; the capacity of the thick line is fully used.

In the second case, only one telephone set at one side is connected to a telephone set at the other side; only one-fourth of the capacity of the thick line is used. This means that a circuit-switched network is efficient only when it is working at its full capacity; most of the time, it is inefficient because it is working at partial capacity.

The reason to make the capacity of the thick line four times the capacity of each voice line is that we do not want communication to fail when all telephone sets at one side want to be connected with all telephone sets at the other side

Packet-Switched Network

In a computer network, the communication between the two computers is done in blocks of data called packets

This allows switches to function for both storing and forwarding because a packet is an independent entity that can be stored and sent later. Fig shows a small packet-switched network that connects four computers at one site to four computers at the other site.
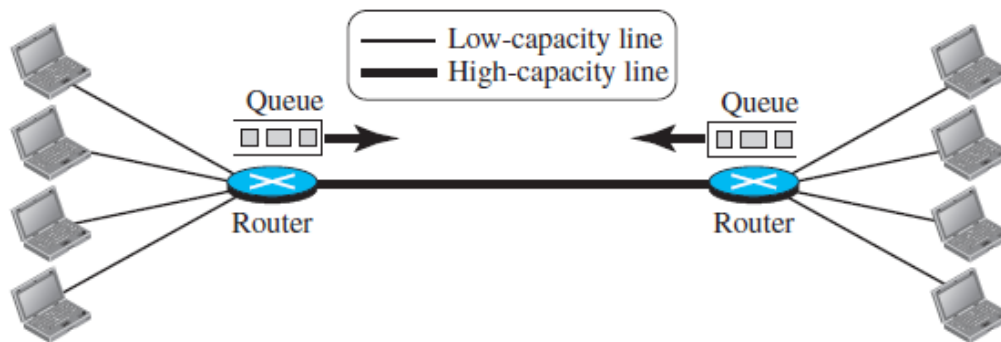

fig: **Packet-Switched Network**

A router in a packet-switched network has a queue that can store and forward the packet.

Example- Now assume that the capacity of the thick line is only twice the capacity of the data line connecting the computers to the routers.

 If only two computers (one at each site) need to communicate with each other, there is no waiting for the packets. However, if packets arrive at one router when the thick line is already working at its full capacity, the packets should be stored and forwarded in the order they arrived. The two simple examples show that a packet-switched network is more efficient than a circuit switched network, but the packets may encounter some delays.

The Internet

 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.

Figure 1.15 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.
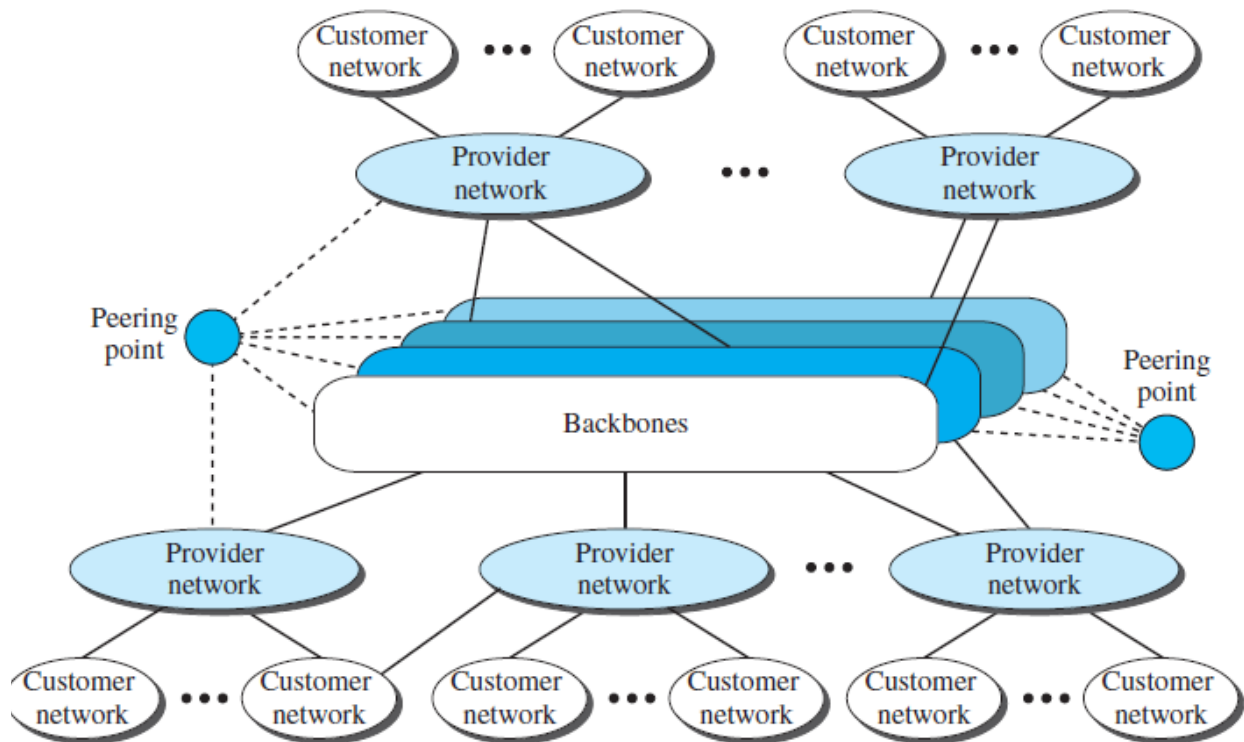
fig: The internet today

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 ISP

Network Models

Protocol Layering

Protocol 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.

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

Scenarios

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 Figure
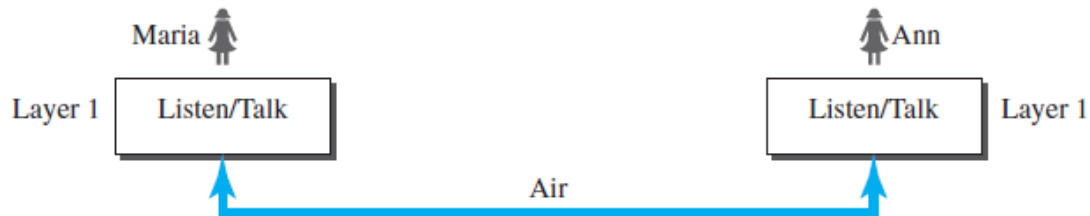

fig: single layer protocol

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.

Now we can say that the communication between Maria and Ann takes place in three layers, as shown in Figure . We assume that Ann and Maria each have three machines (or robots) that can perform the task at each layer.
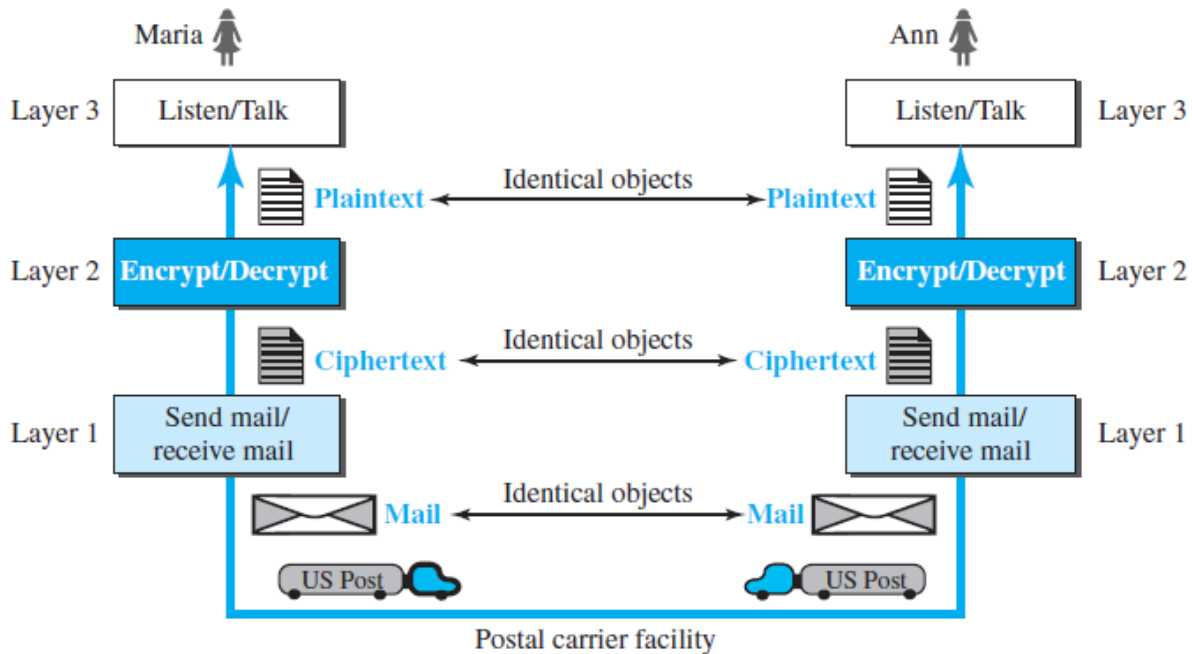

fig: A three layer protocol

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.

Need for protocol layering

1) Protocol layering enables us to divide a complex task into several smaller and simpler tasks.

For example, from fig, we could have used only one machine to do the job of all three machines. However, if 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.

2) 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

 1) Protocol layering allows to separate the services from the implementation. Lower layer 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.

 2) 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.

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 Figure . 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. We will see that the concept of logical connection will help us better understand the task of layering we encounter in data communication and networking.
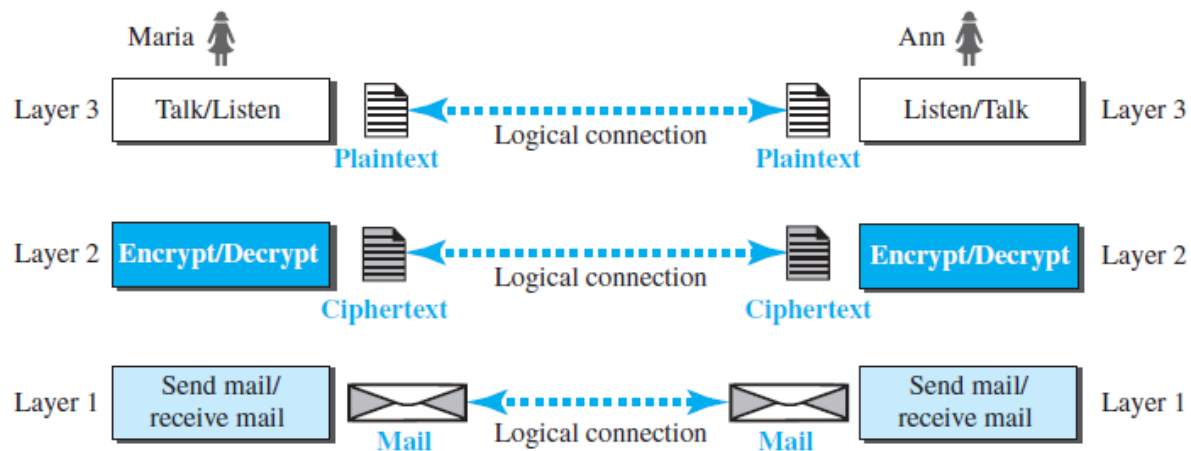

fig: Logical connection between peer layer

TCP/IP PROTOCOL SUITE

TCP/IP 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. Figure 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 Figure

fig: layers in TCP/IP protocol suite
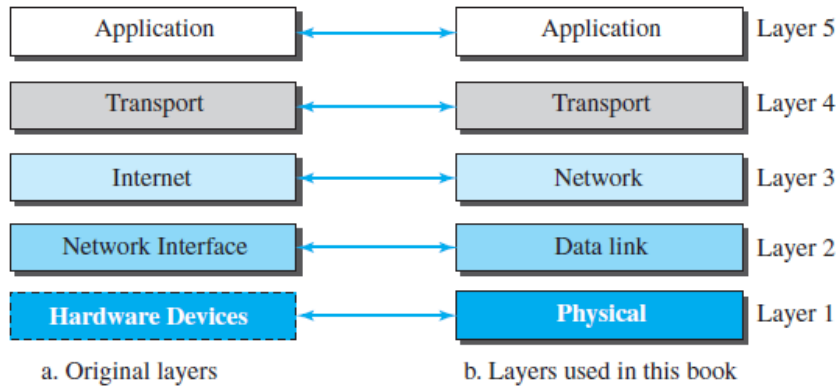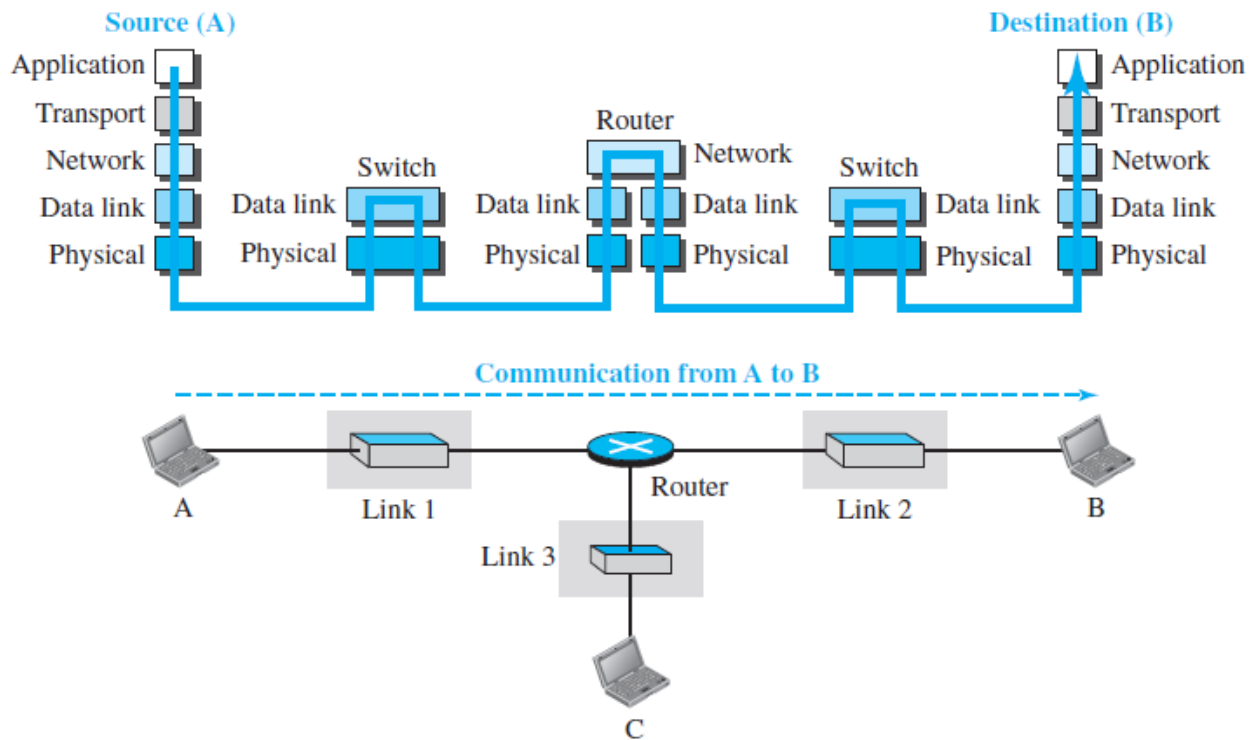

fig : Communication through an internet

Assume that computer A communicates with computer B. As the figure shows, five communicating devices in this communication: source host (computer A), the link-layer switch in link 1, the router, the link-layer switch in link 2, and the destination host (computer B).

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. 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

To better understand the duties of each layer, we need to think about the logical connections between layers.
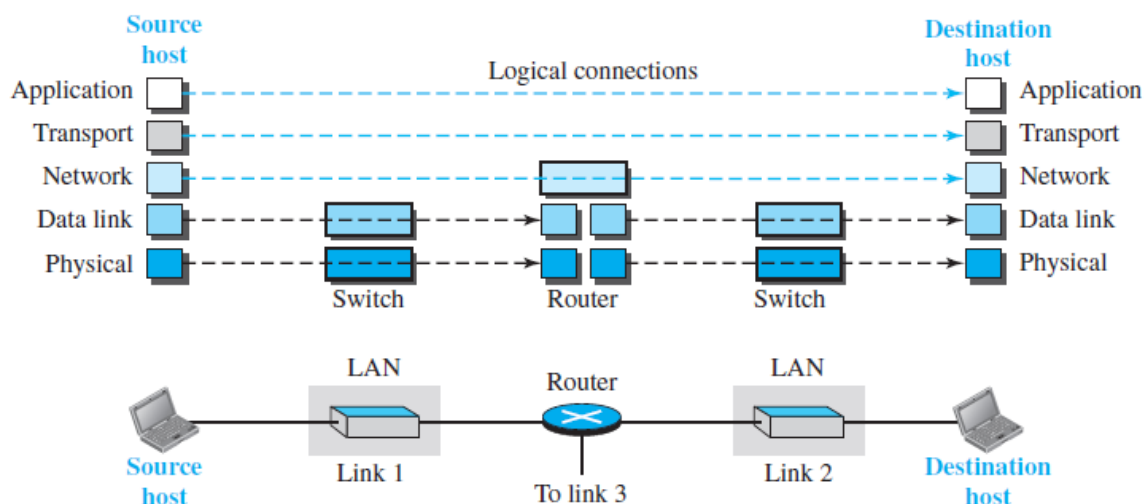
fig: Figure shows logical connections in our simple internet.

Using logical connections makes it easier 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.

Fig shows the second principle discussed previously for protocol layering. We show the identical objects below each layer related to each device.
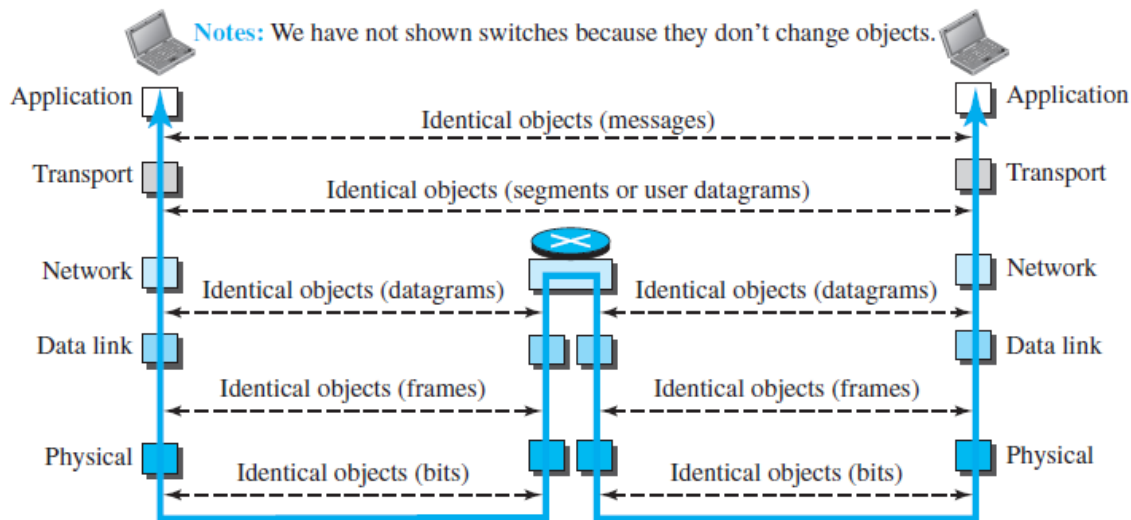

fig: identical objects in the TCP/IP protocol suite

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

Physical Layer

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). 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.

The physical layer of TCP/IP describes hardware standards such as IEEE 802.3, the specification for Ethernet network media, and RS-232, the specification for standard pin connectors.

The following are the main responsibilities of the physical layer

Definition of Hardware Specifications, Encoding and Signaling, Data Transmission and Reception,Topology and Physical Network Design

Data-link Layer

Internet is made up of several links (LANs and WANs) connected by routers. The data-link layer is responsible for taking the datagram and moving it across the link.(node to node communication)

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. The data-link layer takes a datagram and encapsulates it in a packet called a **frame**.

Each link-layer protocol provide a different service like framing, Flow control, Error control and congestion control.

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.

The network layer is responsible packetizing and routing and forwarding the packet through possible routes. others services are error and flow control, congestion control.

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 multitasking. 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.

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.

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.(**process to process coommunication**)

There are more than one protocol in the transport layer, which means that each application program can use the protocol that best matches its requirement. 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.

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.

Application Layer

As Figure shows, 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, 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.

1) The Hypertext Transfer Protocol (HTTP) is a vehicle for accessing the World Wide Web (WWW).

2) The Simple Mail Transfer Protocol (SMTP) is the main protocol used in electronic mail (e-mail) service.

3) The File Transfer Protocol (FTP) is used for transferring files from one host to another.

4)The Terminal Network (TELNET) and Secure Shell (SSH) are used for accessing a site remotely.

5) The Simple Network Management Protocol (SNMP) is used by an administrator to manage the Internet at global and local levels.

6) The Domain Name System (DNS) is used by other protocols to find the network-layer address of a computer.

7) The Internet Group Management Protocol (IGMP) is used to collect membership in a group.

Encapsulation and Decapsulation

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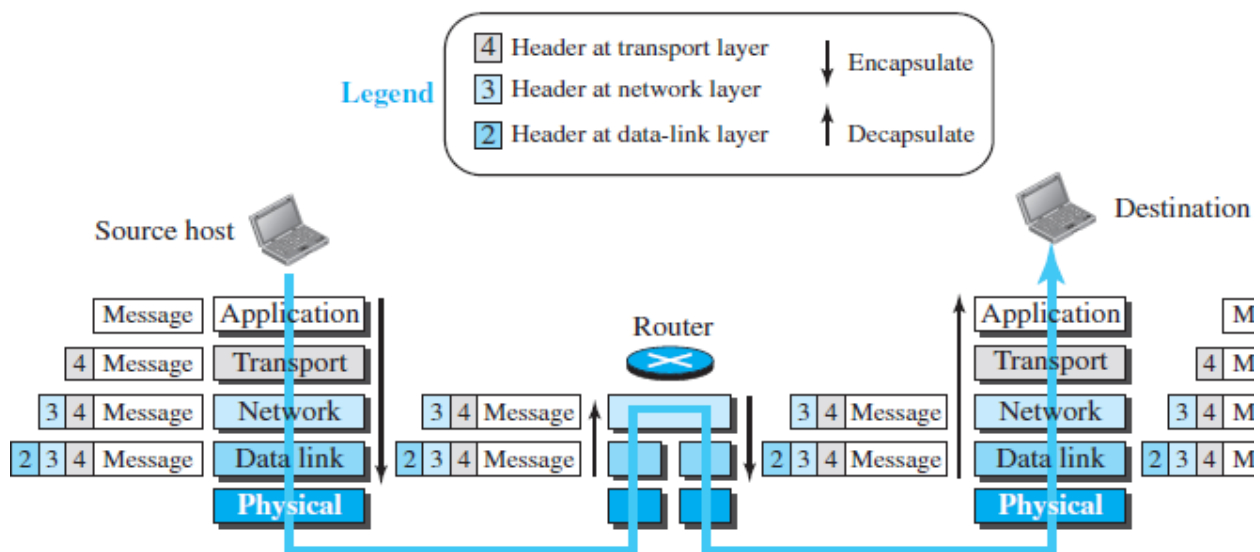

fig: **encapsulation/ decapsulation**

We have not shown the layers for the link-layer switches because no encapsulation/ decapsulation occurs in this device. Figure 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 payload, 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

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.

Figure 2.9 shows the addressing at each 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.
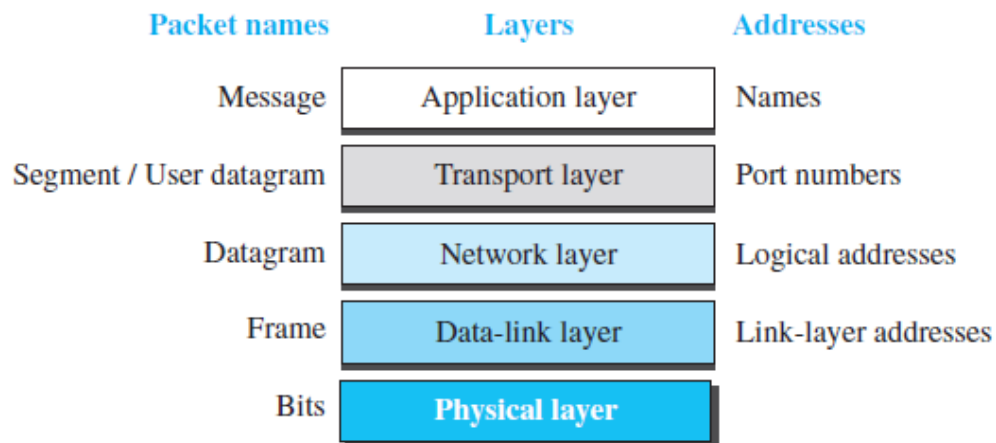
Packet names	Layers	Addresses
Message	Application layer	Names
Segment / User datagram	Transport layer	Port numbers
Datagram	Network layer	Logical addresses
Frame	Data-link layer	Link-layer addresses
Bits	Physical layer	

fig: Addressing in the TCP/IP Protocol suite

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

TCP/IP protocol suite uses several protocols at some layers, we have multiplexing at the source and demultiplexing at the destination.

Multiplexing 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). Figure shows the concept of multiplexing and demultiplexing at the three upper layers.
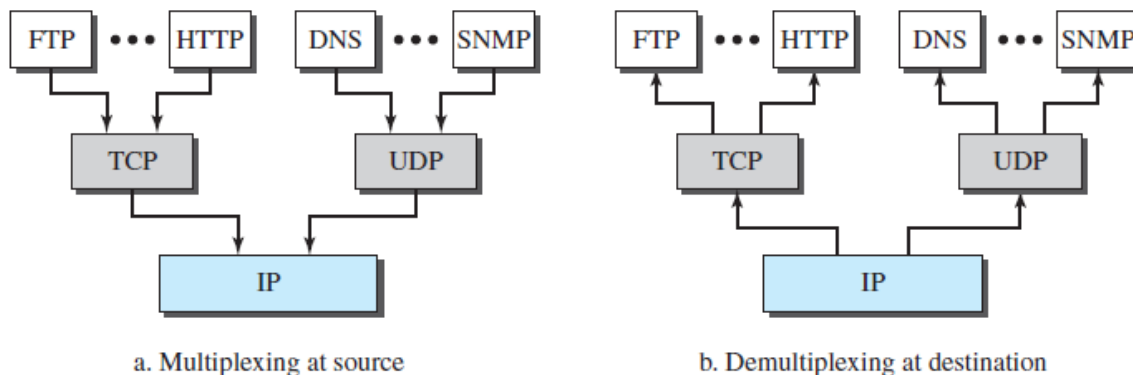

a. Multiplexing at source   b. Demultiplexing at destination

fig: multiplexing and demultiplexing

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

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
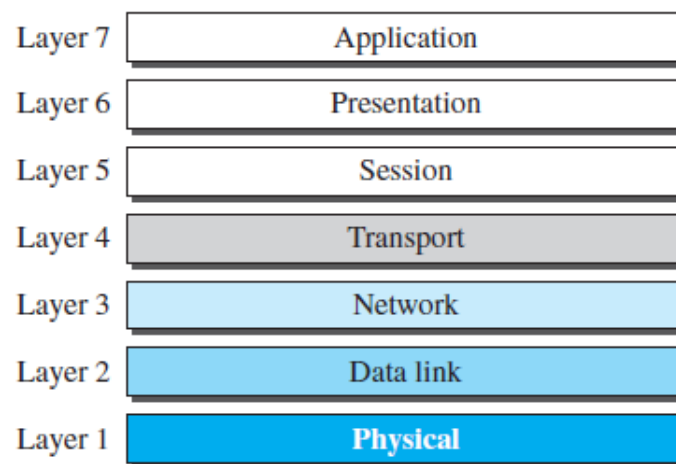
Layer 7	Application
Layer 6	Presentation
Layer 5	Session
Layer 4	Transport
Layer 3	Network
Layer 2	Data link
Layer 1	Physical

fig: OSI model

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 Figure .

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.
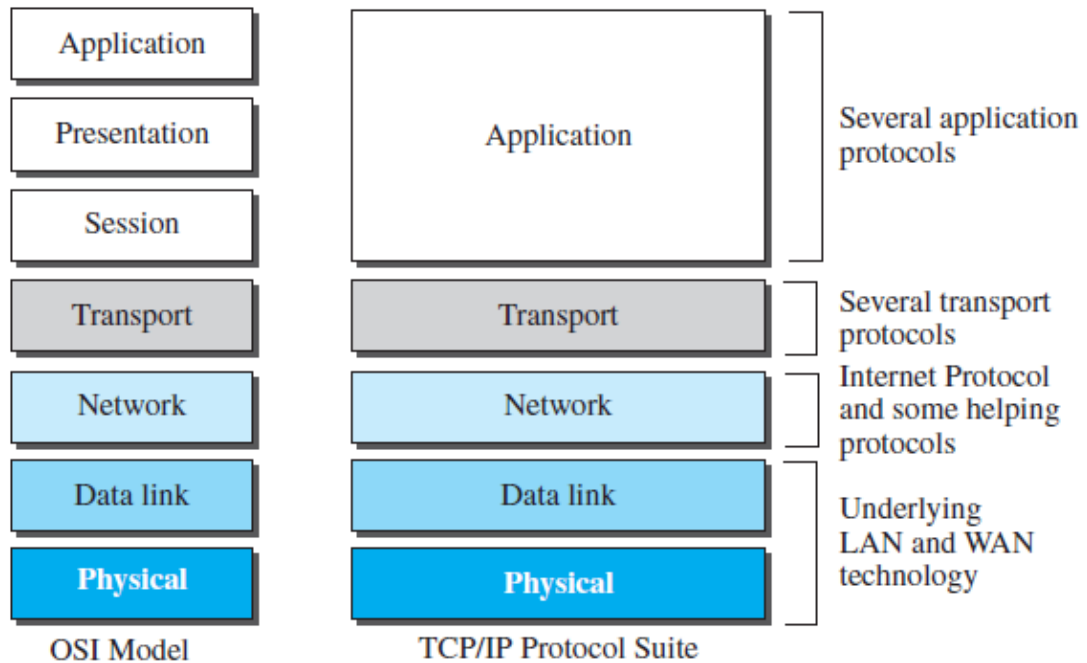

fig:TCP/IP and OSI model

DATA-LINK LAYER

INTRODUCTION

The Internet is a combination of networks glued together by connecting devices (routers or switches). If a packet is to travel from a host to another host, it needs to pass through these networks.. Communication at the data-link layer is made up of five separate logical connections between the data-link layers in the path.
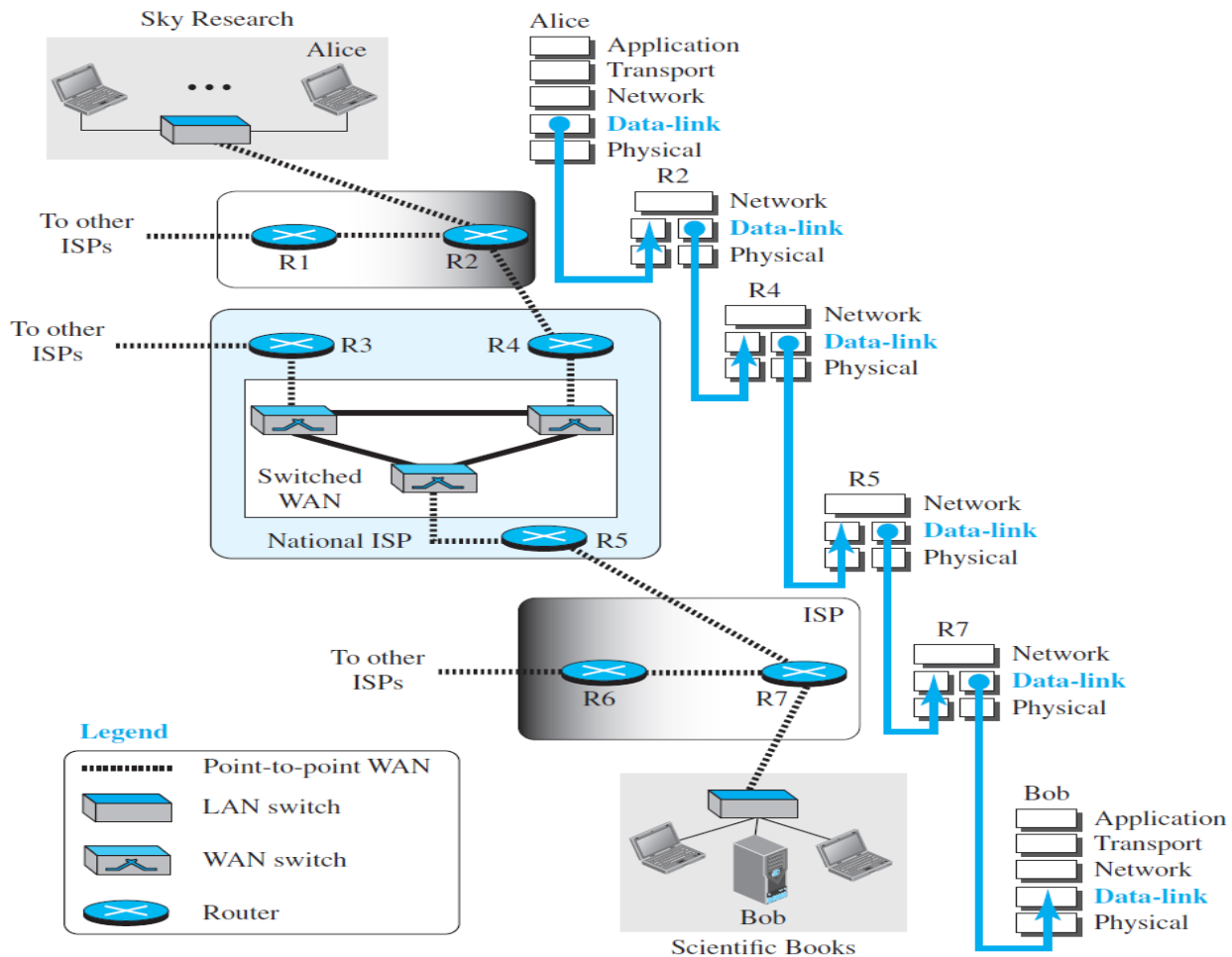

fig: communication at the data-link layer

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.

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. Theses 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. Figure shows a simple representation of links and nodes when the path of the data unit is only six nodes.
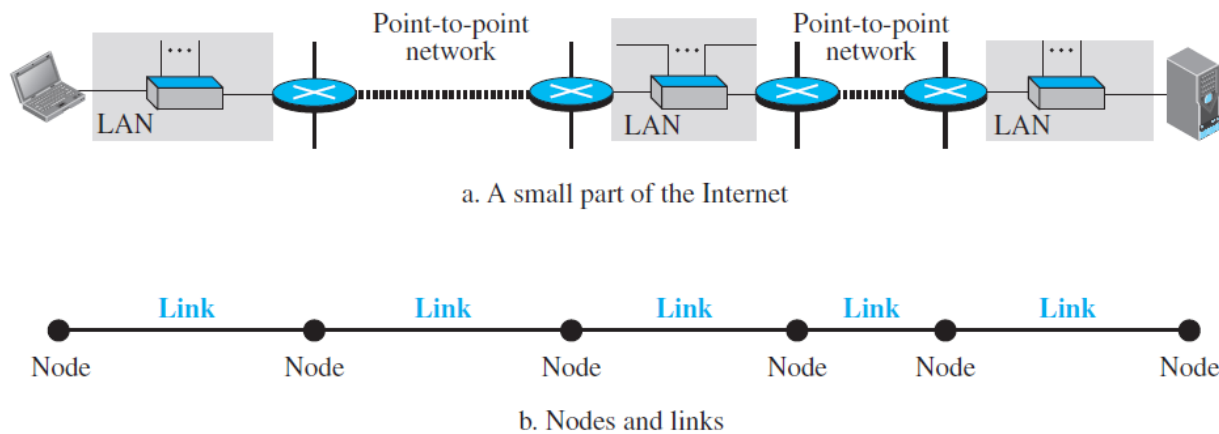

a. A small part of the Internet

b. Nodes and links

fig: links and nodes

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 data link layer provides services to the network layer; it receives services from the physical layer.

 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.

Analogy :-may help in this case. Assume a person needs to travel from her home to her friend's home in another city. The traveller 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 traveller 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 traveller is the same, but she uses three transporting tools to reach the destination. Figure 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.
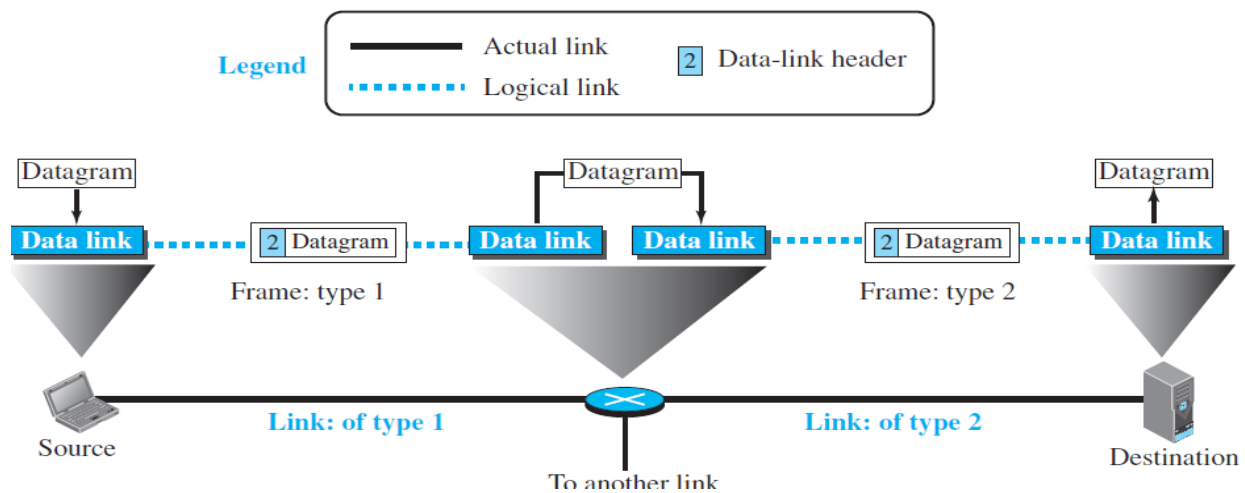
fig: communication with only three nodes

Framing

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, frame may have both a header and a trailer. Different data-link layers have different formats for framing.

Flow Control

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.

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).

Congestion Control

Although a link may be congested with frames, which may result in frame loss, most data-link-layer protocols do not directly use a congestion control to alleviate congestion, although some wide-area networks do. In general, congestion control is considered an issue in the network layer or the transport layer because of its end-to-end nature.

Two Categories of Links

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

The data-link layer is divided into two sublayers: data link control (DLC) and media access control (MAC). LAN protocols actually use the same strategy. **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 fig
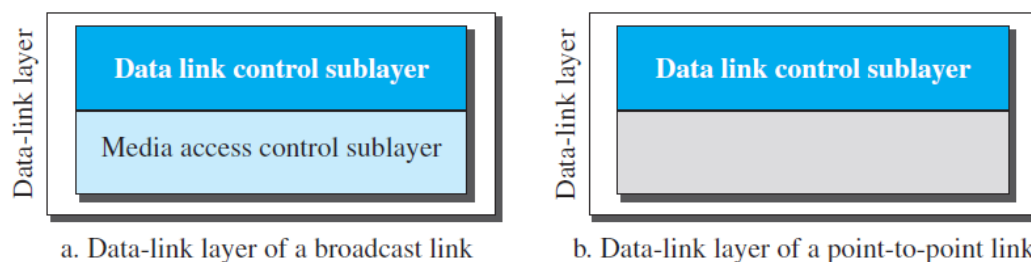

a. Data-link layer of a broadcast link   b. Data-link layer of a point-to-point link

fig: dividing the data link layer into two sublayers

LINK-LAYER ADDRESSING

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 (ICMP).

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. Figure demonstrates the concept in a small internet.
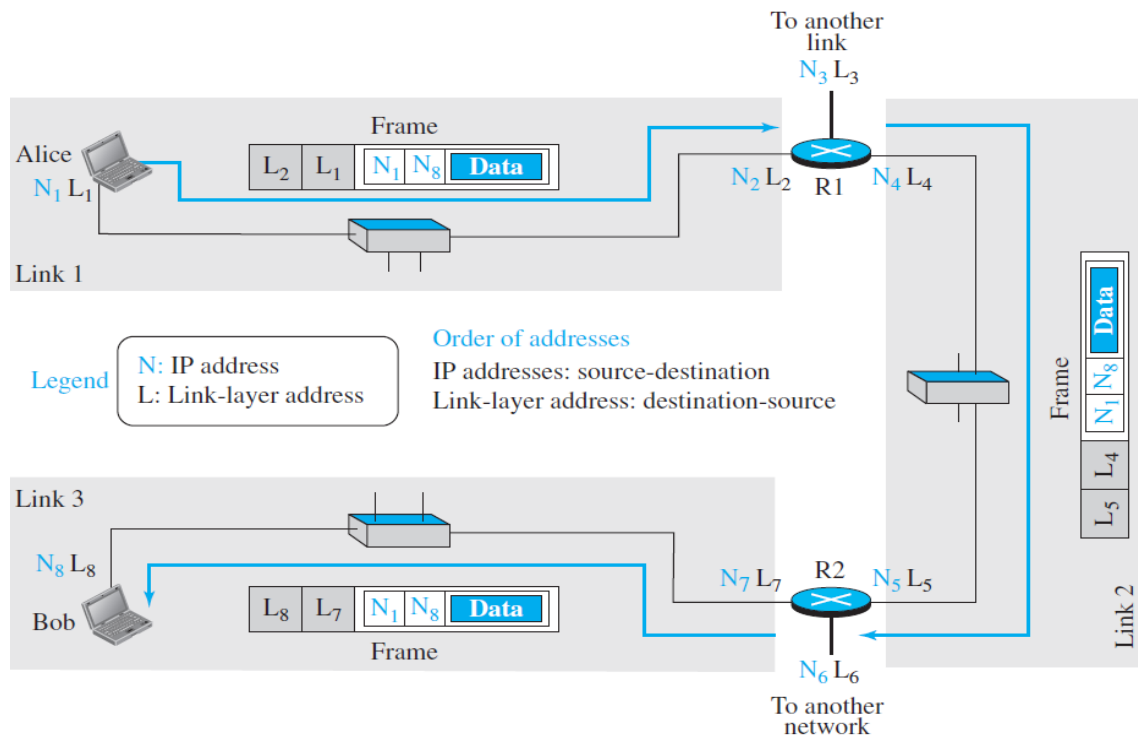

fig: IP address and link layer addresses in a small internet

above Figure shows three links and two routers and also have only two hosts: Alice (source) and Bob (destination). For each host, two addresses, the IP addresses (N) and the link-layer addresses (L) are shown. 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 L8. 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:

❑ 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.

❑ What is the size of link-layer addresses? The answer is that it depends on the protocol used by the link. Although we have only one IP protocol for the whole Internet, we may be using different data-link protocols in different links.

 different link-layer protocols.

Three Types of addresses

Some link-layer protocols define three types of addresses: unicast, multicast, and broadcast.

Unicast Address: Each host or each interface of a router is assigned a unicast address. Unicasting means one-to-one communication. A frame with a unicast address destination is destined only for one entity in the link.

Example: The unicast link-layer addresses in the most common LAN, Ethernet, are 48 bits (six bytes) that are presented as 12 hexadecimal digits separated by colons; for example, the following is a link-layer address of a computer.

A2:34:45:11:92:F1

Multicast Address: Some link-layer protocols define multicast addresses. Multicasting means one-to-many communication. However, the jurisdiction is local (inside the link).

Example: the multicast link-layer addresses in the most common LAN, Ethernet, are 48 bits (six bytes) that are presented as 12 hexadecimal digits separated by colons. The second digit, however, needs to be an even number in hexadecimal. The following shows a multicast address: A3:34:45:11:92:F1

Broadcast Address: Some link-layer protocols define a broadcast address. Broadcasting means one-to-all communication. A frame with a destination broadcast address is sent to all entities in the link.

Example : the broadcast link-layer addresses in the most common LAN, Ethernet, are 48 bits, all 1s, that are presented as 12 hexadecimal digits separated by colons. The following shows a broadcast address: FF:FF:FF:FF:FF:FF

Address Resolution Protocol (ARP)

Anytime a node has an IP datagram to send to another node in a link, it has the IP address of the receiving node. The source host knows the IP address of the default router. Each router except the last one in the path gets the IP address of the next router by using its forwarding table. The last router knows the IP address of the destination host. However, the IP address of the next node is not helpful in moving a frame through a link; we need the link-layer address of the next node. This is the time when the Address Resolution Protocol (ARP) becomes helpful.

The ARP protocol is one of the auxiliary protocols defined in the network layer, as shown in Figure . It belongs to the network layer, it maps an IP address to a logical-link address. ARP accepts an IP address from the IP protocol, maps the address to the corresponding link-layer address, and passes it to the data-link layer.
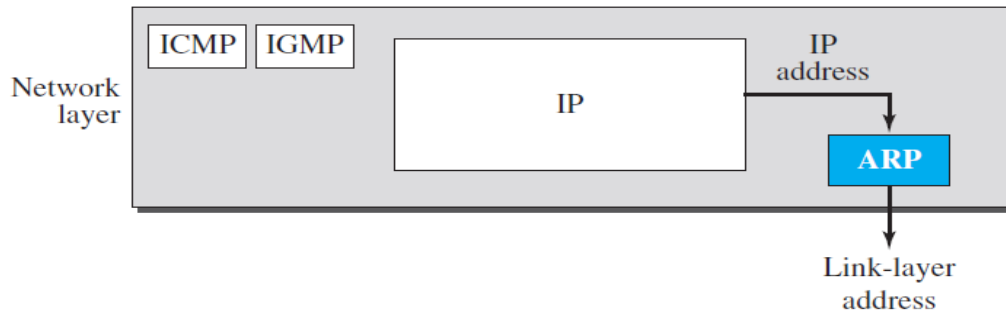
fig: Position of ARP in TCP/IP protocol suite

Anytime a host or a router needs to find the link-layer address of another host or router in its network, it sends an ARP request packet. The packet includes the link-layer and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the link-layer address of the receiver, the query is broadcast over the link using the link-layer broadcast address.
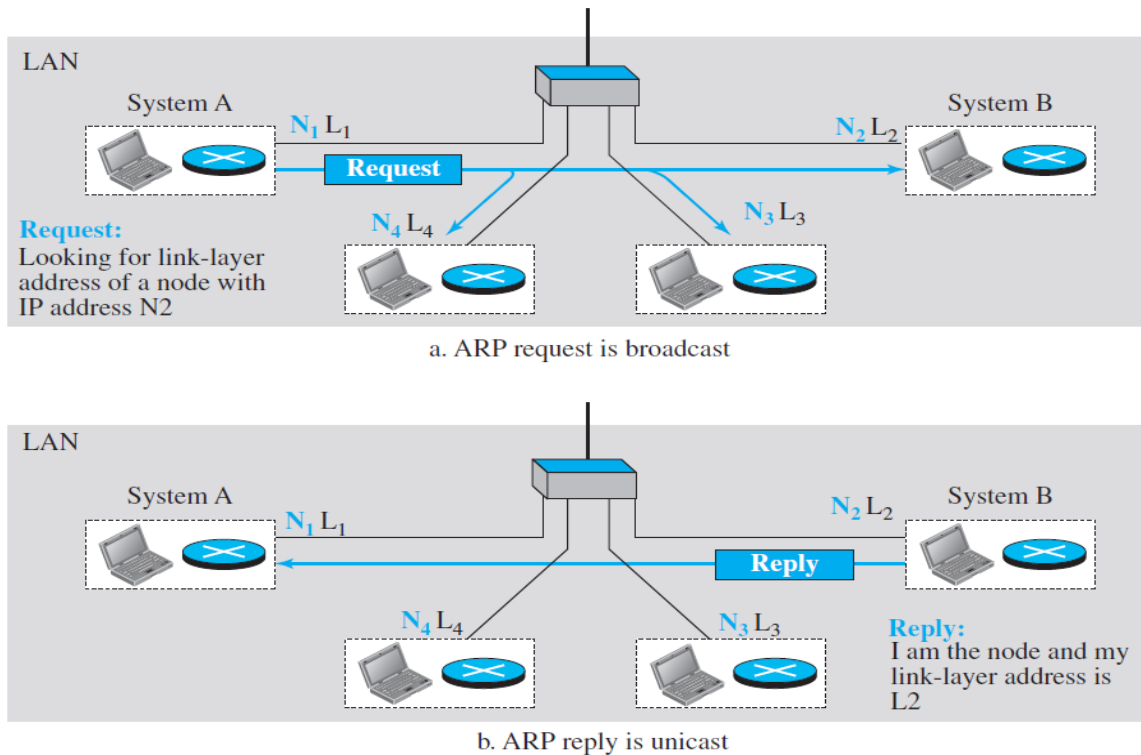

a. ARP request is broadcast

b. ARP reply is unicast

fig: ARP operation

Every host or router on the network receives and processes the ARP request packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet. The response packet contains the recipient's IP and link-layer addresses.

The packet is unicast directly to the node that sent the request packet. In Figure (a) the system on the left (A) has a packet that needs to be delivered to another system (B) with IP address N2. System A needs to pass the packet to its data-link layer for the actual delivery, but it does not know the physical address of the recipient. It uses the services of ARP by asking the ARP protocol to send a broadcast ARP request packet to ask for the physical address of a system with an IP address of N2. This packet is received by every system on the physical network, but only system B will answer it, as shown in Figure (b). System B sends an ARP reply packet that includes its physical address. Now system A can send all the packets it has for this destination using the physical address it received.

Caching

Let us assume that there are 20 systems connected to the network (link): system A, system B, and 18 other systems. We also assume that system A has 10 datagrams to send to system B in one second.

 a. Without using ARP, system A needs to send 10 broadcast frames. Each of the 18 other systems need to receive the frames, decapsulate the frames, remove the datagram and pass it to their network-layer to find out the datagrams do not belong to them.This means processing and discarding 180 broadcast frames.

b. Using ARP, system A needs to send only one broadcast frame. Each of the 18 other systems need to receive the frames, decapsulate the frames, remove the ARP message and pass the message to their ARP protocol to find that the frame must be discarded. This means processing and discarding only 18 (instead of 180) broadcast frames. After system B responds with its own data-link address, system A can store the link-layer address in its cache memory. The rest of the nine frames are only unicast. Since processing broadcast frames is expensive (time consuming), the first method is preferable.

Packet Format

Figure shows the format of an ARP packet.

 The hardware type field - defines the type of the link-layer protocol; Ethernet given the type 1. The protocol type field - defines the network-layer protocol: IPv4 protocol is (0x0800).

The source hardware and source protocol addresses are variable-length fields defining the link-layer and network-layer addresses of the sender.

The destination hardware address and destination protocol address fields define the receiver link-layer and network-layer addresses.

An ARP packet is encapsulated directly into a data-link frame. The frame needs to have a field to show that the payload belongs to the ARP and not to the network-layer datagram
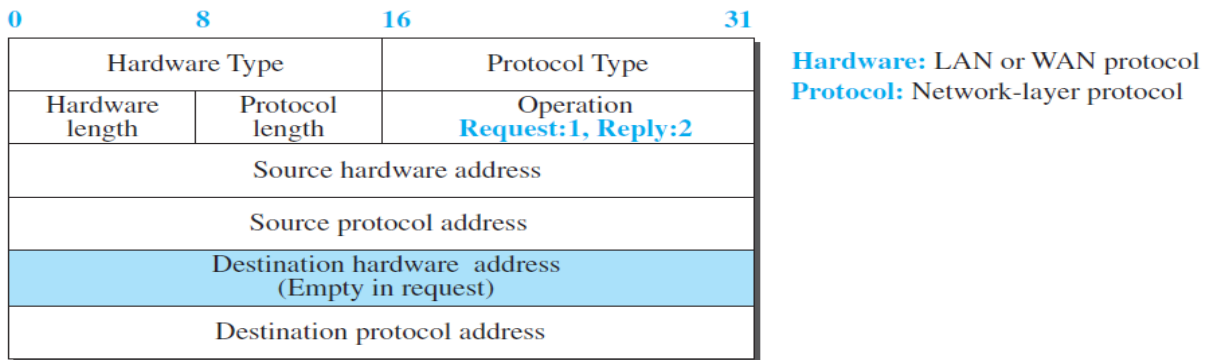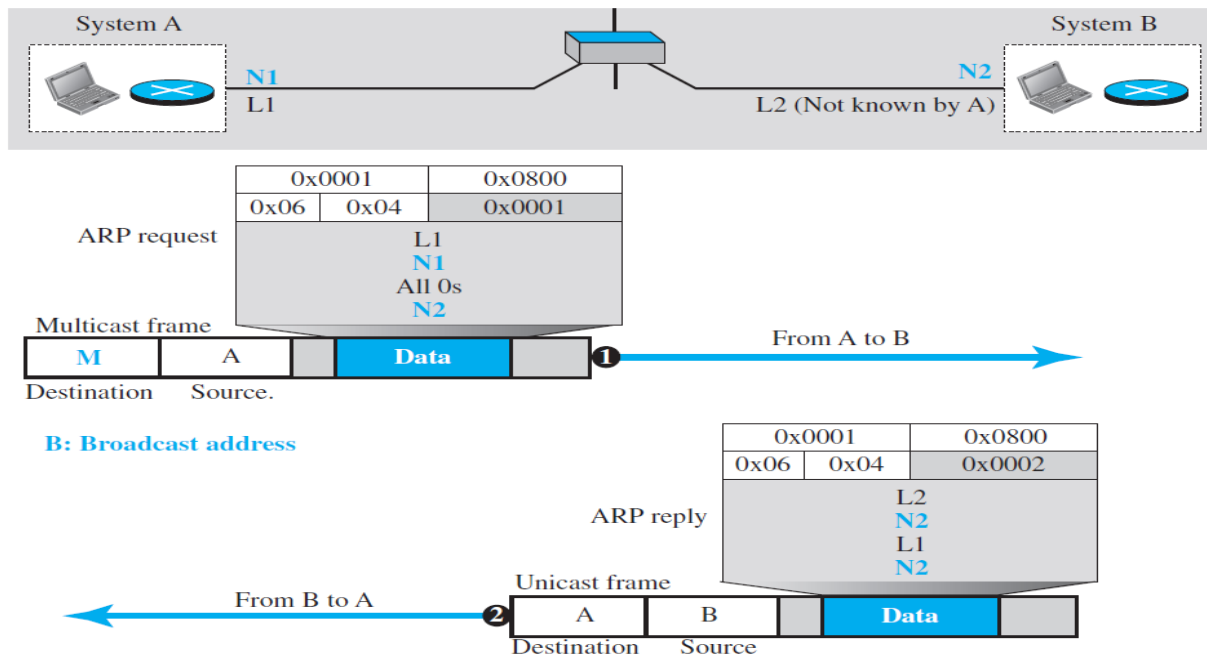

fig: ARP packet

Example : A host with IP address N1 and MAC address L1 has a packet to send to another host with IP address N2 and physical address L2 (which is unknown to the first host). The two hosts are on the same network. Figure shows the ARP request and response messages

DATA LINK CONTROL

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.

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.

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.

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 (Asynchronous transfer mode)ATM WAN, which uses frames of fixed size called cells.

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.

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. Figure shows the format of a frame in a character-oriented protocol.
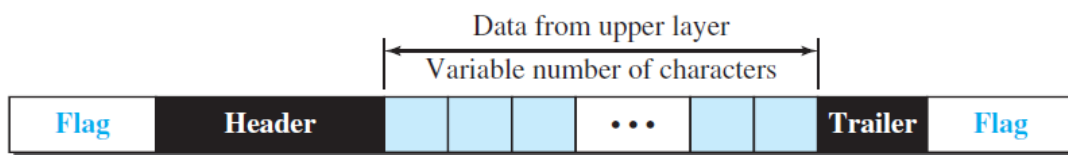
fig: A frame in a character-oriented protocol

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. Figure shows the situation.
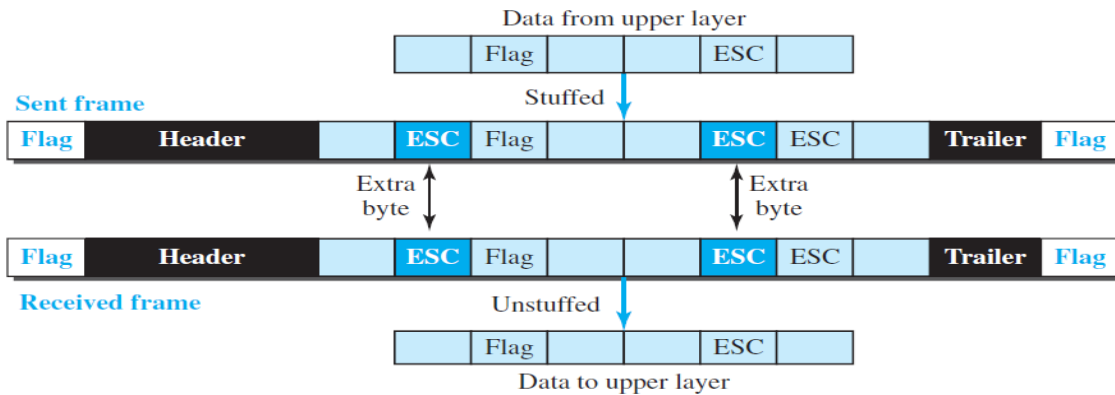

**fig: Byte stuffing and unstuffing**

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 fig**
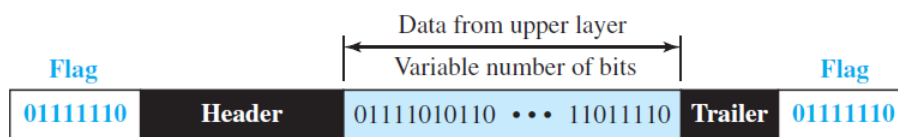
fig: A frame in bit-oriented protocol

If the flag pattern appears in the data, 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.

Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

Figure 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.
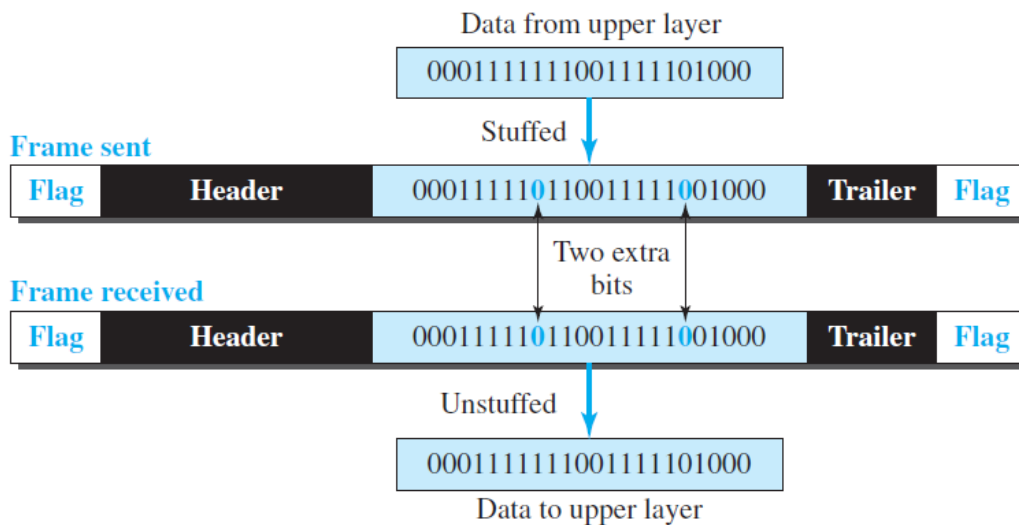

fig: Bit stuffing and unstuffing

Flow and Error Control

One of the responsibilities of the data-link control sublayer is flow and error control at the data-link layer.

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.
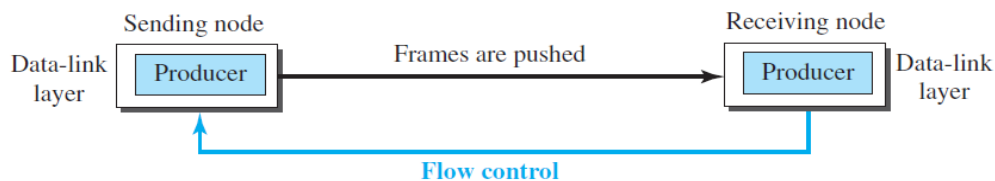

fig: Flow control at the data link layer

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.

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.

Combination of Flow and Error Control

Flow and error control can be combined. In a simple situation, the acknowledgment that is sent for flow control can also be used for error control to tell the sender the packet has arrived uncorrupted. The lack of acknowledgment means that there is a problem in the sent frame. We show this situation when we discuss some simple protocols in the next section. A frame that carries an acknowledgment is normally called an ACK to distinguish it from the data frame.

Connectionless and Connection-Oriented

A DLC protocol can be either connectionless or connection-oriented..

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.

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.

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.

The behavior of a data-link-layer protocol can be better shown as a finite state machine (FSM). An FSM is thought of as a machine with a finite number of states. The figure shows a machine

with three states. There are only three possible events and three possible actions. The machine starts in state I. If event 1 occurs, the machine performs actions 1 and 2 and moves to state II. When the machine is in state II, two events may occur. If event 1 occurs, the machine performs action 3 and remains in the same state, state II. If event 3 occurs, the machine performs no action, but move to state I.
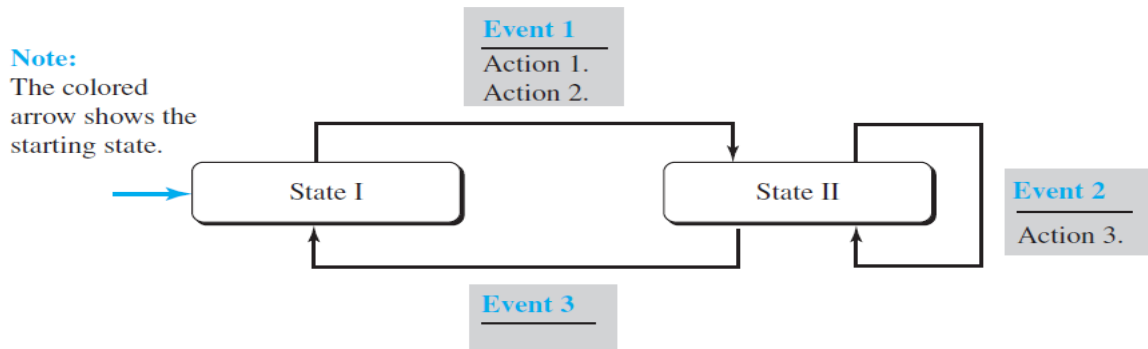

fig: Connectionless and connection oriented service represented as FSMs

Simple Protocol

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. **Figure** shows the layout for this protocol.
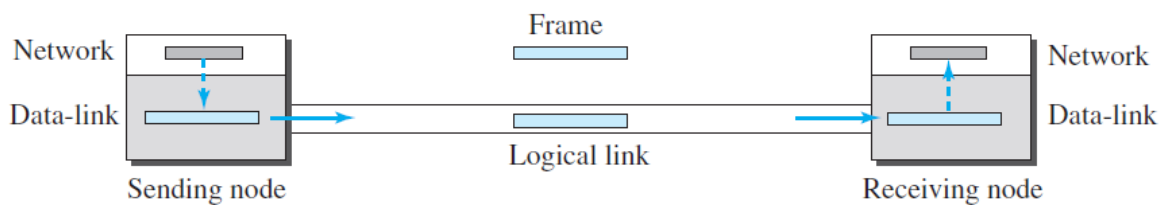

fig: Simple Protocol

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.

FSMs The sender site should not send a frame until its network layer has a message to send. The receiver site cannot deliver a message to its network layer until a frame arrives. We can show these requirements using two FSMs. Each FSM has only one state, the ready state.

sending machine remains in the ready state until a request comes from the process in the network layer. When this event occurs, the sending machine encapsulates the message in a frame and sends it to the receiving machine.

The receiving machine remains in the ready state until a Frame arrives from the sending machine. When this event occurs, the receiving machine decapsulates the message out of the frame and delivers it to the process at the network layer. **Figure** shows the FSMs for the simple protocol.
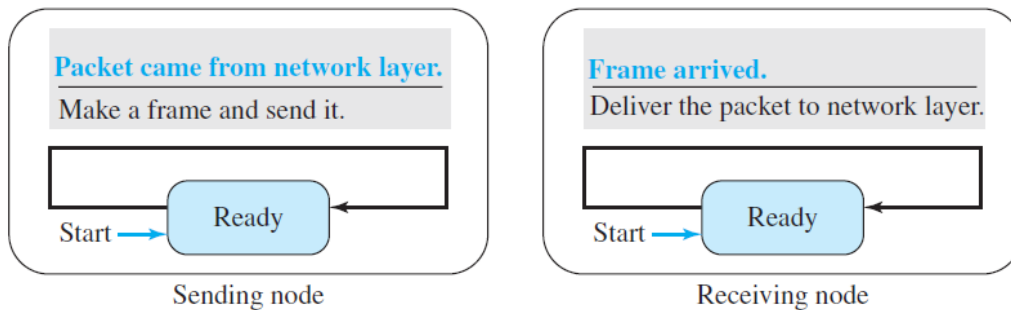

fig: FSMs for the simple protocol

Example

Flow diagram 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.
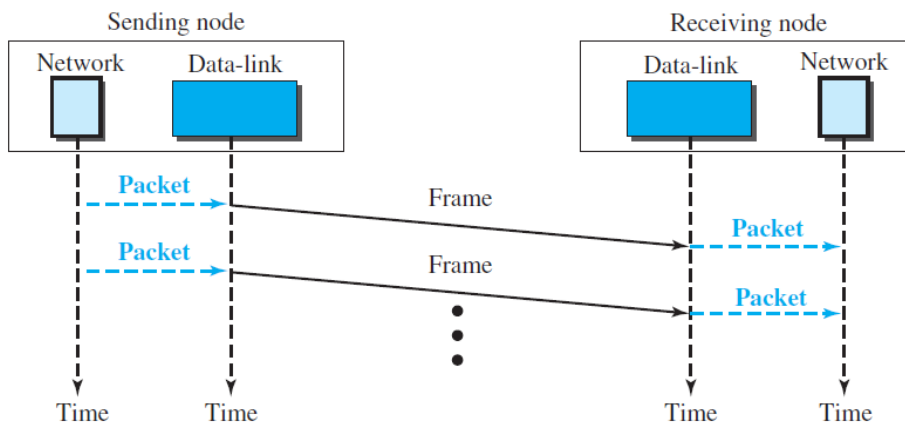

fig: Flow diagram for above example

Stop-and-Wait Protocol

Our second protocol is called the Stop-and-Wait protocol, which **uses both flow and error control**.

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. Figure 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.
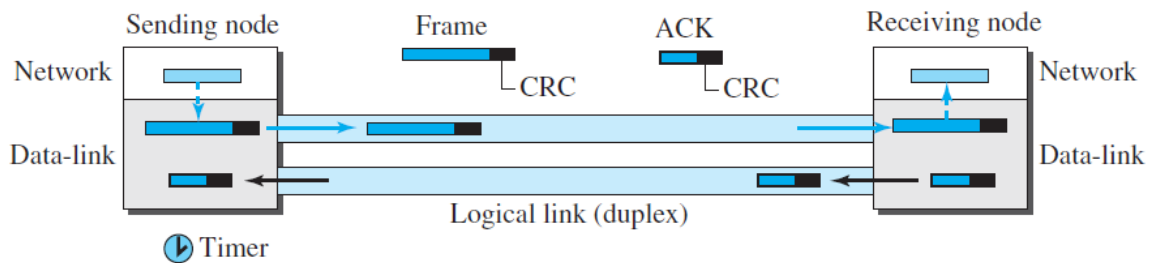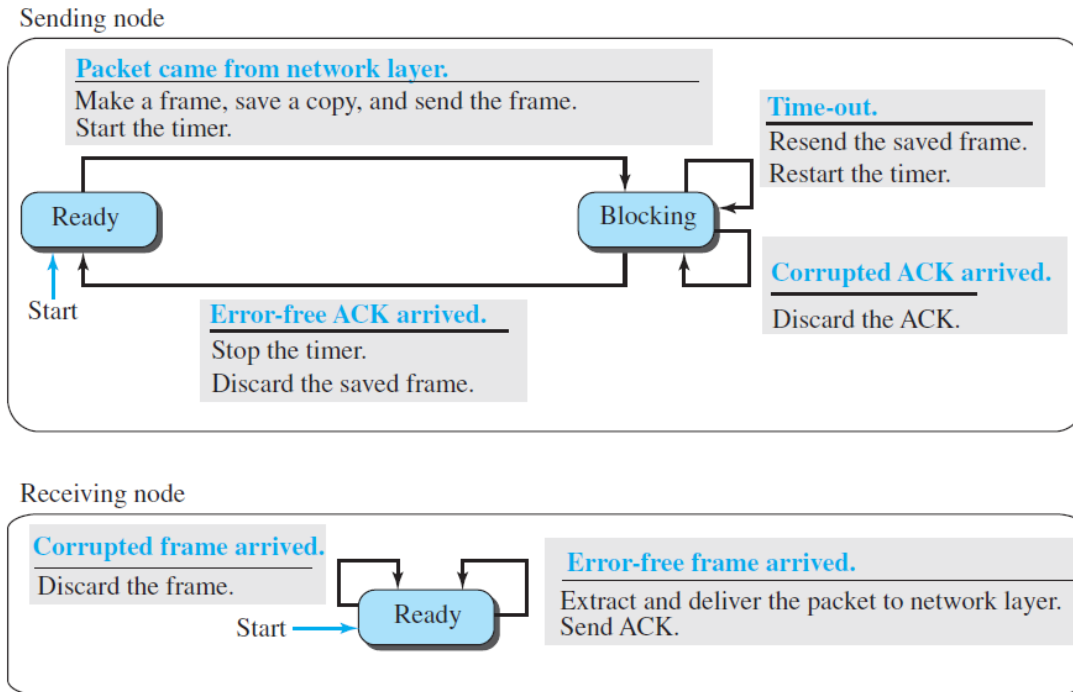

fig: Stop and Wait protocol

fig: FSMs for Stop-and-Wait protocol.

FSMs Figure shows the FSMs for primitive Stop-and-Wait protocol.

We describe the sender and receiver states below.

Sender States. The sender is initially in the ready state, but it can move between the ready and blocking state.

Ready State. When the sender is in this state, it is only waiting for a packet from the network layer. If a packet comes from the network layer, the sender creates a frame, saves a copy of the frame, starts the only timer and sends the frame. The sender then moves to the blocking state.

❑ **Blocking State.** When the sender is in this state, three events can occur:

a. If a time-out occurs, the sender resends the saved copy of the frame and restarts the timer.

 b. If a corrupted ACK arrives, it is discarded.

c. If an error-free ACK arrives, the sender stops the timer and discards the saved copy of the frame. It then moves to the ready state.

Receiver

The receiver is always in the ready state. Two events may occur:

a. If an error-free frame arrives, the message in the frame is delivered to the network layer and an ACK is sent.

b. If a corrupted frame arrives, the frame is discarded

Example

Flow diagram for this example is shown below.

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

Problem in above Example needs to be addressed and corrected. Duplicate packets, as much as corrupted packets, need to be avoided. 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, . . . ; the acknowledgment numbers can also be 1, 0, 1, 0, 1, 0, … 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.**
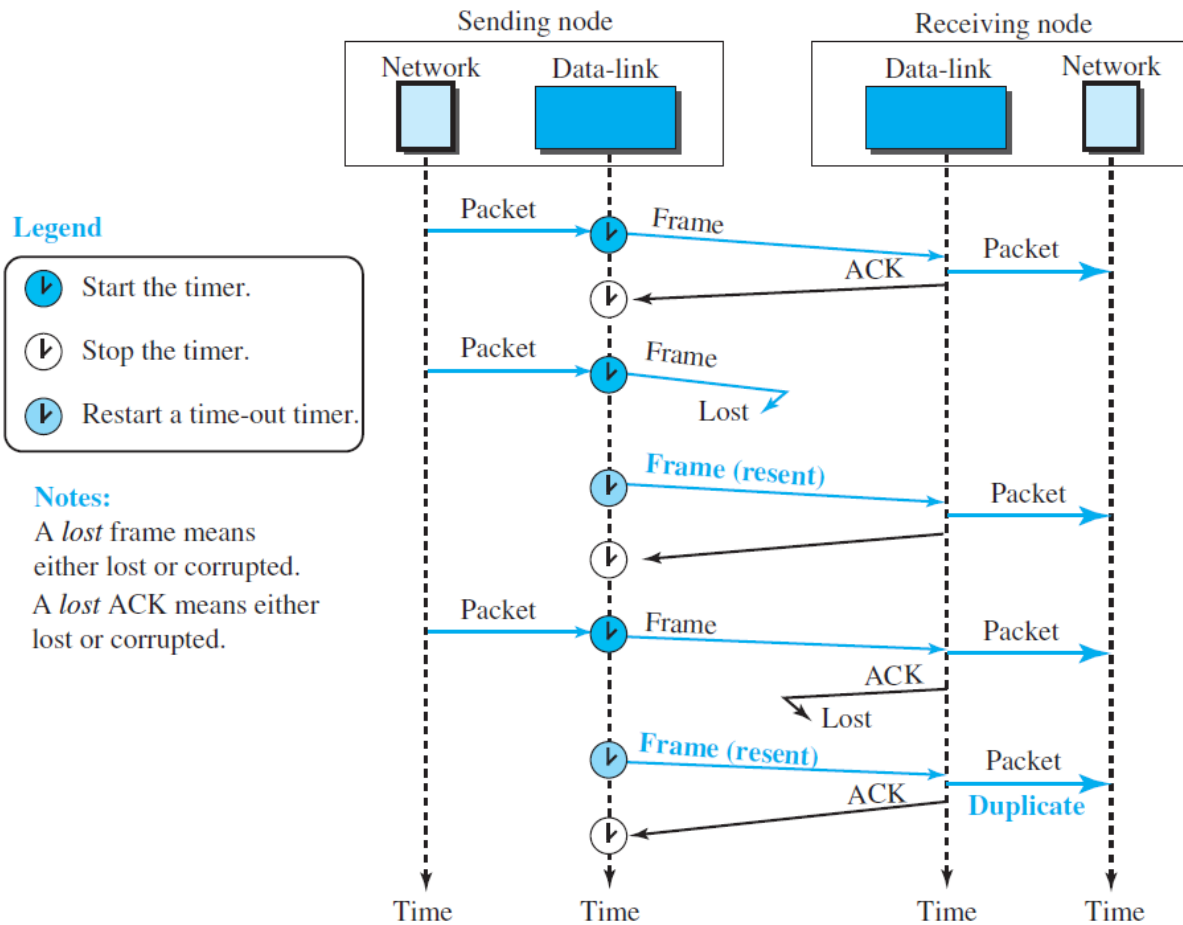
fig: flow diagram for example

example

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

FSMs with Sequence and Acknowledgment Numbers

We can change the FSM in Figure(FSM for stop and wait protocol) to include the sequence and acknowledgment numbers.
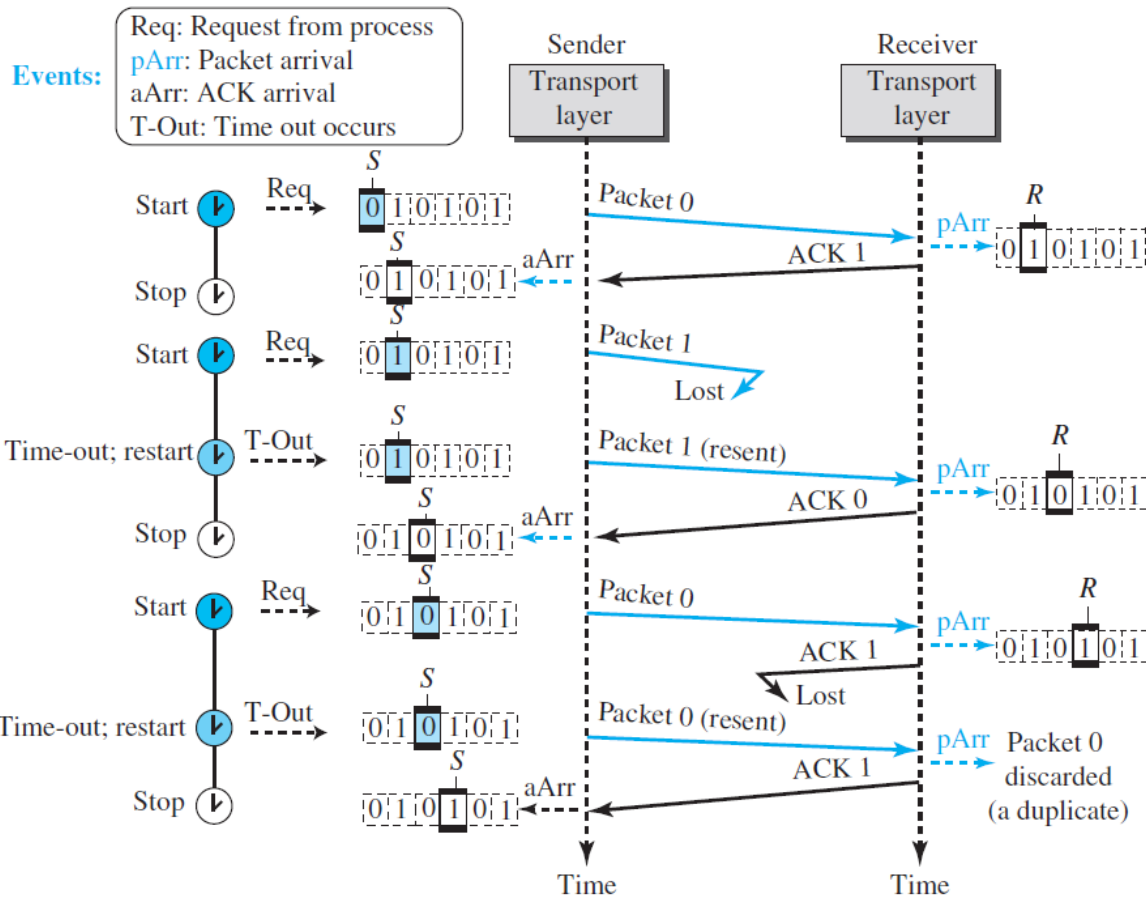
Piggybacking

The two protocols 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.