1702EC504 - COMPUTER NETWORKS

Prof S.Praveen Kumar M.E,(Ph.D), EMCAA, MISTE, IAENG.,
Assistant Professor/CSE
E.G.S Pillay Engineering College, Nagapattinam

9786465881
praveenkumar@egspec.org
Praveen2506.weebly.com

UNIT 1 - INTRODUCTION AND CONCEPTS OF NETWORKS

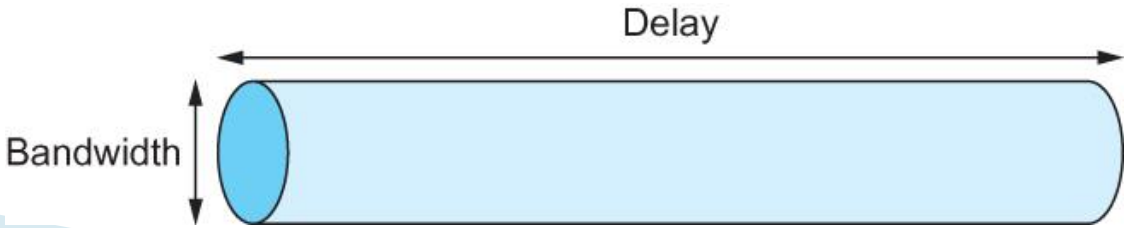
- **Networks**
  - Categories of Networks
  - Network hardware
  - Network software
- **Network Architecture**
  - Reference models
  - Network LAN technologies

- **Networks:**
  - A network is a set of devices (often referred to as *nodes) connected by communication* links.
  - A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network.
- **Computer Networks:**
  - Computer network is a set of computers (often referred to as nodes) connected by communication links

- **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 is often evaluated by two networking metrics:
  - Delay(Latency) and
  - Bandwidth.

- **Bandwidth** = Data rate supported by the network interface.
- **Latency** = Propagation + transmission + queue
- **Propagation Speed/Delay** = Time taken by the first bit of the packet to reach the receiver/receiver router.
- **Transmission Speed/Delay** = It is the time taken to push all the bits of a packet on to the link.
- **Queuing Delay** = It is the time that a packet has to wait in the queue before it can be transmitted over the link. Packets are put in the queue when the speed of the incoming link to the router is faster than the outgoing link.

Propagation time = Distance/Propagation speed



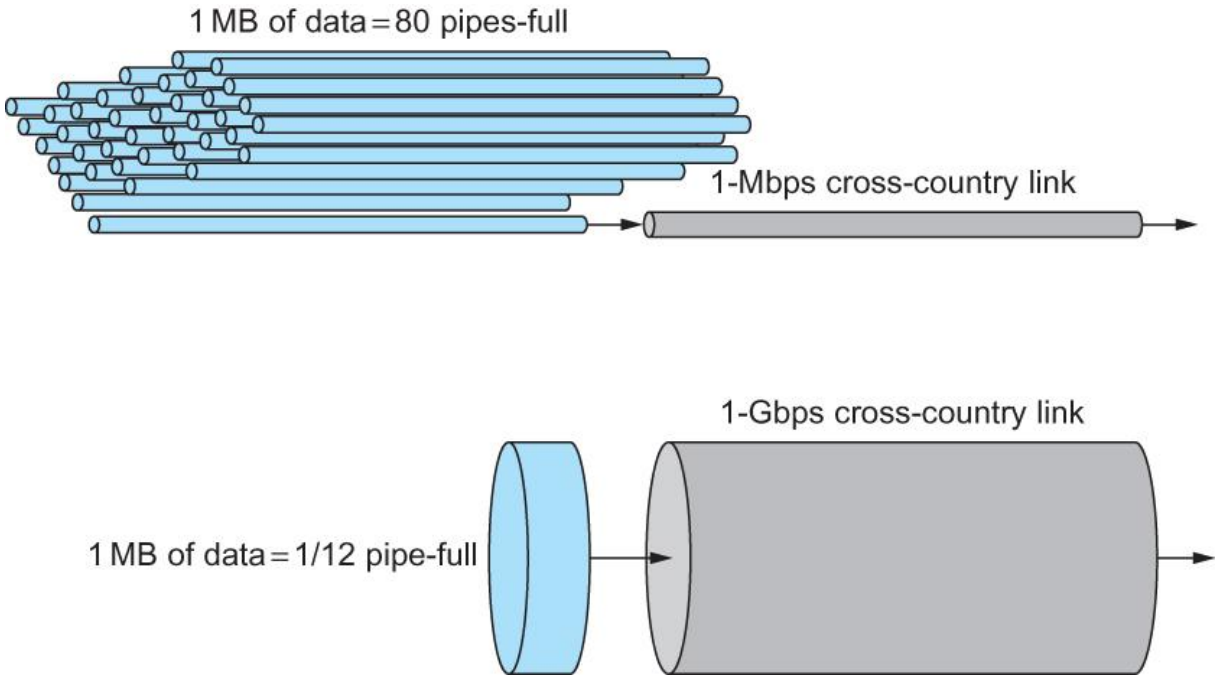
Network as a pipe

- **Relative importance of bandwidth and latency depends on application**
  - For large file transfer, bandwidth is critical
  - For small messages (HTTP, NFS, etc.), latency is critical
  - Variance in latency (jitter) can also affect some applications (*e.g.*, audio/video conferencing)

- **Infinite bandwidth**
  - RTT dominates
  - Throughput = TransferSize / TransferTime
  - TransferTime = RTT + 1/Bandwidth x TransferSize

- **Its all relative**
  - 1-MB file to 1-Gbps link looks like a 1-KB packet to 1-Mbps link

- Relationship between bandwidth and latency

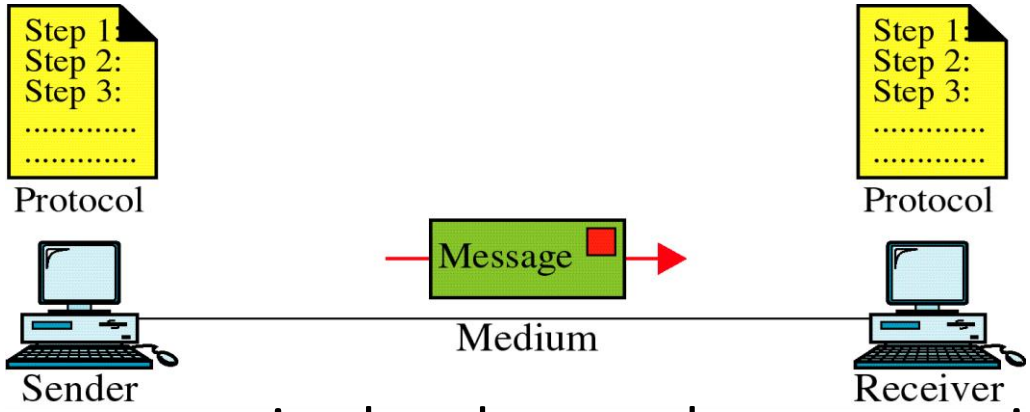


A 1-MB file would fill the 1-Mbps link 80 times, but only fill the 1-Gbps link 1/12 of one time

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

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

Components of Networks



I.Message: The message is the data to be communicated, which includes text, numbers, pictures, audio, and video.

2.Sender: The sender is the device that sends the data.

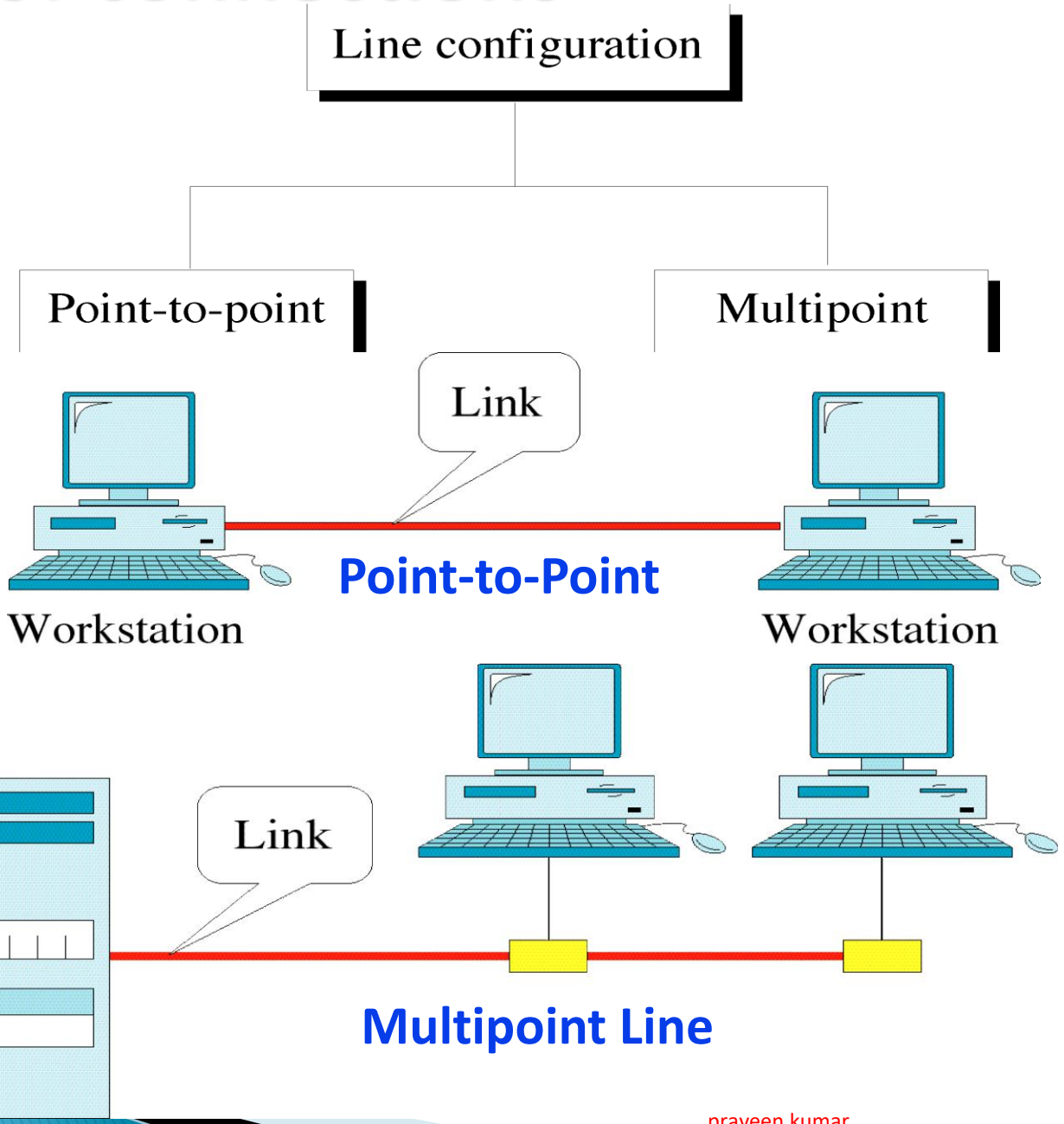
3.Receiver: The receiver is the device that receives the Data.

4.Transmission medium: The transmission medium is the physical path by which a message travels from sender to receiver.

Ex: Twisted-pair cable, coaxial cable, fiber-optic cable, radio waves….

5.Protocol: It 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 Tamil cannot be understood by a person who speaks only English.

Types of connections

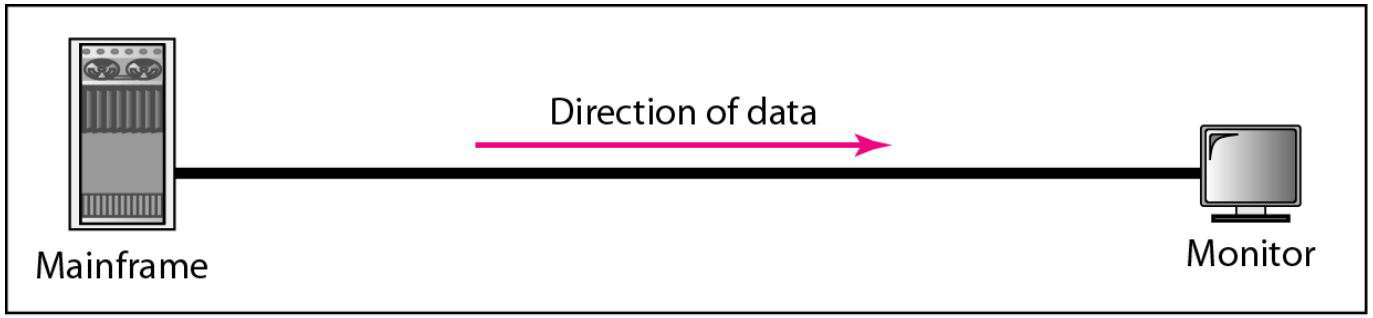


Types of connections

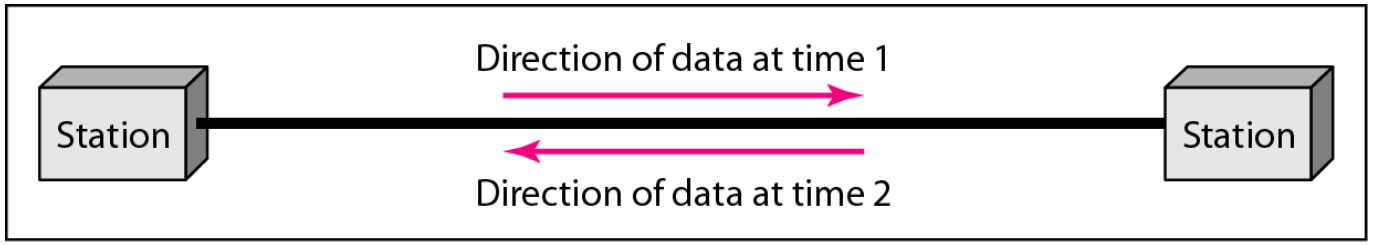
- **Point-to-Point:** A point-to-point connection provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices.
- **Ex:** When you change television channels by infrared remote control, you are establishing a point-to-point connection between the remote control and the television's control system.
- **Multipoint:** A multipoint (also called multi drop) connection is one in which more than two specific devices share a single link

Direction of data flow

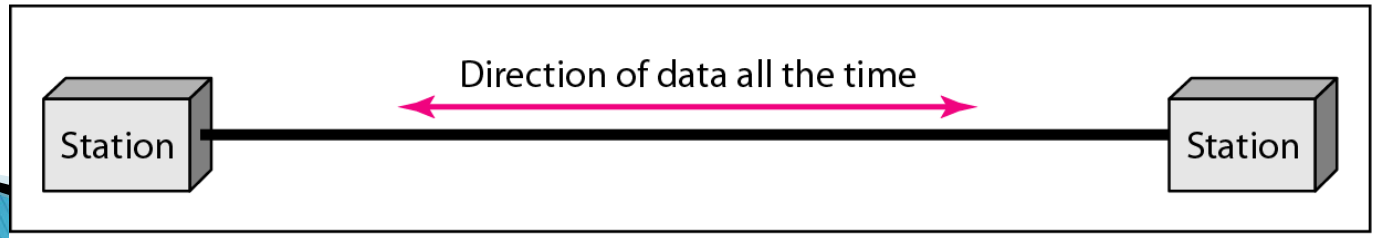
- Communication between two devices can be simplex, half-duplex, or full-duplex



a. Simplex



b. Half-duplex



c. Full-duplex

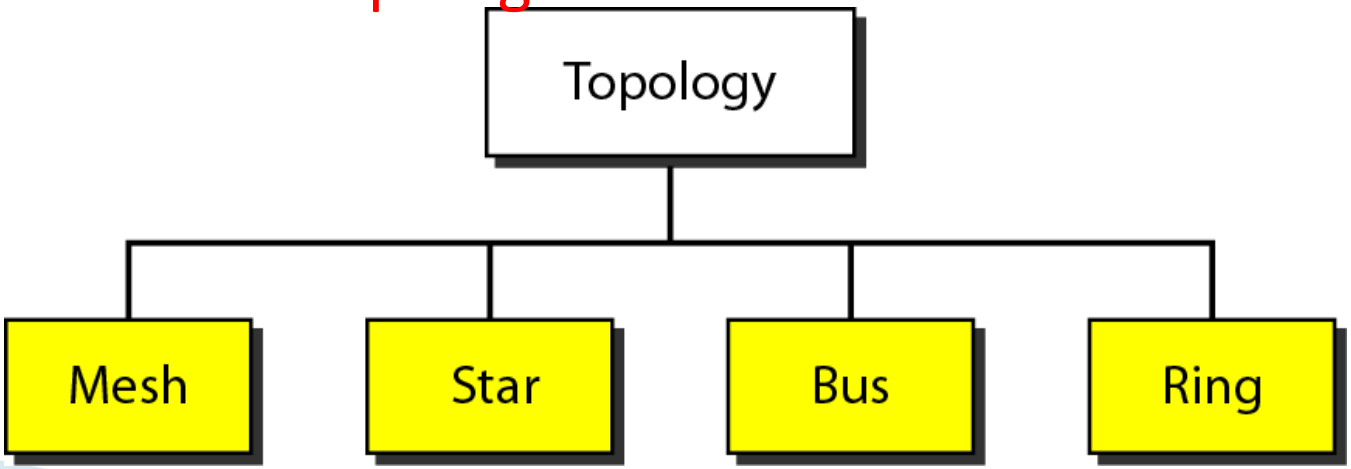
Direction of data flow

- There are three ways of data transmission:
- Simplex – communication is unidirectional. One can transmit and other can only receive. Ex: FM Radio
- Half-Duplex – Each station can transmit and receive, but not at the same time. Ex: Walke Talkie
- Full-Duplex - Each station can transmit and receive at the same time. Ex: Chatting

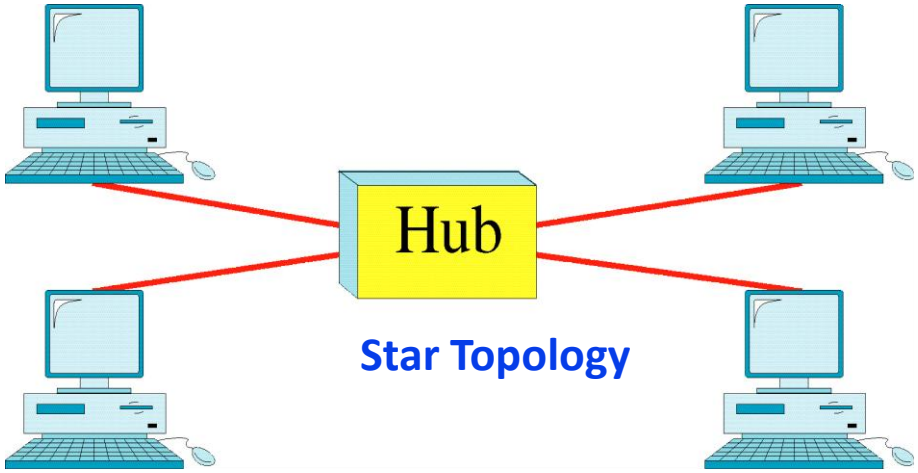
Network Topologies

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

1. Star
2. Mesh
3. Bus and
4. Ring



Star Topology



✓In a star topology, each device has a dedicated point-to-point link only to a central controller, usually called a hub/switch.

✓The devices are not directly linked to one another. 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.

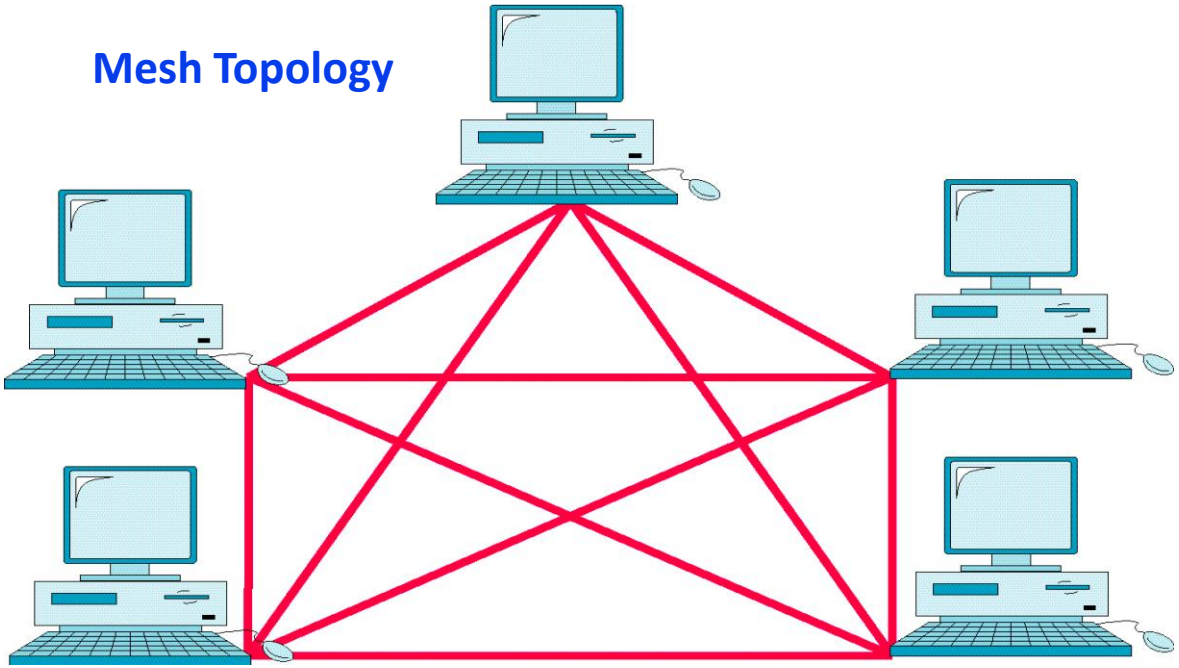
Advantages

- Star topology is less expensive
- Each device needs only one link and one I/O port to connect.
- Easy to install and reconfigure.
- Uses less no of cables
- Additions, moves, and deletions involve only one connection: between that device and the hub/switch.
- **Robustness:** If one link fails, only that link is affected. All other links remain active.

Disadvantages

- Dependency of the whole topology on one single point, the hub/switch. If the hub/switch goes down, the whole system is dead.
- Star requires each node must be linked to a central hub/switch.

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.
- N devices = n(n-1)/2 links are needed.

Advantages

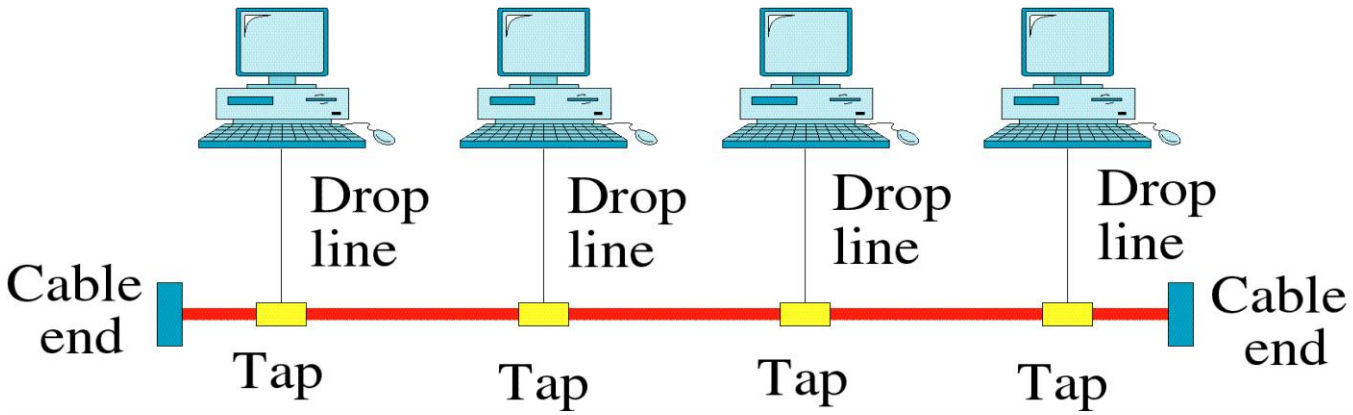
- The use of dedicated links guarantees that each connection can carry its own data load.
- If one link becomes unusable, it does not incapacitate the entire system.
- Privacy or Security: When every message travels along a dedicated line, only the intended recipient sees it.

Disadvantages

- More number of cabling and I/O ports required.
- Installation and reconnection are difficult.
- The hardware required to connect each link (I/O ports and cable) can be prohibitively expensive.

Bus Topology

Bus Topology



- 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.

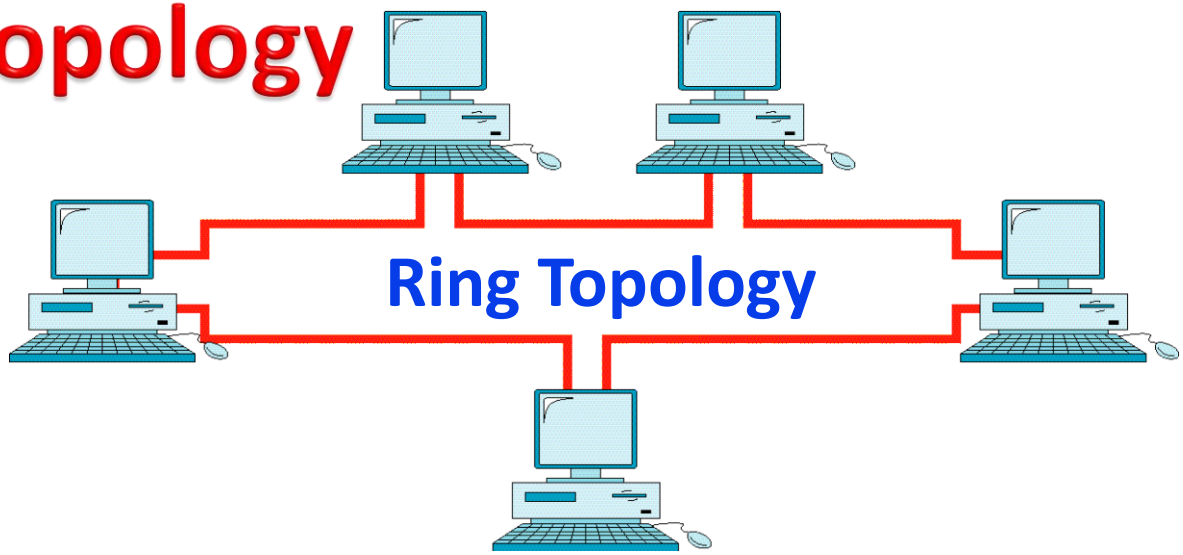
Advantages

- Easy to install.
- It uses less cabling than mesh or star topologies

Disadvantages

- Difficult to reconnect and fault isolation.
- Difficult to add new devices. Adding new devices may therefore require modification or replacement of the backbone.
- A fault or break in the bus cable stops all transmission.

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 respectively.

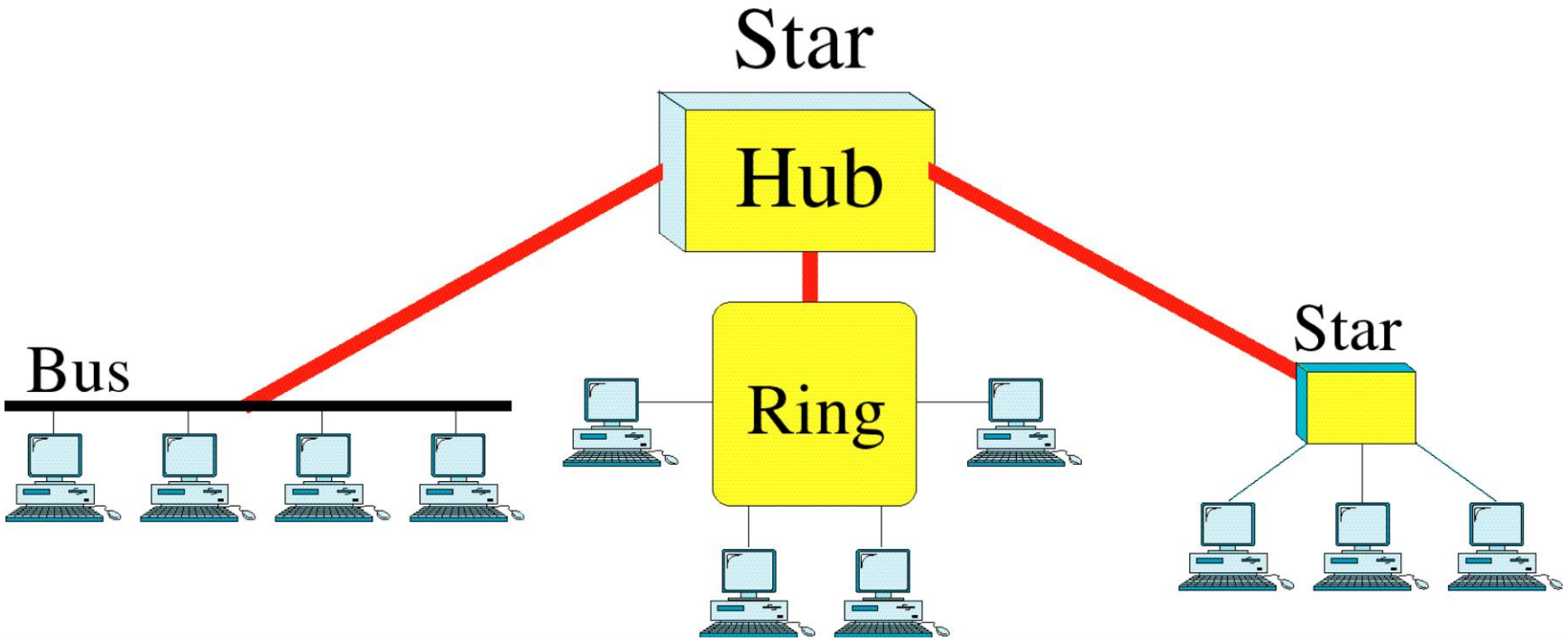
Advantages

- Easy to install and reconfigure.
- Each device is linked to only its immediate neighbors.
- To add or delete a device requires changing only two connections.

Disadvantages

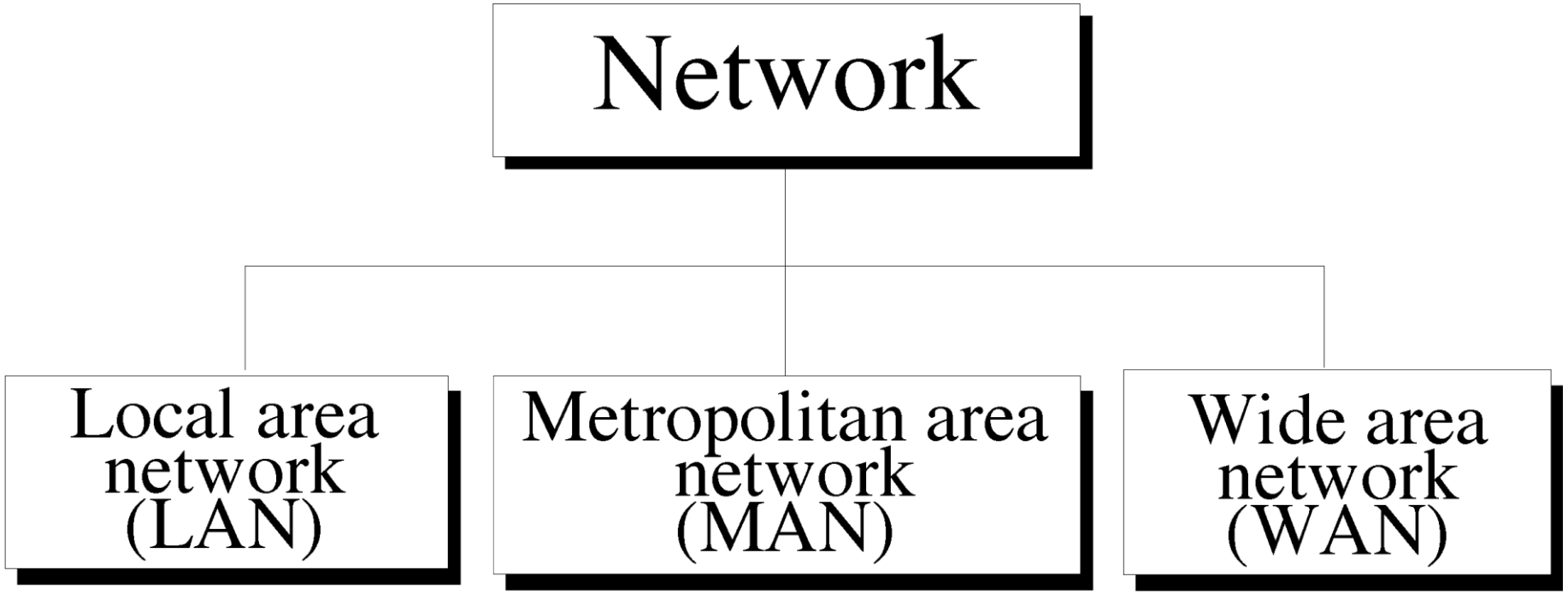
- Traffic is unidirectional.
- A break in the ring (such as a disabled station) can disable the entire network.
- This weakness can be solved by using a dual ring or a switch capable of closing off the break.

Hybrid Topology

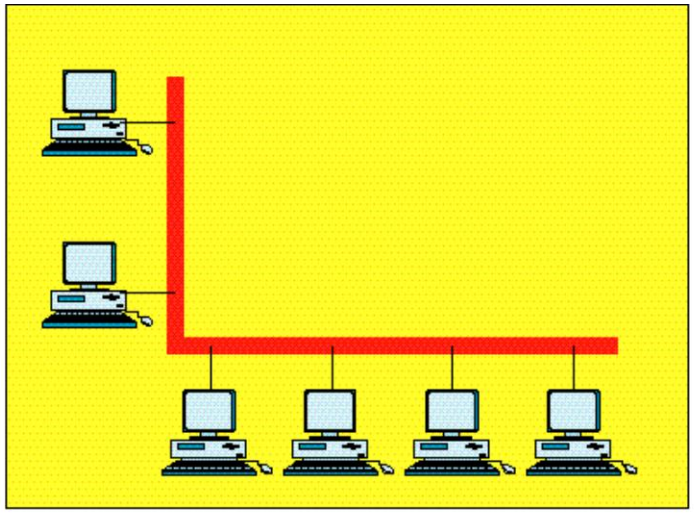


Categories of Networks

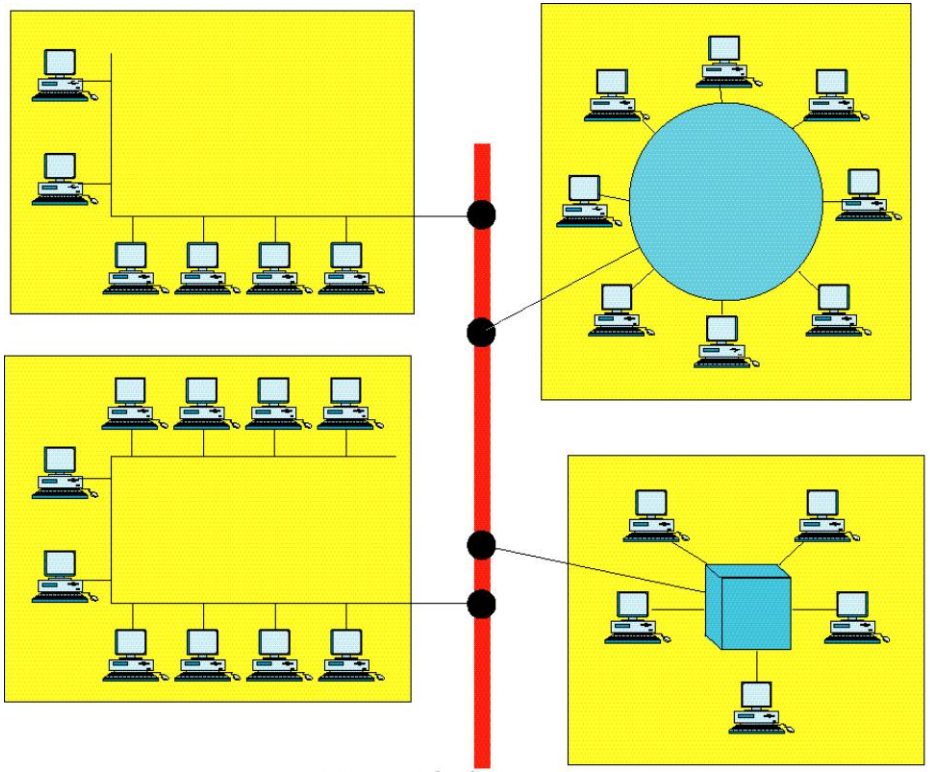
- There are three major categories of Networks
  - Local Area Networks(LAN)
  - Metropolitan Area Networks(MAN)
  - Wide Area Networks(WAN)



Local Area Networks(LAN)



Single building LAN

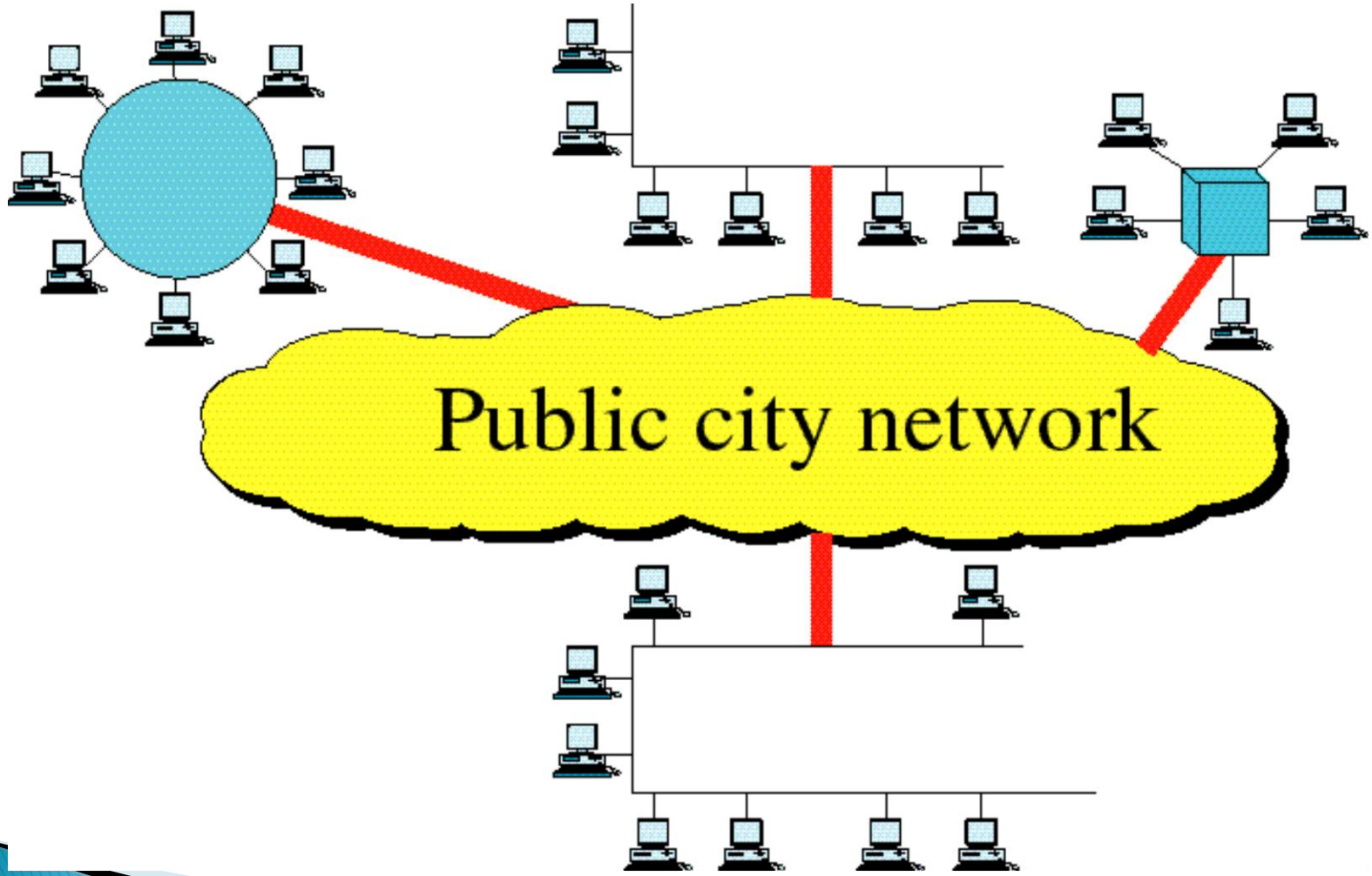


Multiple building LAN

Local Area Networks(LAN)

- A local area network (LAN) is usually privately owned and links the devices in a single office, building, or campus
- LANs are designed to allow resources to be shared between personal computers or workstations.
- The resources to be shared can include hardware (e.g., a printer), software (e.g., an application program), or data.

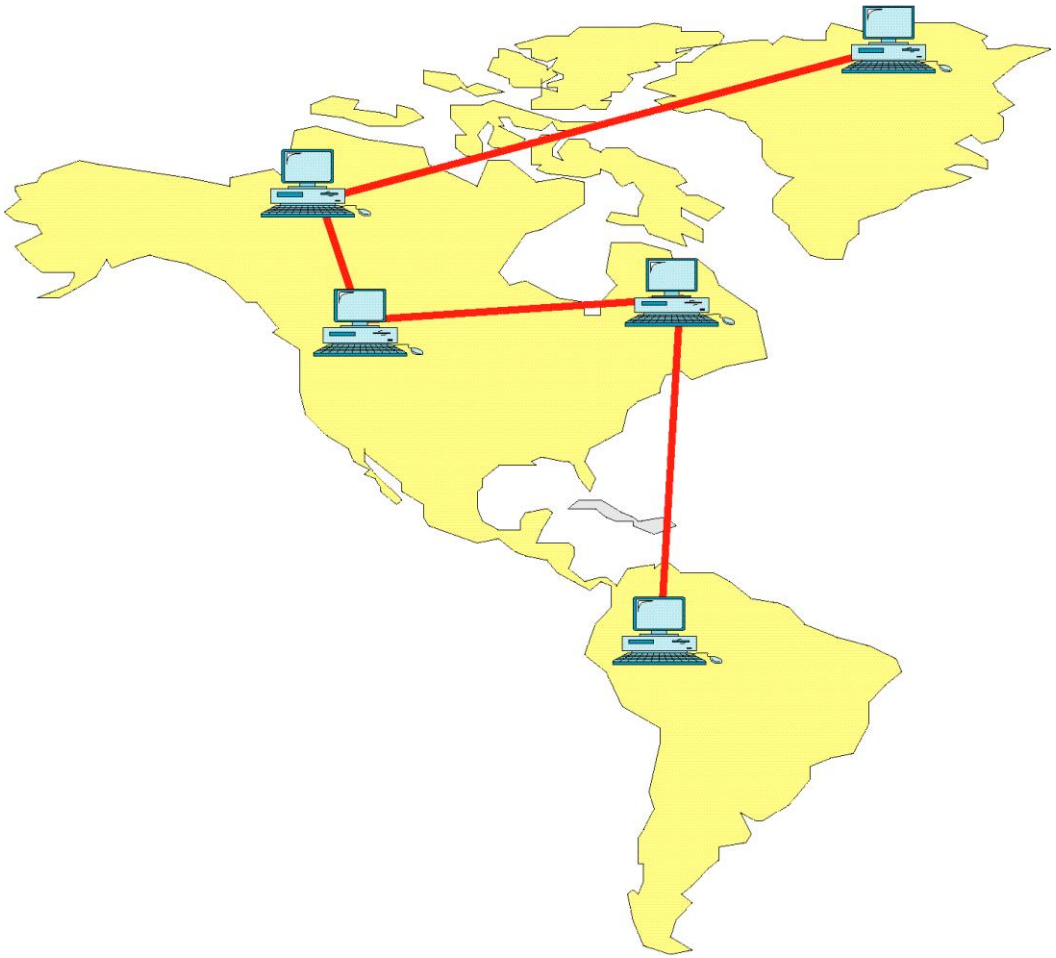
Metropolitan Area Networks(MAN)



Metropolitan Area Networks(MAN)

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

Wide Area Networks(WAN)



Wide Area Networks(WAN)

- A wide area network (WAN) provides long-distance transmission of data, image, audio, and video information over large geographic areas that may comprise a country, a continent, or even the whole world.

PROTOCOLS

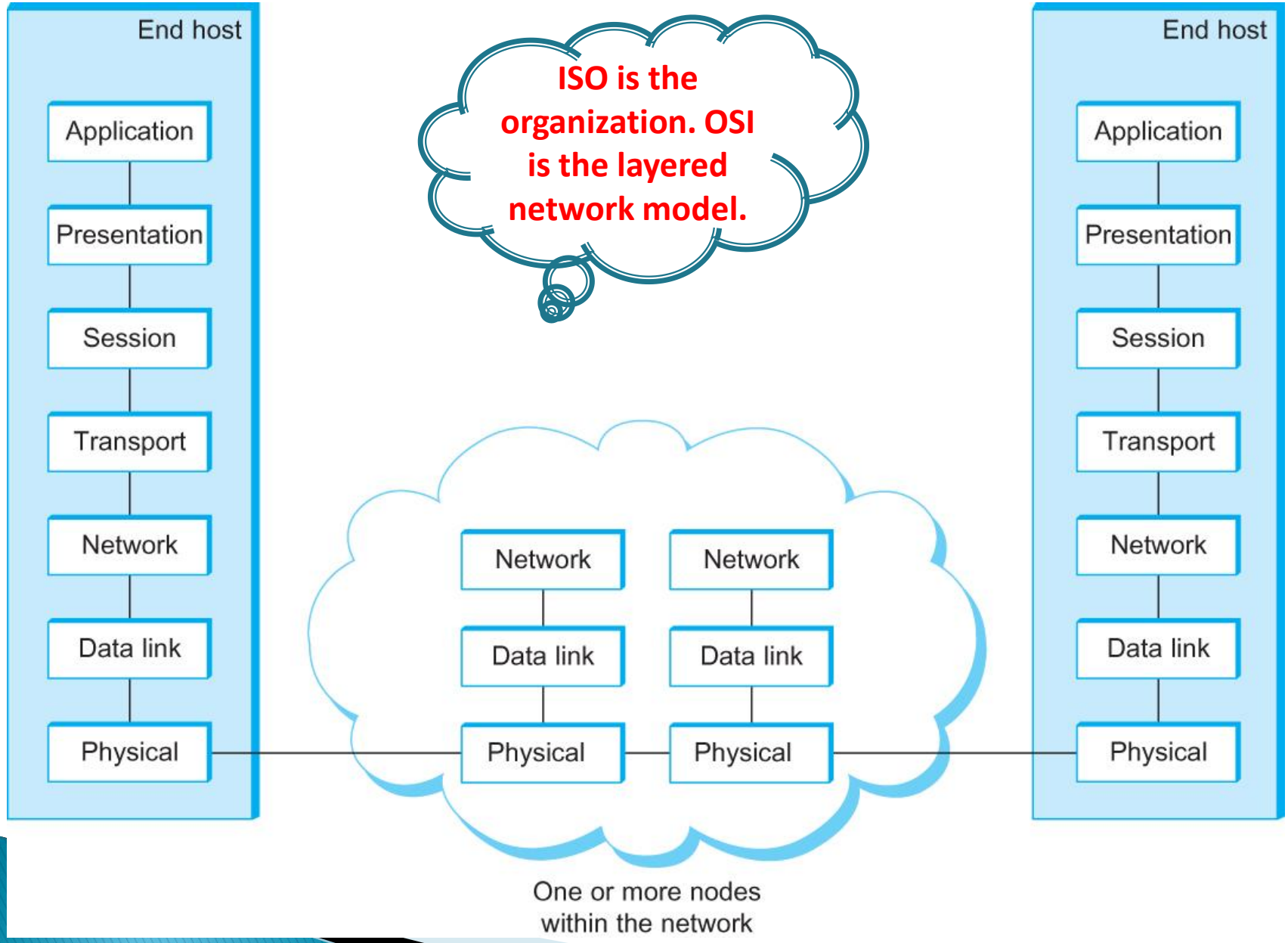
- Protocol defines
  - The **interfaces** between the layers in the same system and with the layers of peer system.
  - **Building blocks** of a network architecture.

- Each protocol object has two different interfaces
  - **Service interface**: Operations that local objects can perform on the protocol
  - **Peer-to-peer interface:** Messages exchanged with peer

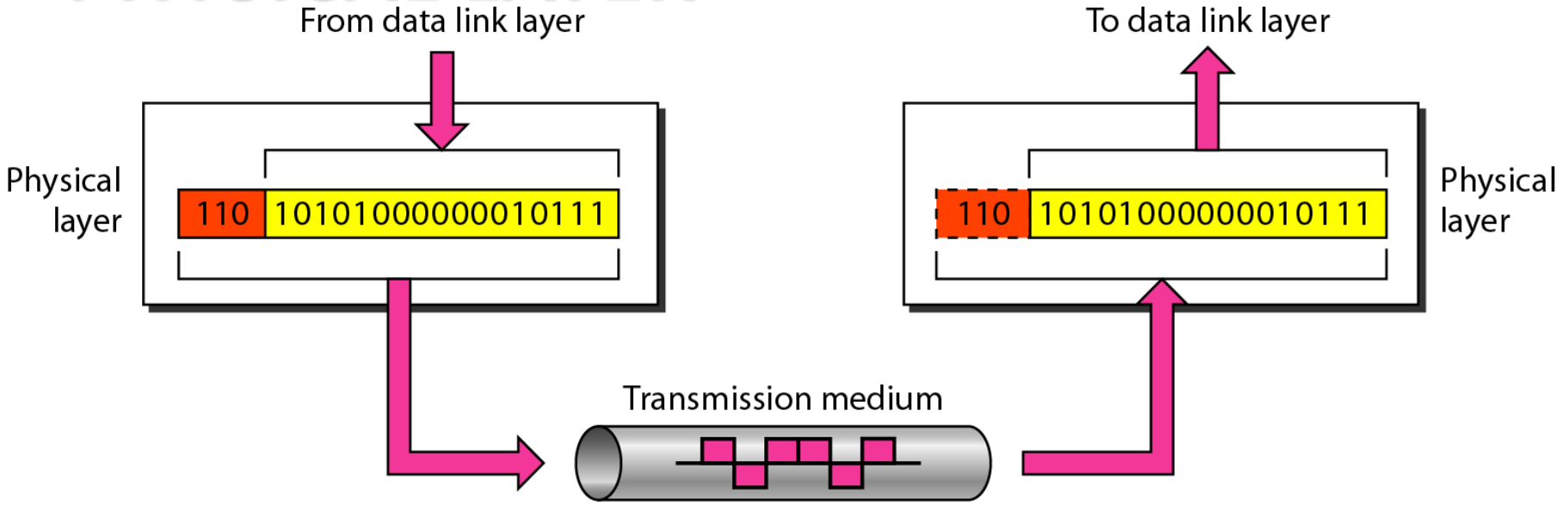
ISO/OSI ARCHITECTURE

- 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.
- The purpose of the OSI model is to show how to facilitate communication between different systems.
- The OSI model is not a protocol.
- It is a model for understanding and designing a network architecture that is flexible, robust, and interoperable.

ISO/OSI ARCHITECTURE



PHYSICAL LAYER

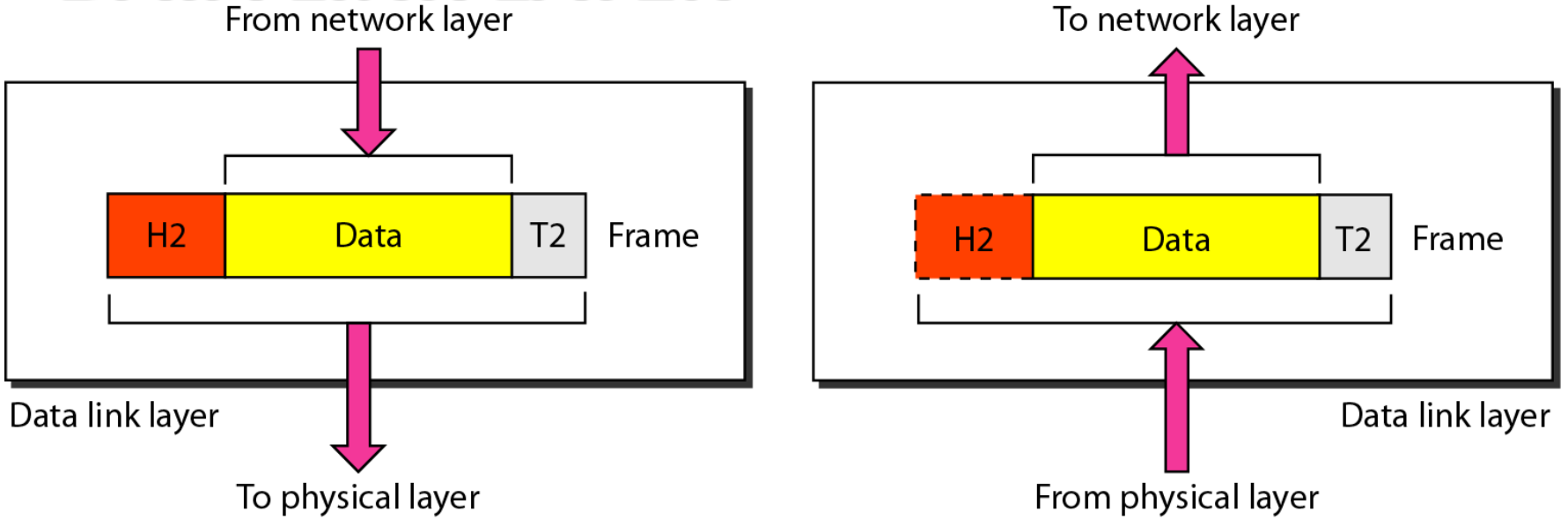


- The physical layer is responsible for movements of individual bits from one hop (node) to the next.
- The physical layer coordinates the functions required to carry a bit stream over a physical medium.
- It deals with the mechanical and electrical specifications of the interface and transmission medium.

- **Functions of Physical Layer:**
- **1.Physical characteristics of interfaces and medium:** The characteristics of the interface between the devices and the transmission medium. It also defines the type of transmission medium.
- **2.Representation of bits:** The physical layer data consists of a stream of bits (sequence of Os or 1s) with no interpretation.
- Bits must be encoded into signals--electrical or optical. The physical layer defines the type of encoding (how Os and I s are changed to signals).
- **3.Data rate:** The transmission rate, the number of bits sent each second is also defined by the physical layer.

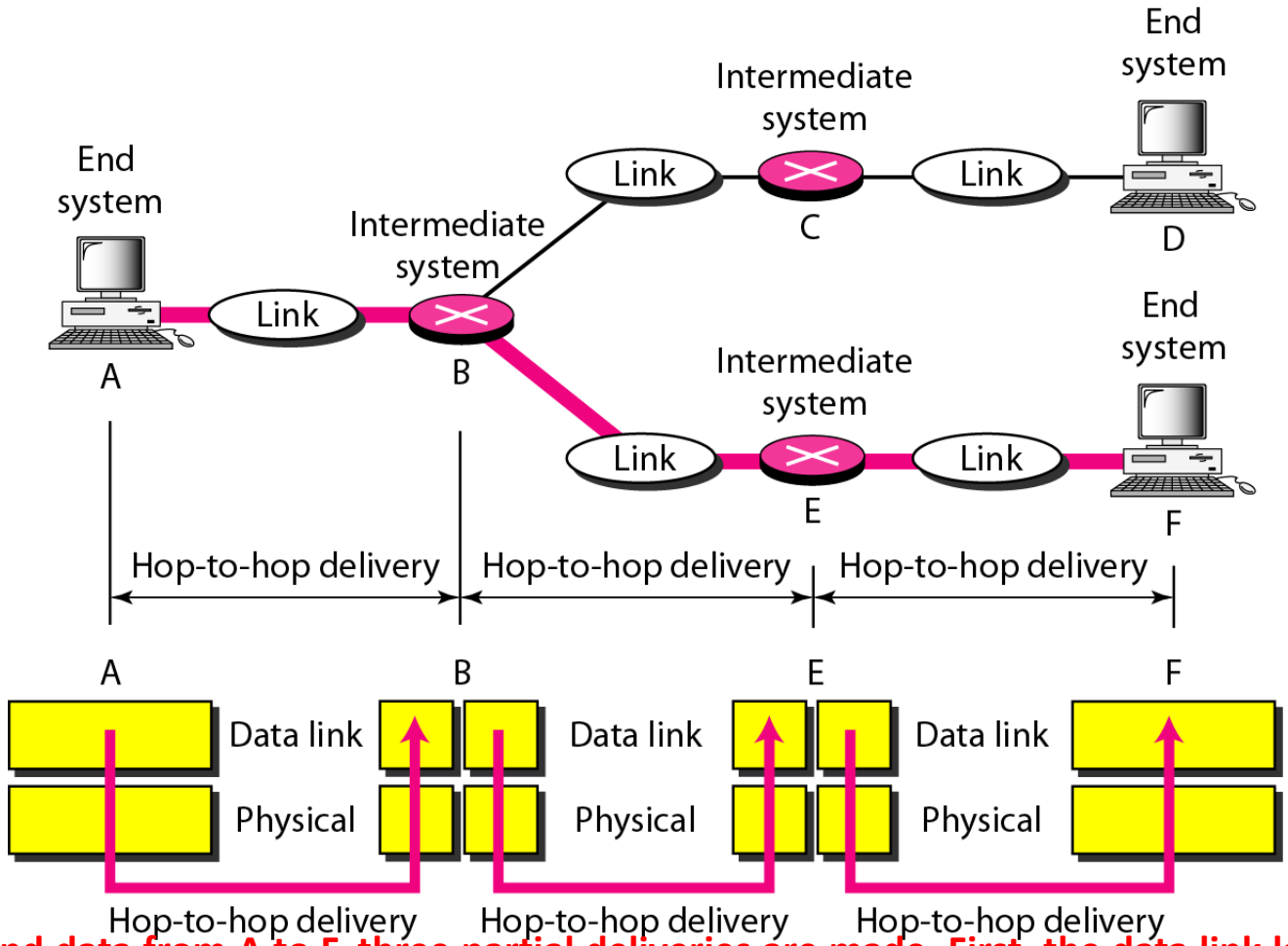
- **4.Synchronization of bits:** The sender and receiver not only must use the same bit rate but also the sender and the receiver clocks must be synchronized.
- **5.Line configuration:** The physical layer is concerned with the connection of devices to the media. In a point-to-point configuration, two devices are connected through a dedicated link. In a multipoint configuration, a link is shared among several devices.
- **6.Physical topology:** The physical topology defines how devices are connected to make a network. Devices can be connected by using various types of topologies **7.Transmission mode:** The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex.

DATA LINK LAYER



- The data link layer transforms the physical layer to a reliable link.
- It makes the physical layer appear error-free to the upper layer(network layer).
- It collects a stream of bits into a larger aggregate called **frame**
- The data link layer is responsible for moving frames from one hop (node) to the next.

Hop to Hop Delivery

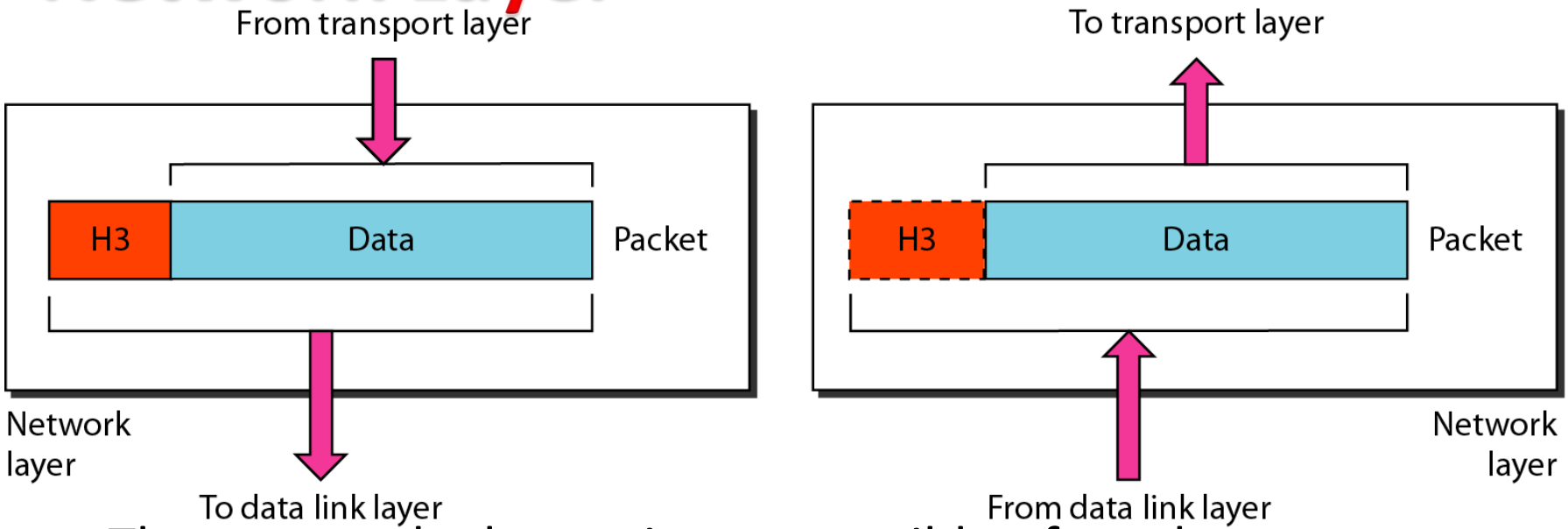


- To send data from A to F, three partial deliveries are made. First, the data link layer at A sends a frame to the data link layer at B (a router). Second, the data link layer at B sends a new frame to the data link layer at E. Finally, the data link layer at E sends a new frame to the data link layer at F.
- The frames that are exchanged between the three nodes have different values in the headers. The frame from A to B has B as the destination address and A as the source address. The frame from B to E has E as the destination address and B as the source address. The frame from E to F has F as the destination address and E as the source address.
- The values of the trailers can also be different if error checking includes the header of the frame.

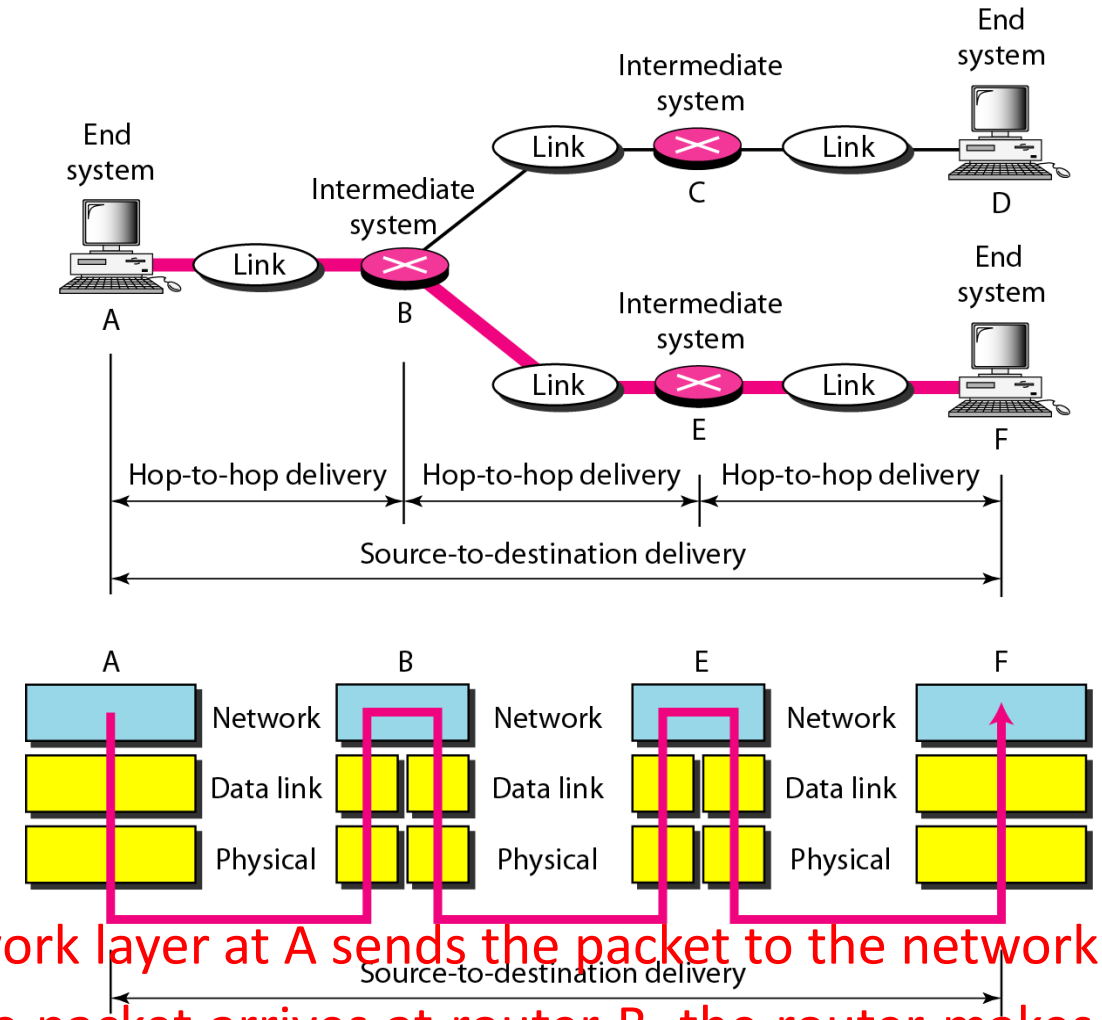
- **Functions of data link layer:**
- **1.Framing:** The data link layer divides the stream of bits received from the network layer into manageable data units called frames.
- **2.Physical addressing:** If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame.

- **3.Flow control:** If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.
- **4.Error control:** The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames.
- Error control is normally achieved through a trailer added to the end of the frame.
- **5.Access control:** When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

Network Layer



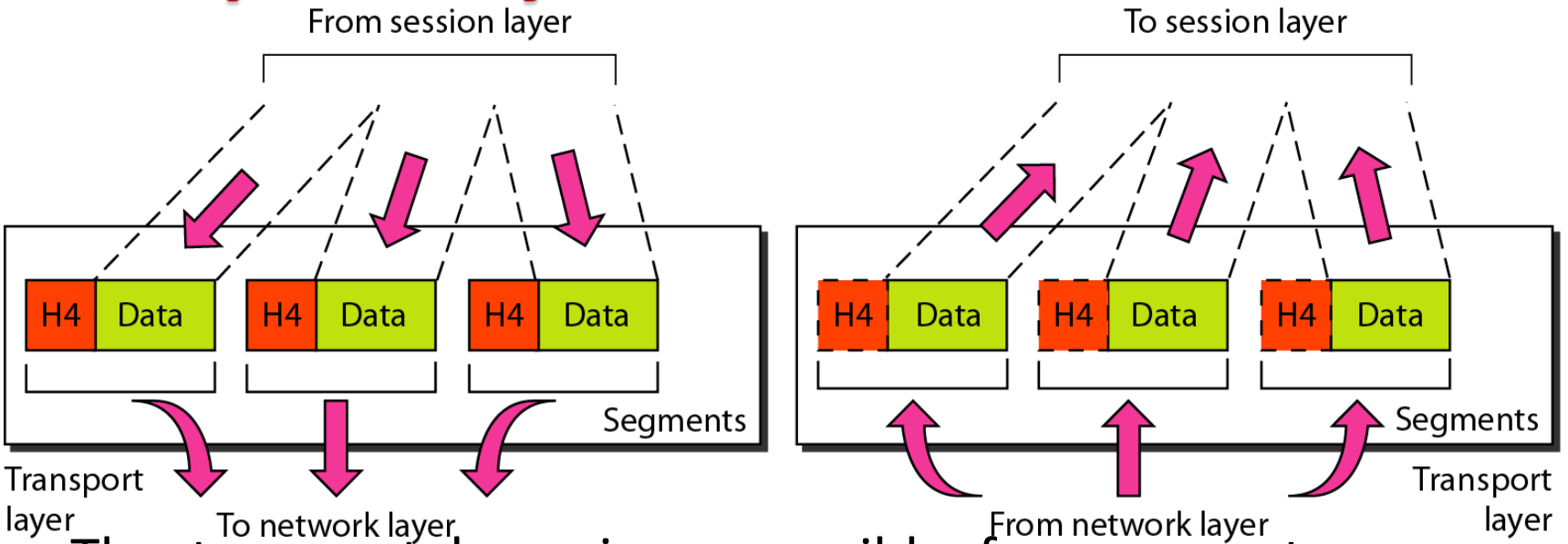
- The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links).
- The network layer ensures that each packet gets from its point of origin to its final destination.
- If two systems are connected to the same link, there is usually no need for a network layer. However, if the two systems are attached to different networks (links) with connecting devices between the networks (links), there is often a need for the network layer to accomplish source-to-destination delivery.



- The network layer at A sends the packet to the network layer at B.
- When the packet arrives at router B, the router makes a decision based on the final destination (F) of the packet. Router B uses its routing table to find that the next hop is router E.
- The network layer at B, therefore, sends the packet to the network layer at E. The network layer at E, in turn, sends the packet to the network layer at F.

- **Functions of Network Layer:**
- **1.Logical addressing:** The physical addressing implemented by the data link layer handles the addressing problem locally. If a packet passes the network boundary, we need another addressing system to help distinguish the source and destination systems.
- The network layer adds a header to the packet coming from the upper layer that, includes the logical addresses of the sender and receiver.
- **2.Routing:** When independent networks or links are connected to create *internetworks*(network of networks) or a large network, the connecting devices (called *routers* or *switches) route or switch the packets to their final destination.*

Transport Layer

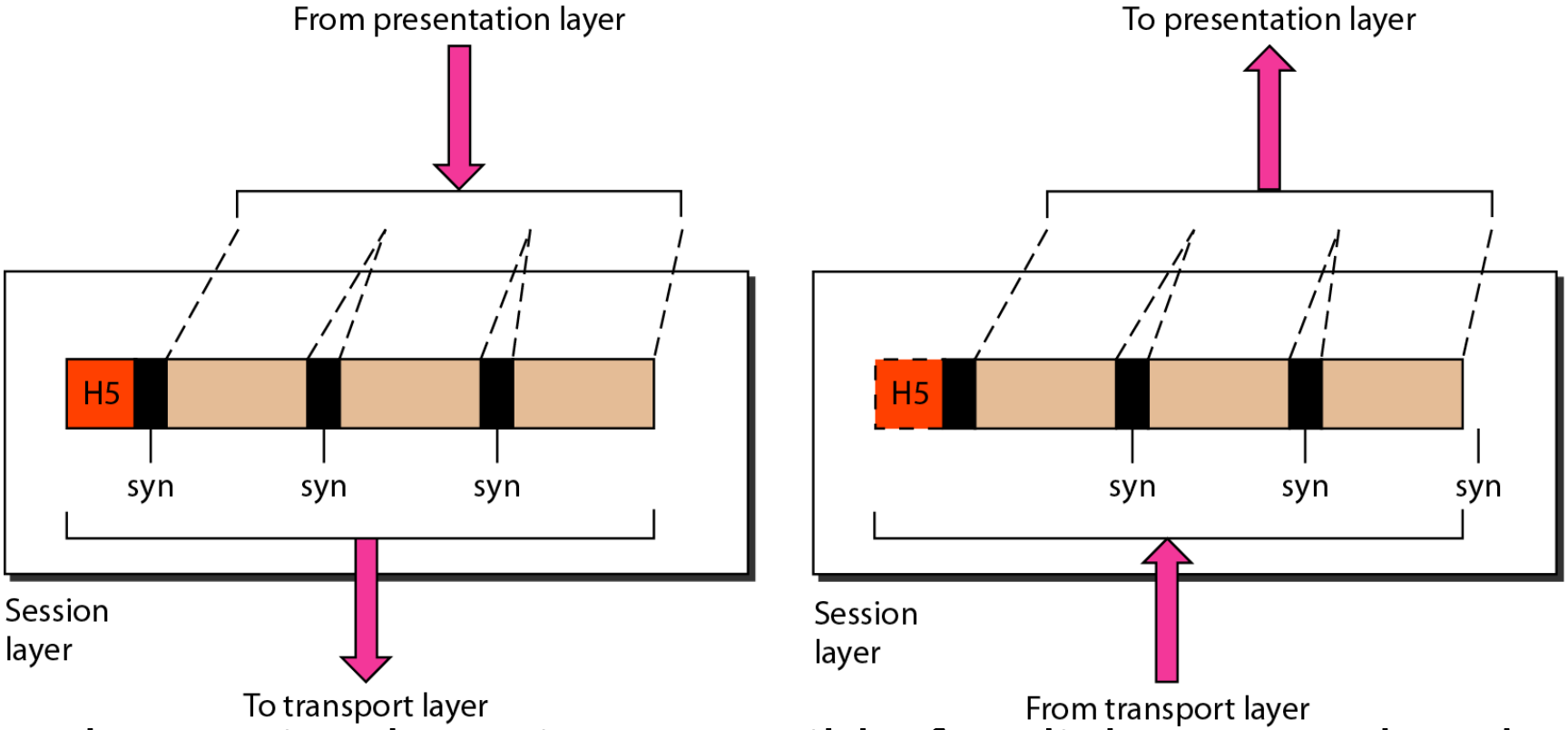


- The transport layer is responsible for process-to-process delivery of the entire message.
- A process is an application program running on a host.
- The network layer oversees source-to-destination delivery of individual packets, it does not recognize any relationship between those packets
- The transport layer, ensures that the whole message arrives intact and in order, overseeing both error control and flow control at the source-to-destination level.

FUNCTIONS OF TRANSPORT LAYER

- **1.Service-point addressing:** The transport layer header includes a type of address called a *service-point address (or port address).*
- The transport layer gets the entire message to the correct process on the destination computer by the help of this address.
- **2.Segmentation and reassembly:** A message is divided into segments, containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination.
- **3.Connection control:** The transport layer can be either connectionless or connection oriented. A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine.
- A connection oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data are transferred, the connection is terminated.
- **4.Flow control:** Flow control at this layer is performed at end to end
- **5.Error control:** Error control at this layer is performed process-to-process rather than across a single link.

SESSION LAYER

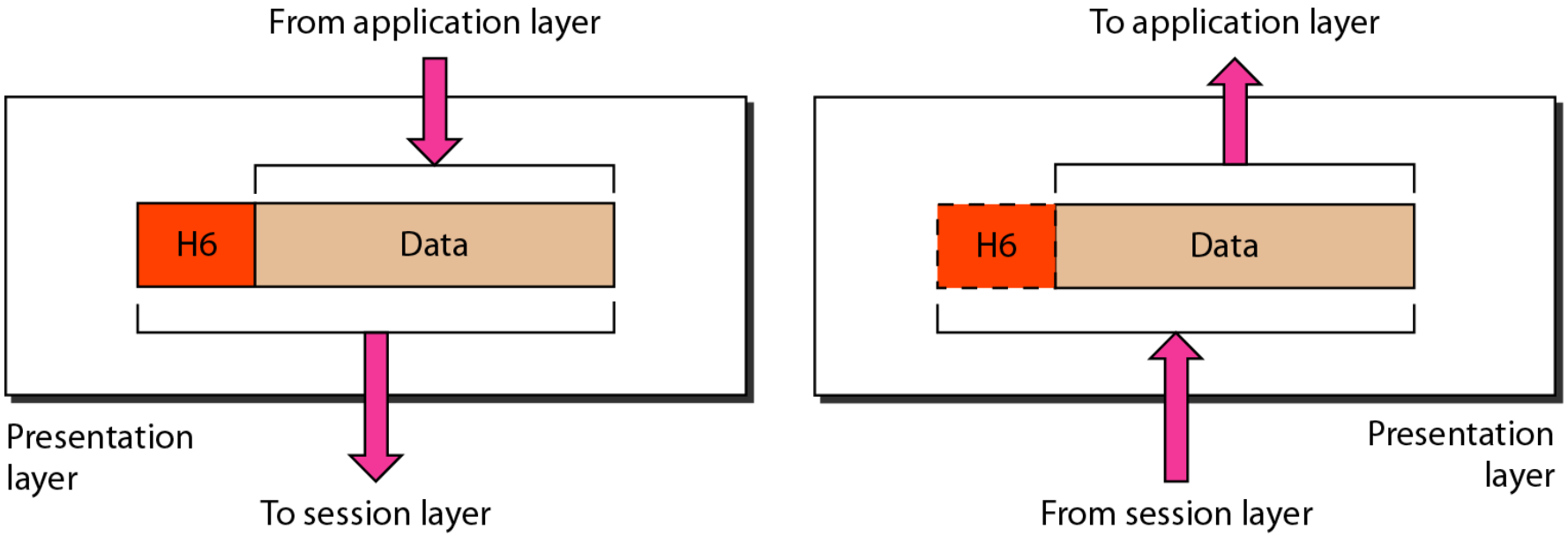


- The session layer is responsible for dialog control and synchronization.
- It establishes, maintains, and synchronizes the interaction among communicating systems.

FUNCTIONS OF SESSION LAYER

- **1.Dialog control:** The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either half duplex(one way at a time) or full-duplex (two ways at a time) mode.
- **2.Synchronization:** The session layer allows a process to add synchronization points, to a stream of data.
- For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently.
- In this case, if a crash happens during the transmission of page 523, the only pages that need to be resent after system recovery are pages 501 to 523.
- Pages previous to 501 need not be resent.

PRESENTATION LAYER

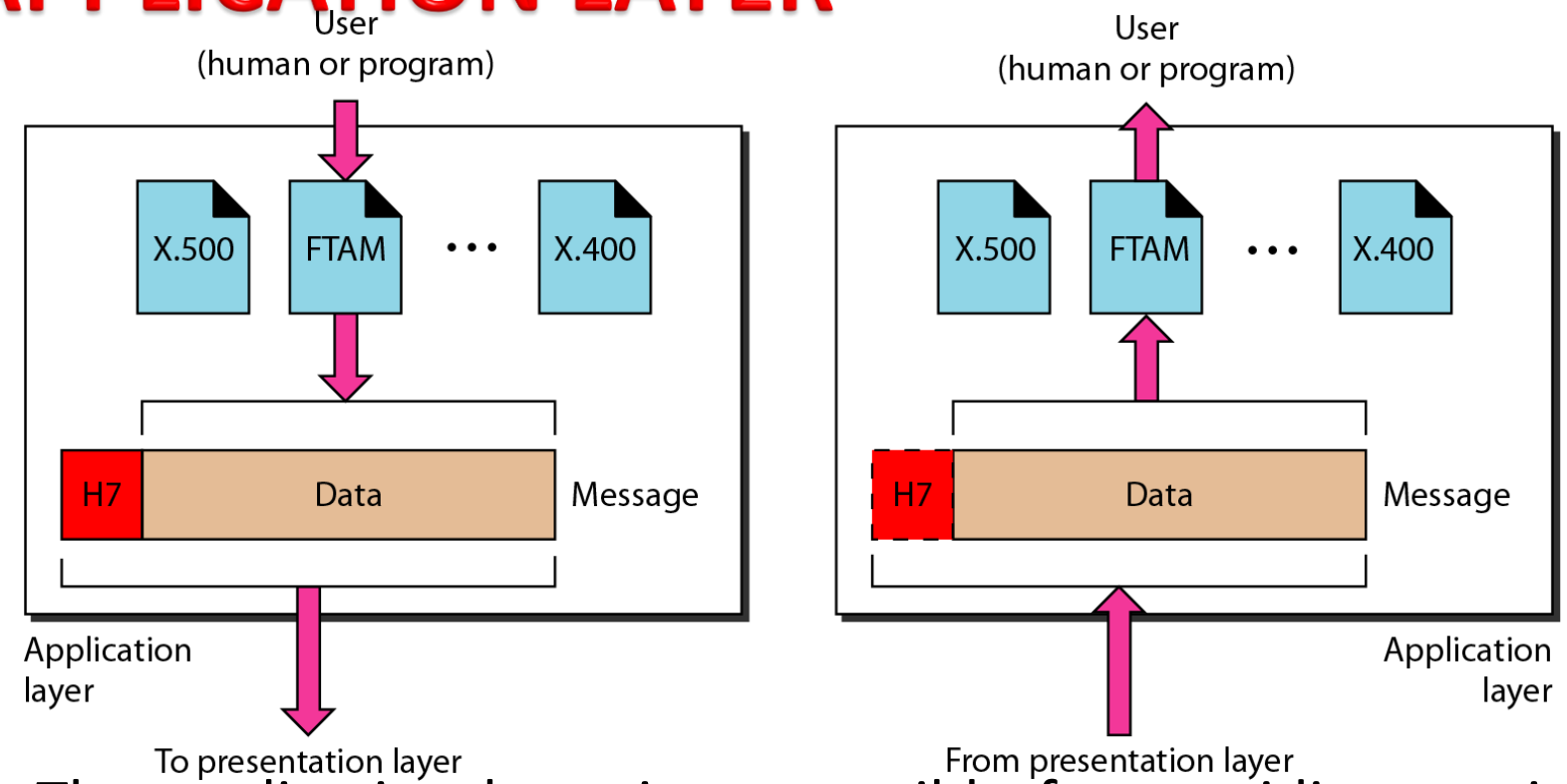


- The presentation layer is responsible for translation, compression, and encryption.

FUNCTIONS OF PRESENTATION LAYER

- **1.Translation:** The processes (running programs) in two systems are usually exchanging information in the form of character strings, numbers, and so on.
- The information must be changed to bit streams before being transmitted.
- **2.Encryption:** To carry sensitive information, a system must be able to ensure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network.
- Decryption reverses the original process to transform the message back to its original form.
- **3.Compression:** Data compression reduces the number of bits contained in the information. Data compression becomes particularly important in the transmission of multimedia such as text, audio, and video.

APPLICATION LAYER

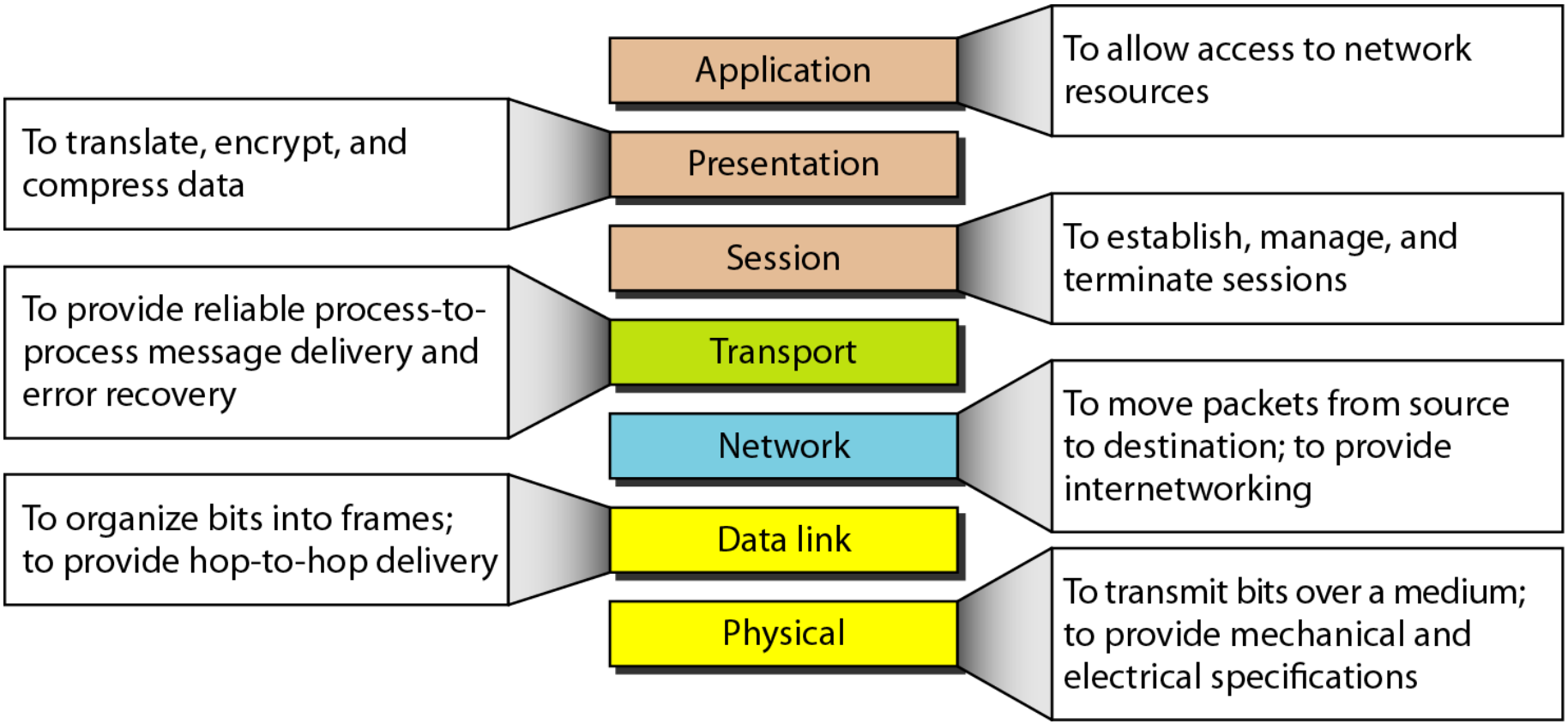


- The application layer is responsible for providing services to the user.
- The application layer enables the user, to access the network.
- It provides user interfaces and support for services such as electronic mail, remote file access and transfer, and other types of distributed information services.

FUNCTIONS OF APPLICATION LAYER

- **1.Network virtual terminal:** A network virtual terminal is a software version of a physical terminal, and it allows a user to log on to a remote host.
- **2.File transfer, access, and management:** This application allows a user to access files in a remote host (to make changes or read data), to retrieve files from a remote computer for use in the local computer, and to manage or control files in a remote computer locally.
- **3.Mail services:** This application provides the basis for e-mail forwarding and storage.
- **4.Directory services:** This application provides distributed database sources and access for global information about various objects and services.

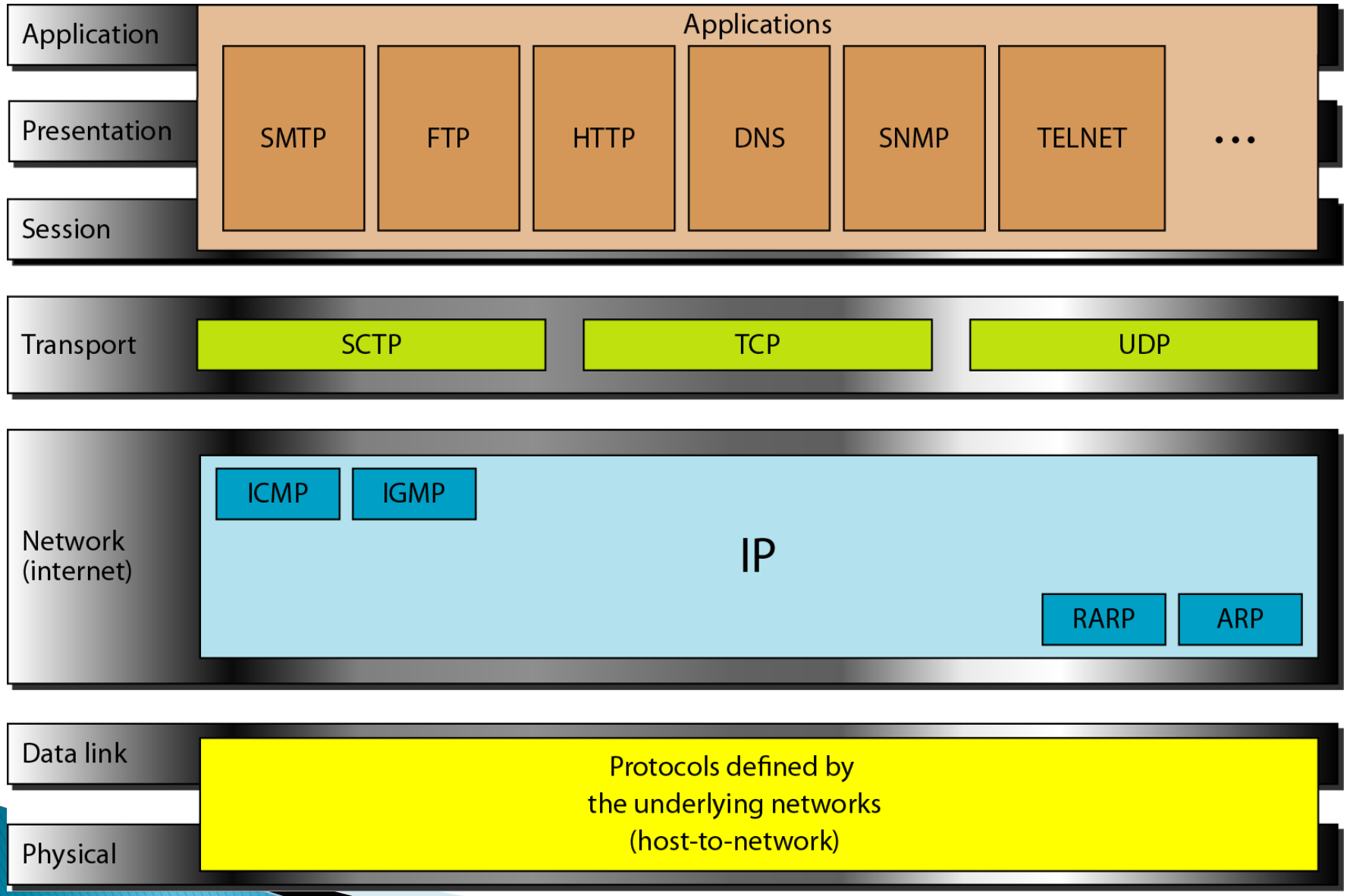
SUMMARY OF LAYERS IN OSI MODEL



TCP/IP PROTOCOL

- The layers in the TCP/IP protocol suite do not exactly match those in the OSI model.
- The TCP/IP protocol suite was defined as having four layers: host-to-network, internet, transport, and application.
- when TCP/IP is compared to OSI, the host-to-network layer is equivalent to the combination of the physical and data link layers.
- The internet(IP) layer is equivalent to the network layer
- The application layer is doing the job of the session, presentation, and application layers
- The transport layer in TCP/IP taking care of part of the duties of the session layer.

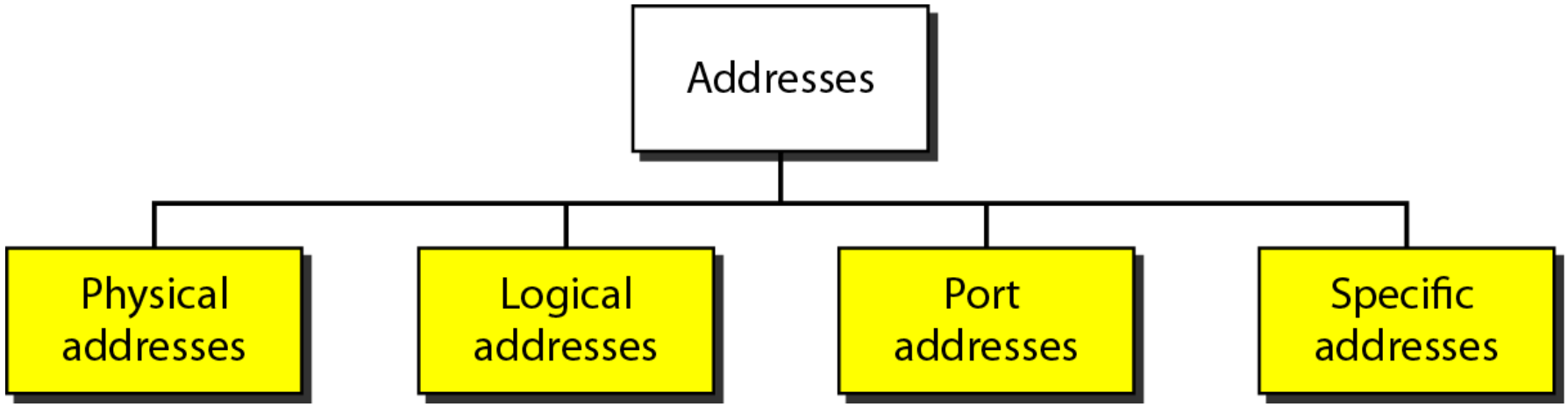
TCP/IP and OSI model



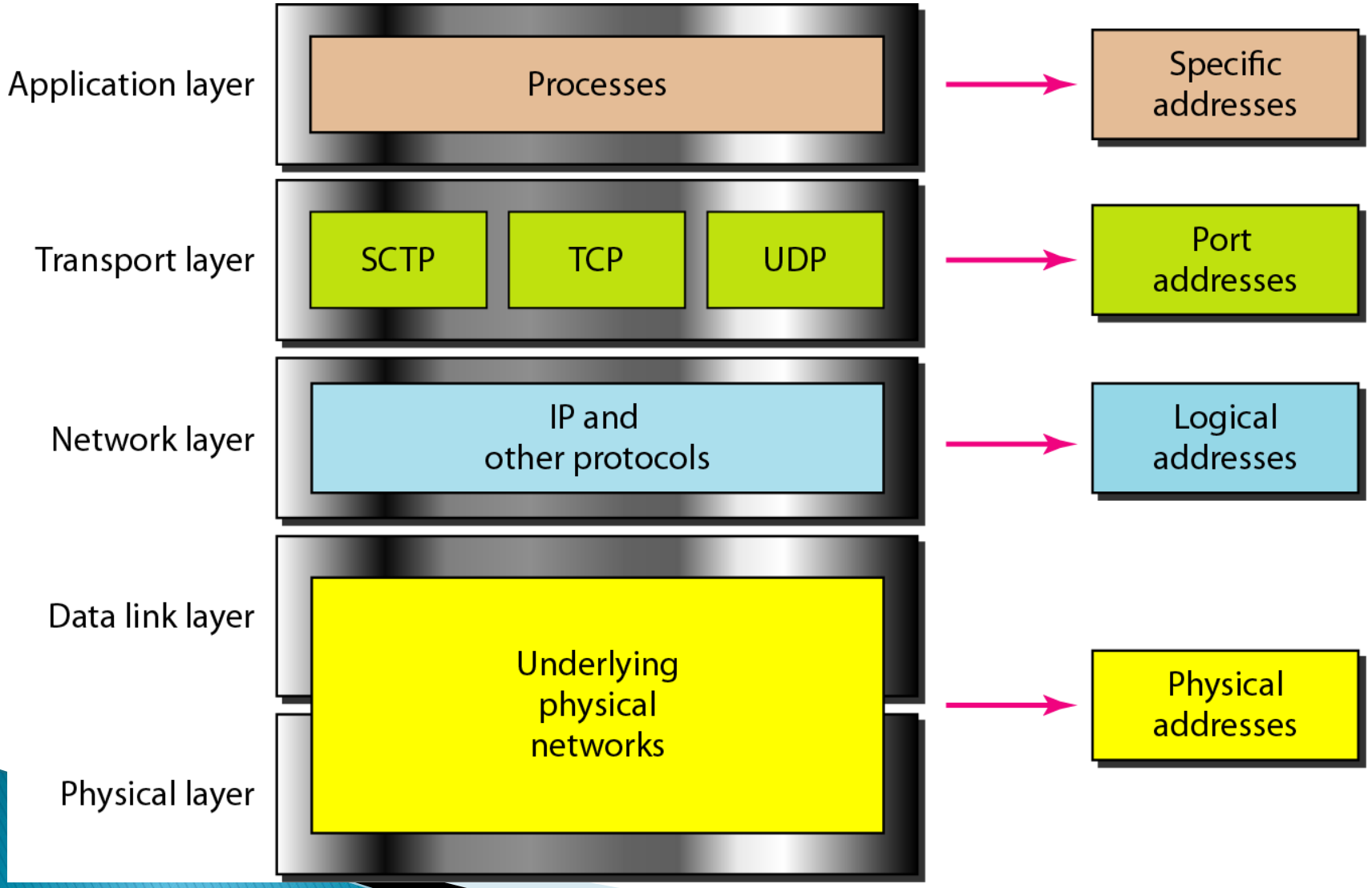
- **HOST -TO-NETWORK LAYER:** In this layer TCP/IP does not define any specific protocol.
- *It* supports all the standard and proprietary protocols. A network in a *TCP/IP internetwork* can be a local-area network or a wide-area network.
- **IP LAYER:** In this layer TCP/IP *supports* the Internetworking Protocol and uses four supporting protocols: ARP, RARP, ICMP, and IGMP.
- **TRANSPORT LAYER:** In this layer TCP/IP defines three protocols: Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Stream Control Transmission Protocol (SCTP).
- **APPLICATION LAYER:** In this layer TCP/IP supports protocols such as SMTP, FTP, HTTP, DNS, TELNET…

ADDRESSING

- Four levels of addresses are used in an internet employing the *TCP/IP protocols:* physical (link) addresses, logical (IP) addresses, port addresses, and specific addresses.



- **Relationship of layers and addresses in TCP/IP:** Each address is related to a specific layer in the TCP/IP architecture



- **1.Physical Addresses(MAC Address):** It is also known as the link address of a node as defined by its LAN or WAN.
- It is included in the frame used by the data link layer. It is the lowest-level address.
- **2.Logical Addresses(IP Address):** It is necessary for universal communications that are independent of underlying physical networks.
- Physical addresses are not adequate in an internetwork environment where different networks can have different address formats.
- A universal addressing system is needed in which each host can be identified uniquely, regardless of the underlying physical network.
- The logical addresses are designed for this purpose.
- **The physical addresses will change from hop to hop, but the logical addresses usually remain the same**.
- No two publicly addressed and visible hosts on the Internet can have the same IP address.

- **3.Port Addresses:** computers are devices that can run multiple processes at the same time. The end objective of Internet communication is a process communicating with another process.
- For example, computer A can communicate with computer C by using TELNET. At the same time, computer A communicates with computer B by using the File Transfer Protocol (FTP).
- For these processes to receive data simultaneously, we need a method to label the different processes.
- In the TCP/IP architecture, the label assigned to a process is called a port address.
- A port address in TCP/IP is 16 bits in length.

- **4.Specific Addresses:** Some applications have user-friendly addresses that are designed for that specific address.
- Examples include the e-mail address (for example, noornilo@gmail.com) and the Universal Resource Locator (URL) (for example, www.nmtjonline.com).
- The first defines the recipient of an e-mail and the second is used to find a document on the World Wide Web.

NETWORK SOFTWARE(APIs)

- **Application Programming Interface(Socket):** Most of the network protocols are implemented (those in the high protocol stack) in software and nearly all computer systems provides interfaces to implement their network protocols as a part of the operating system.
- This interface is called the network Application Programming Interface (Socket).
- Socket Interface was originally provided by Unix.
- Now supported virtually by all operating systems
- Each protocol provides a certain set of *services*, and the API provides a syntax by which those services can be invoked in this particular OS.

- **What is a socket?**
- The point where a local application process attaches to the network.
- An interface between an application and the network.
- An application creates the socket
- **This interface defines operations for:**
- Creating a socket
- Attaching a socket to the network
- Sending and receiving messages through the socket
- Closing the socket

- **Socket Family**
  - PF_INET denotes the Internet family
  - PF_PACKET denotes direct access to the network interface (i.e., it bypasses the TCP/IP protocol stack)

- **Socket Type**
  - SOCK_STREAM is used to denote a byte stream
  - SOCK_DGRAM is an alternative that denotes a service, such as that provided by UDP

- **Creating a Socket:**
- int sockfd = socket(address_family, type, protocol);
- int sockfd = socket (PF_INET, SOCK_STREAM, 0);
- int sockfd = socket (PF_INET, SOCK_DGRAM, 0);
- The combination of PF_INET and SOCK_STREAM implies TCP

Client Server Model with TCP:

- Server
  - Passive open
  - Prepares to accept connection, does not actually establish a connection
- Server invokes
- int bind ();
- int listen ();
- int accept ();

- **Bind:**
  - Binds the newly created socket to the specified address i.e. the network address of the local participant (the server)
  - Address is a data structure which combines IP and port
- **Listen:**
  - Defines how many connections can be pending on the specified socket
- **Accept:**
  - Carries out the passive open.
- **Client:**
  - Application performs active open
  - It says who it wants to communicate with
- **Client invokes:**
- int connect ();

- Once a connection is established, the application process invokes two operation:
- int send ();
- int recv ();

Example Application : Client

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#define SERVER_PORT 5432
#define MAX_LINE 256
int main(int argc, char * argv[])
{
        FILE *fp;
        struct hostent *hp;
        struct sockaddr_in sin;
        char *host;
        char buf[MAX_LINE];
        int s;
        int len;
        if (argc==2) {
                host = argv[1];
        }
        else {
                fprintf(stderr, "usage:
simplex-talk host\n");
        exit(1);
        }
```

```
/* translate host name into peer's IP address */
hp = gethostbyname(host);
if (!hp) {fprintf(stderr, "simplex-talk: unknown host:
%s\n", host);exit(1);}/* build address data structure
*/bzero((char *)&sin, sizeof(sin));
   sin.sin_family = AF_INET;
   bcopy(hp->h_addr, (char *)&sin.sin_addr, hp-
>h_length);
        sin.sin_port = htons(SERVER_PORT);
        /* active open */
if ((s = socket(PF_INET, SOCK_STREAM, 0)) < 0) {
                perror("simplex-talk: socket");
                exit(1);}
        if (connect(s, (struct sockaddr *)&sin,
sizeof(sin)) < 0) {
                perror("simplex-talk: connect");
                close(s);
                exit(1);
        }
        /* main loop: get and send lines of text */
        while (fgets(buf, sizeof(buf), stdin)) {
                buf[MAX_LINE-1] = '\0';
                len = strlen(buf) + 1;
                send(s, buf, len, 0); }}
```

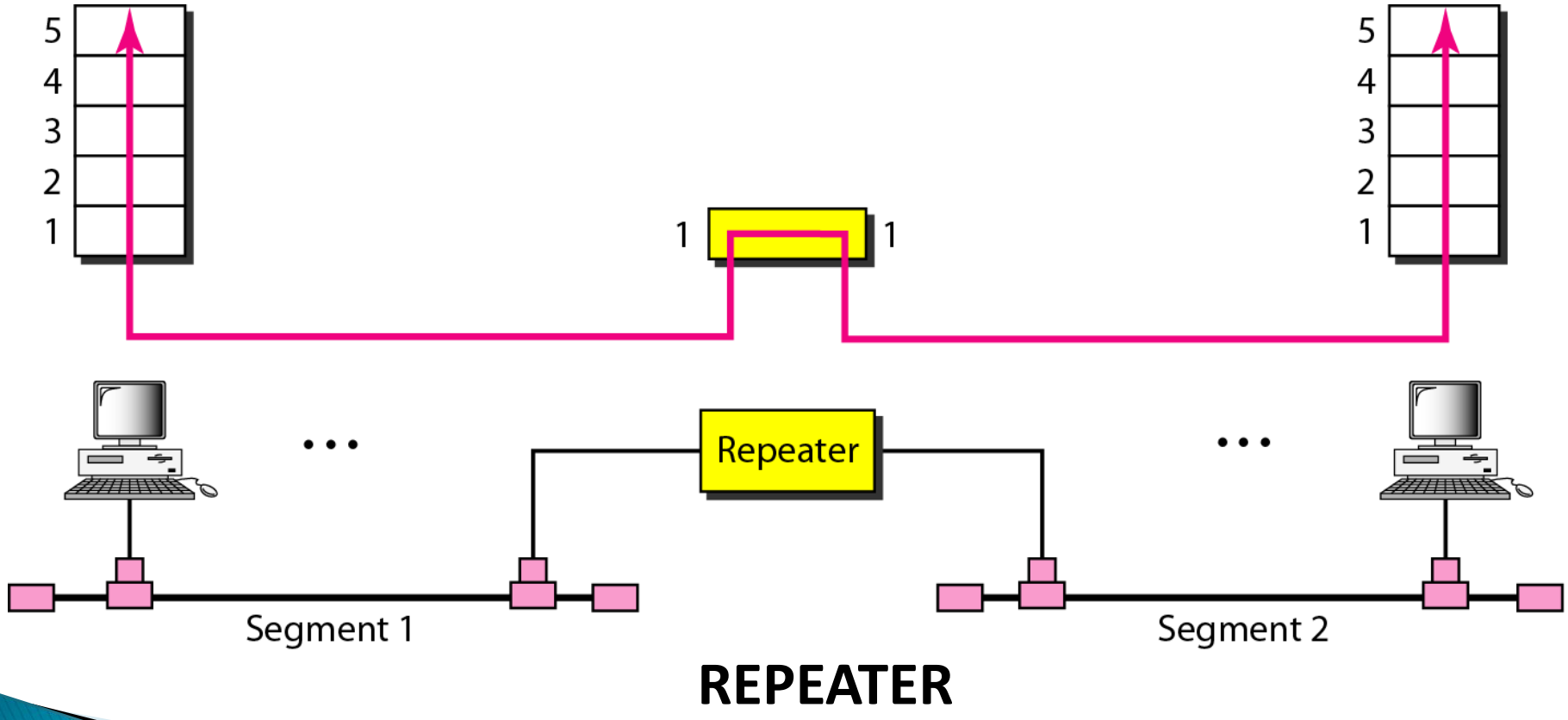
Example Application : Server

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#define SERVER_PORT 5432
#define MAX_PENDING 5
#define MAX_LINE 256
int main()
{
        struct sockaddr_in sin;
        char buf[MAX_LINE];
        int len;
        int s, new_s;
        /* build address data structure */
        bzero((char *)&sin, sizeof(sin));
        sin.sin_family = AF_INET;
        sin.sin_addr.s_addr =INADDR_ANY;
        sin.sin_port =
         htons(SERVER_PORT);
        /* setup passive open */
if ((s = socket(PF_INET, SOCK_STREAM, 0))
 < 0) {perror("simplex-talk: socket");
                        exit(1);    }
if ((bind(s, (struct sockaddr *)&sin,
sizeof(sin))) < 0) {
                        perror("simplex-talk:
bind");
                        exit(1);
        }
        listen(s, MAX_PENDING);
        /* wait for connection, then receive
and print text */
        while(1) {
                        if ((new_s = accept(s,
(struct sockaddr *)&sin, &len)) < 0) {
                        perror("simplex-talk:
accept");
                        exit(1);
        }
        while (len = recv(new_s, buf,
sizeof(buf), 0))
                        fputs(buf, stdout);
                        close(new_s);
        }
}
```

NETWORK HARDWARE

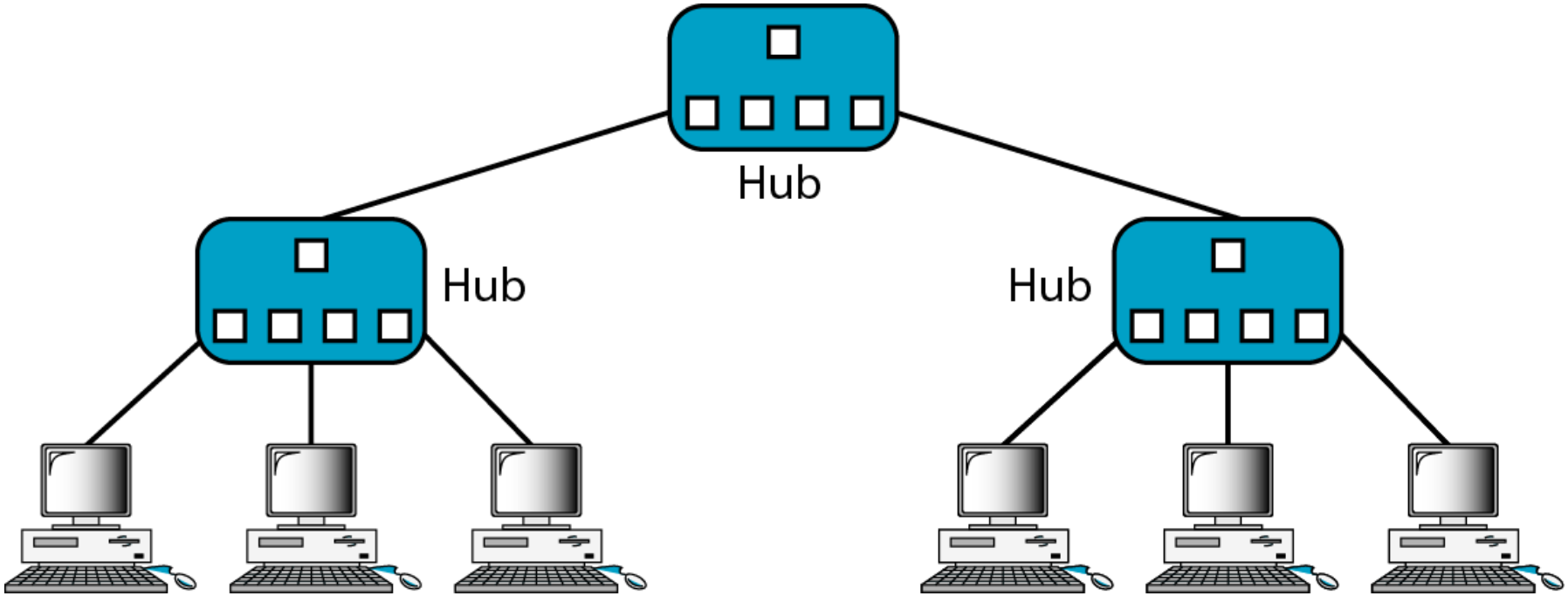
- **Repeaters**
- A repeater is a device that operates only in the physical layer.
- Signals that carry information within a network can travel a fixed distance before attenuation endangers the integrity of the data.
- A repeater receives a signal and, before it becomes too weak or corrupted, regenerates the original bit pattern and sends the refreshed signal.
- A repeater can extend the physical length of a LAN

- A repeater does not actually connect two LANs; it connects two segments of the same LAN.
- The segments connected are still part of one single LAN.



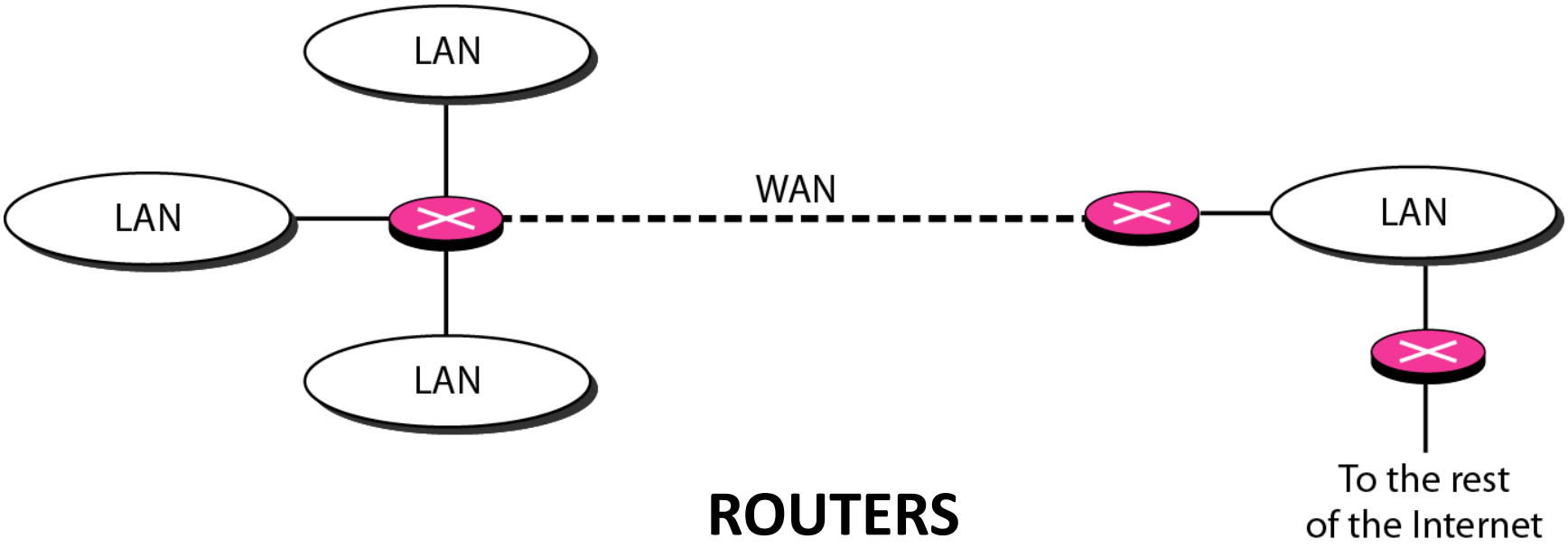
REPEATER

- **Hub:**
- A hub is a multi-port repeater, used in star-wired LANs (Ethernet).
- Because of the amount of traffic and collisions, hubs can only be used in small network configurations.
- **Passive Hub**
- A passive hub is just a connector. It connects the wires coming from different branches.
- **Active Hub**
- It is actually a multiport repeater. It is normally used to create connections between stations in a physical star topology.



HUB

- **Routers:**
- A router is a layer three device that routes packets based on their logical addresses (host-to-host addressing).
- A router normally connects LANs and WANs in the Internet and has a routing table that is used for making decisions about the route.
- The routing tables are normally dynamic and are updated using routing protocols.



ROUTERS

- **Switch**
- A switch, in the context of networking is a high-speed device that receives incoming data packets and redirects them to their destination on a local area network (LAN).
- A LAN switch operates at the data link layer (Layer 2) or the network layer of the OSI Model.
- A hub simply connects all the nodes on the network resulting in collisions.
- But a switch creates an electronic tunnel between source and destination ports that no other traffic can enter. This results in communication without collisions.
- Switches are similar to routers, but a router has the additional ability to forward packets between different networks, whereas a switch is limited to node-to-node communication on the same network.

10.1.3. Bridges:

10.1.3.1. Introduction:

- ✓ Bridges operate in both the physical and the data link layers of the OSI model.
- ✓ The main idea of using a bridge is to divide a big network into smaller sub-networks, called as ***segments.***

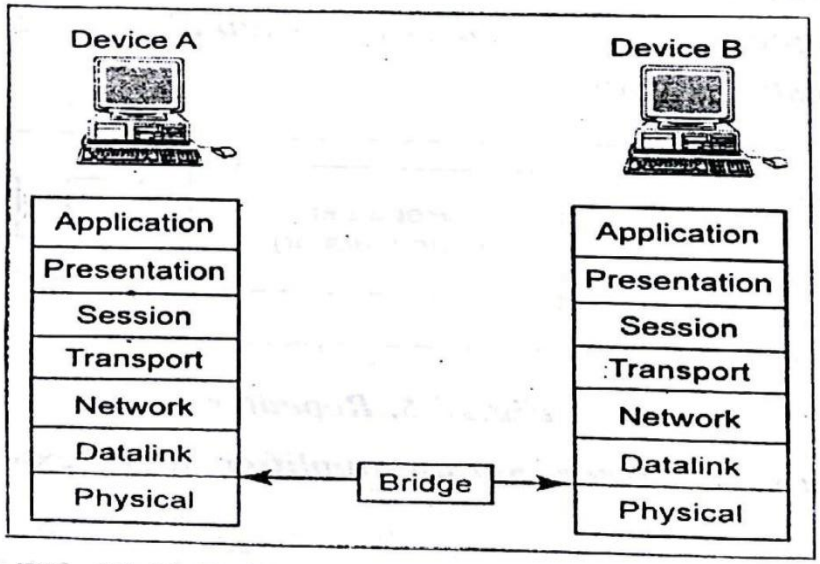


Fig.10.7. Bridges at the last two OSI layer

- ✓ A bridge operates at the data link layer, giving access to the physical (source and destination) addresses of all stations connected it.

Scanned by CamScanner

✓ As a physical layer device, it generates the signal it receives.

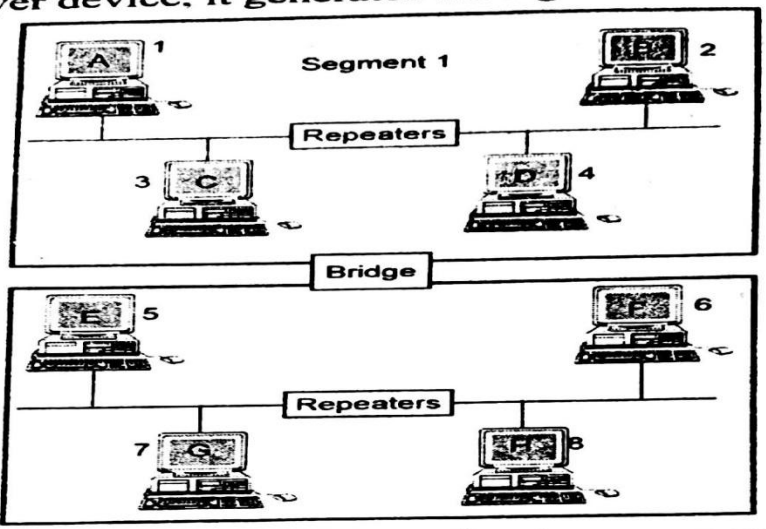


Fig. 10.8. Bridge connecting two segments

✓ When a frame enters a bridge, the bridge not only regenerates the signal but checks the address of the destination and forwards the new copy only to the segment to which the address belongs.

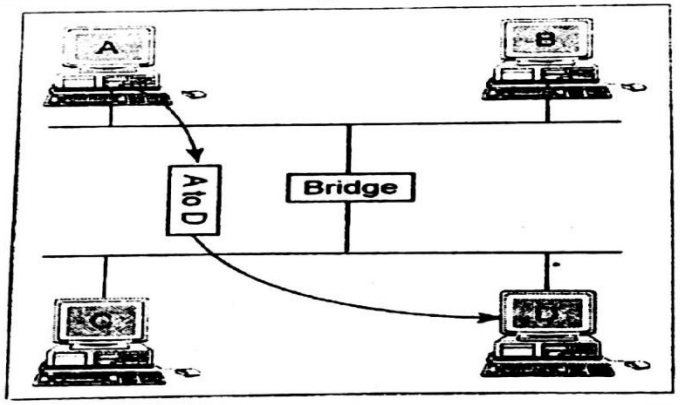
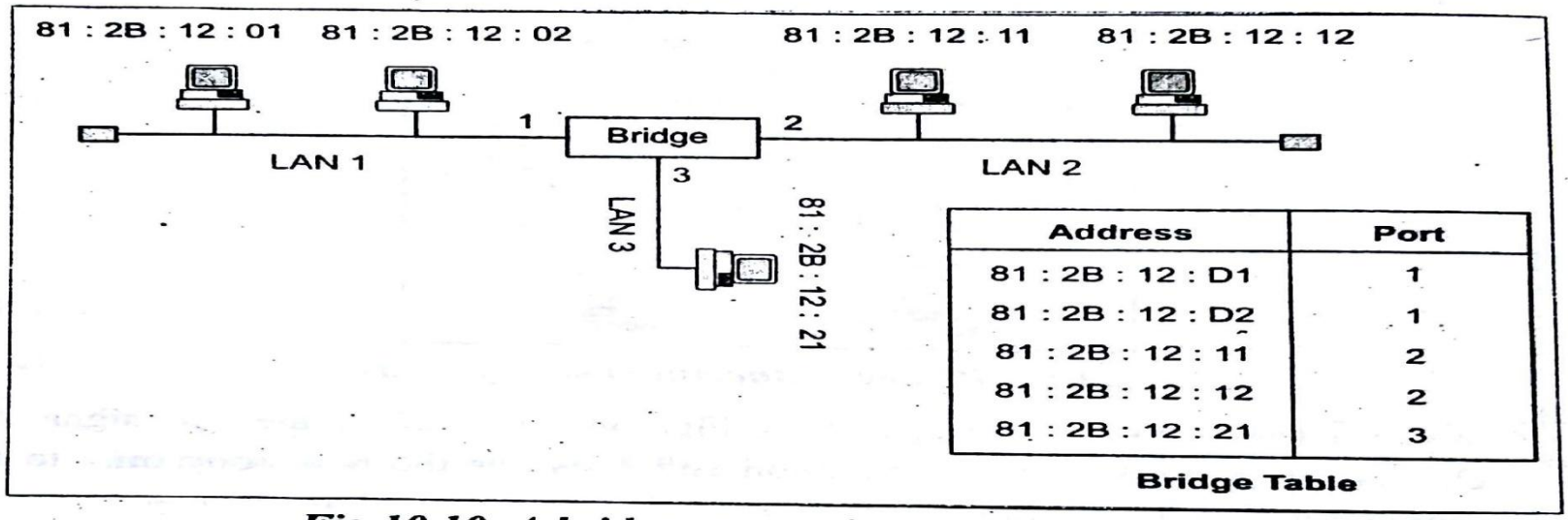


Fig. 10.9. A packet from A to D

✓ A packet generated by station A is intended for station D. The bridge allows the packet to cross and relays it to the entire lower segment, where it is received by station D.

Filtering: Filtering

- ✓ *A bridge has **filtering capability**. It can check the destination address of a frame and decide if the frame should be forwarded or dropped.*
- ✓ If the frame is to be forwarded, the decision must specify the port. A bridge has a table that maps addresses to ports.



Address	Port
81 : 2B : 12 : D1	1
81 : 2B : 12 : D2	1
81 : 2B : 12 : 11	2
81 : 2B : 12 : 12	2
81 : 2B : 12 : 21	3

Bridge Table

Fig.10.10. A bridge connecting LANs using table

10.1.3.2. Types of Bridges:

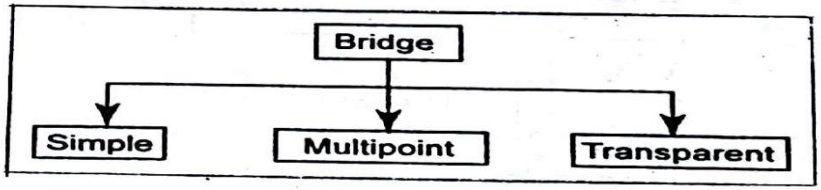


Fig.10.11. Types of bridges

- ***Simple bridge:***
  - ✓ *Simple bridges are the most **primitive** and **least expensive** type of bridge. A simple bridge connects two segments.*
  - ✓ Therefore, it maintains a table of host addresses versus segment numbers mapping for the two segments.
  - ✓ For example from the Fig. (Bridge connecting two segments)

Host address	Segment number
A	1
B	1
C	1
D	1
E	2
F	2
G	2
H	2

Fig.10.12. Host address to segment mapping

- ✓ This table has to be entered by an operator manually by doing data entry of all the host addresses and their segment numbers.
- ✓ Whenever a new host is added, or an existing host is replaced / deleted, the table has to be ***updated*** again.
- ✓ For these reasons, simple bridges are the ***cheapest***, but they also have a lot of scope for error due to ***manual intervention***.

Multiport Bridge:

- ✓ A multiport bridge is a special case of either the simple or the ***learning bridge***. *When a simple or learning bridge connects more than two network segments (LANs). It is called as* ***multiport bridge***.

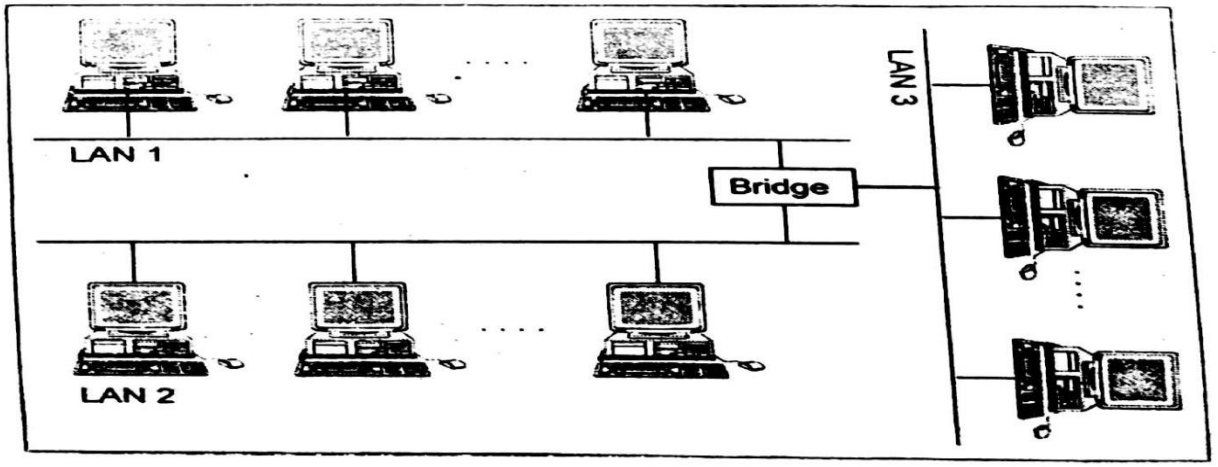


Fig.10.13. Multiport bridge

- ***Transparent bridge:***
  - ✓ *A transparent bridge can forward and filter frames and automatically build its forwarding table.*
  - ✓ A ***transparent*** also called as an ***adaptive bridge***, does not have to be programmed manually unlike a simple bridge. Instead, it performs its own bridging functions.
  - ✓ When the transparent bridge is first installed, its table is empty. As it encounters each packet, it looks as both destination and the source addresses. It checks the destination to decide where to send the packet.
  - ✓ If it does not yet recognize the destination address. It relays the packet to all of the stations on both segments.
  - ✓ It must meet ***three criteria***.
    - → **Forwarding** – Frames must be forwarded from one station to another.
    - → **Learning** – The forwarding table is automatically made by learning frame movements.
    - → **Looping** – Loops in the system must be prevented.