19Z504 Computer Networks

BE CSE

Syllabus

- INTRODUCTION: Building a Network - Network Edge and Core - Layering and Protocols - TCP/IP Protocol suite - OSI Reference Model - Network Topologies – Internet Architecture
- PHYSICAL LAYER: Signal Characteristics – Transmission media – Signal Encoding Techniques – Performance Metrics.
- LINK LAYER SERVICES: Link Layer Services - Framing - Flow Control - Error Control - Media Access Control - Ethernet - Wireless LAN–Introduction about Bluetooth, Zigbee.

Syllabus

- SWITCHING AND ROUTING : Switching : Circuit Switching - Packet Switching - IPV4 - Global Address - Datagram Forwarding - Subnetting - CIDR - ICMP - Routing Algorithms: Distance Vector Routing and Link State Routing - IPV6 Addressing–IPV6 Protocol.
- CONNECTION-ORIENTED AND CONNECTION-LESS SERVICES : Overview of Transport Layer - UDP - TCP – Reliable Byte Stream - Connection Management - Flow Control - Congestion Control - SCTP.

Syllabus

- APPLICATION LAYER SERVICES: Needs/Principles of Application Layer Protocols – Role of proxy, Web and HTTP - FTP - Electronic Mail (SMTP- POP3 - IMAP- MIME)- DHCP - DNS - DASH - QUIC.

- **Book**

- 1 . Larry L Peterson and Bruce S Davie, ”Computer Networks: A systems approach”, Morgan Kaufmann Publishers, USA, Fifth Edition 2011.
- 2. James F Kurose, Keith W Ross, ”Computer Networking - A Top-Down Approach Featuring the Internet”, Pearson Education, New Delhi, Sixth Edition, 2012.

Course Outcomes

- **COURSE OUTCOMES**
- Upon completion of this course, the students will be able to
- CO1: understand the components of network architecture and analyze the performance of computer networks with respect to key performance metrics.
- CO2: understand the design principles of physical and data link layers
- CO3: understand the design issues in host level connectivity, identify the IP address classes and design subnets for different scenarios
- CO4: understand the issues involved in establishing process to process connectivity and congestion management
- CO5: Outline the protocols for network application development and network management

INTRODUCTION

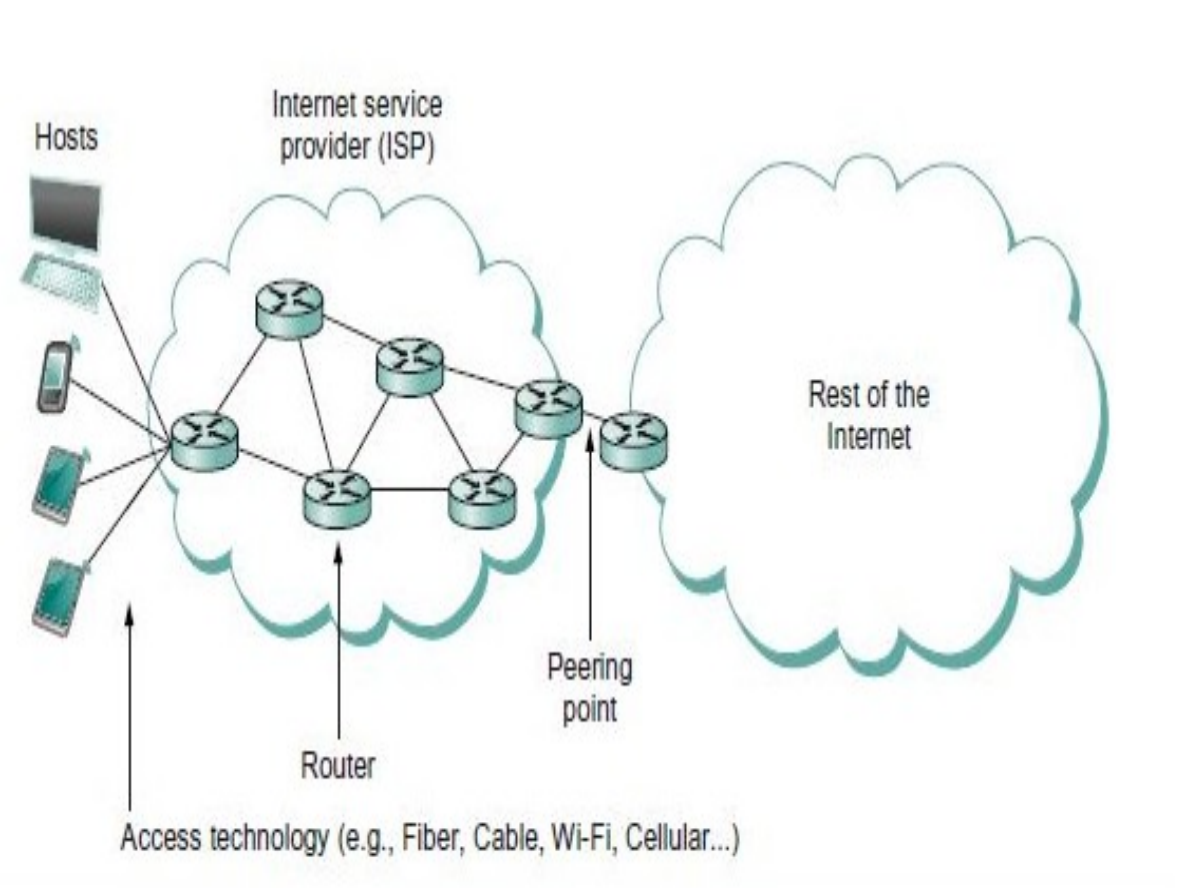
- Building a Network
- Network Edge and Core
- Layering and Protocols - TCP/IP Protocol suite
- OSI Reference Model
- Network Topologies
- Internet Architecture

Computer Network

- An interconnection of multiple devices, also known as hosts, that are connected using multiple paths for the purpose of sending/receiving data or media.
- include multiple **devices/mediums** which help in the communication between two different devices- **Network devices** and include things such as **routers, switches, hubs, and bridges**.

Hardware Building Blocks of Network

- Nodes
  - Computers and other devices in a network
- Links
  - Physical connection between the nodes by coaxial cables or optical fibers.



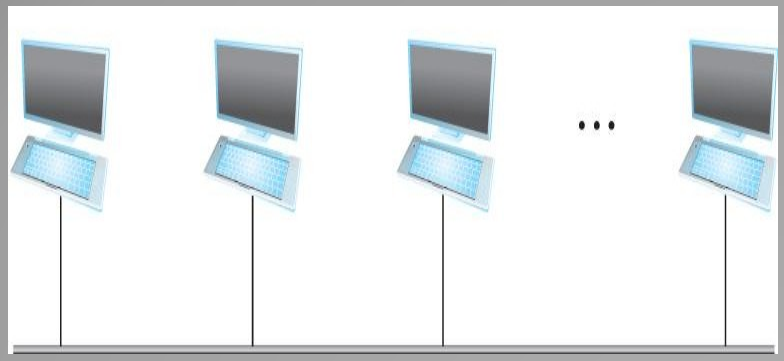
Connectivity : Direct connection

- Nodes are directly connected to each other over a **common physical medium**.
- **Types**
  - Point-to-Point
  - Multiple-access/Multi point
- **Problems**
  - Number of computers that could be connected by this kind of connection is very limited.
  - Number of wires coming out of each node can become unmanageable and expensive

Direct Connection

- Point-to-Point:
  - Links between a pair of nodes
- .

- Multiple-access:
  - More than two nodes share a single physical link.

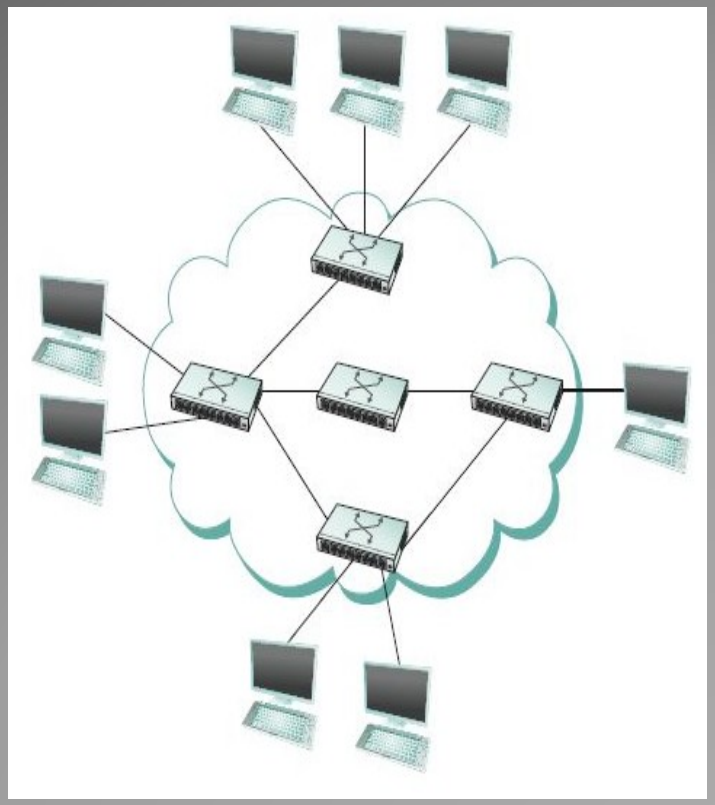


Connectivity: Indirect connection

- Nodes may be connected to each other by a set of cooperating nodes.
- **Types**
  - Switched Network
    - Circuit Switched
    - Packet Switched.
  - Internetwork

Indirect Connection

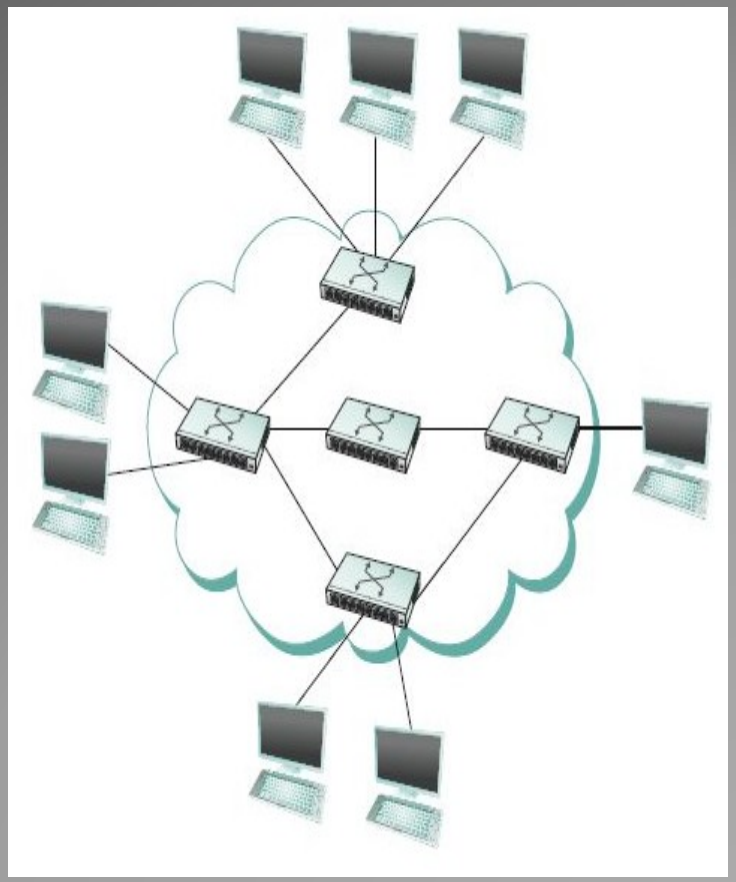
- Switched Network
  - Each nodes are attached to one or more point-to-point links.
  - Nodes that are attached to atleast two links run software that forwards data received from one link to another.
  - The forwarding nodes form switched network.
  - Types
    - Circuit switched
    - Packet switched



Switched Network

- Switches
  - Primary function
    - Store-and-forward packets in a packet switched connectivity.
- Hosts
  - Nodes that support users and run application programs.

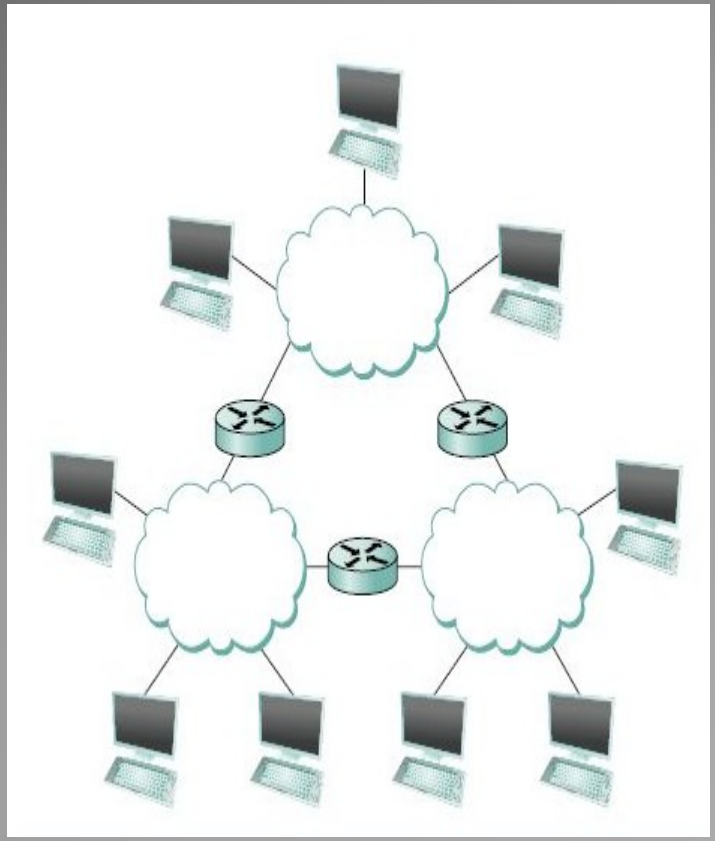
**The cloud in the diagram is used to represent any type of network(point-to-point, multiple-access or switched network)*



Indirect Connection

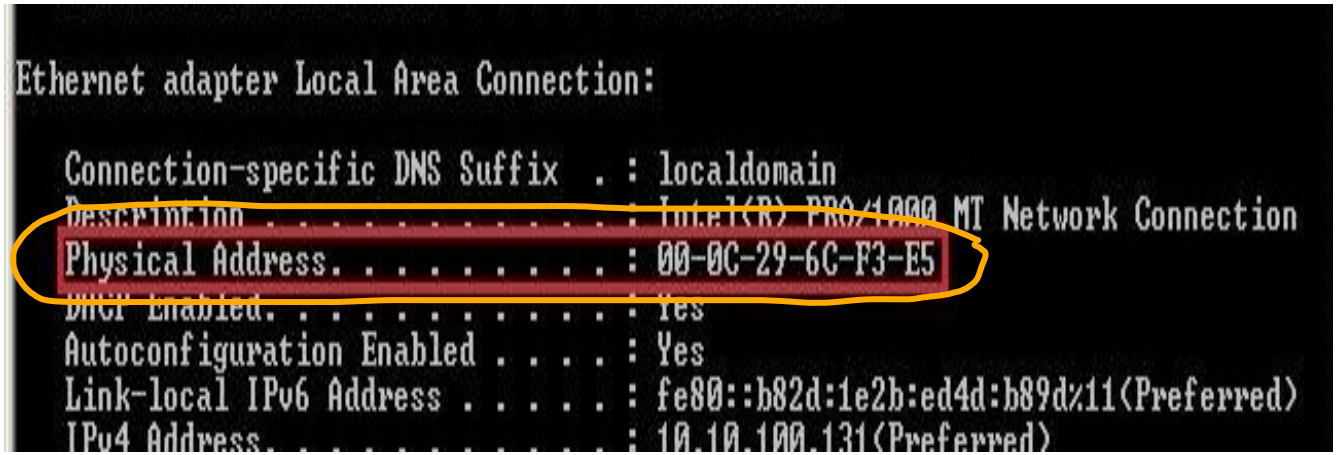
- Internetwork/Internet
  - Set of independent networks connected together
  - Router or gateway
    - A node that is connected to two or more networks.
    - It forwards messages from one network to another.

**Internet can be viewed as another kind of network-interconnection of internets.*



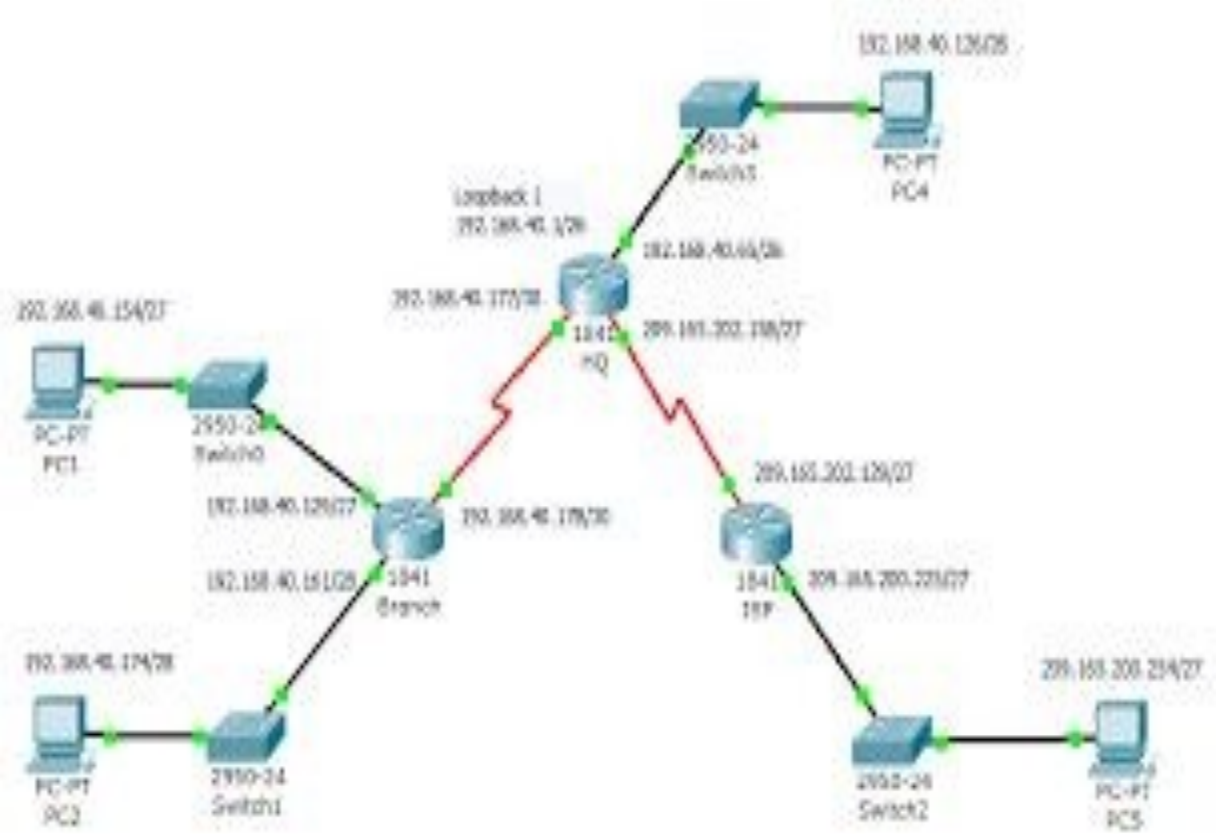
Addressing

- Byte string that identifies a node
- Network can use a node's address to distinguish it from the other nodes connected to the network.



Routing

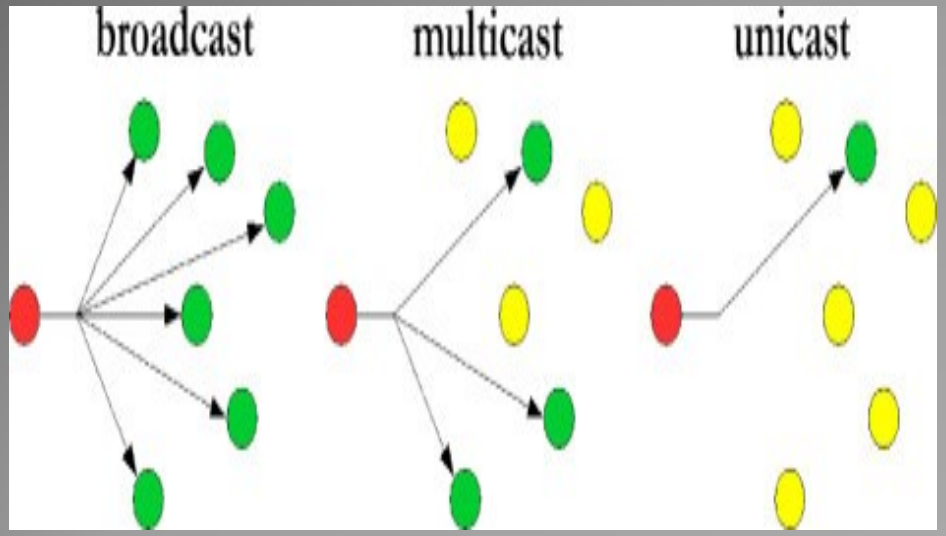
The process of determining systematically how to forward messages towards the destination node based on it address



Communication types

- Unicast
  - Source node sends a message to a single destination node
- Broadcast
  - Source node sends a message to all the nodes on the network.
- Multicast
  - Source node sends a message to a subset of nodes but not all nodes.

**A network must support multicast and broadcast address in addition to node-specific address*

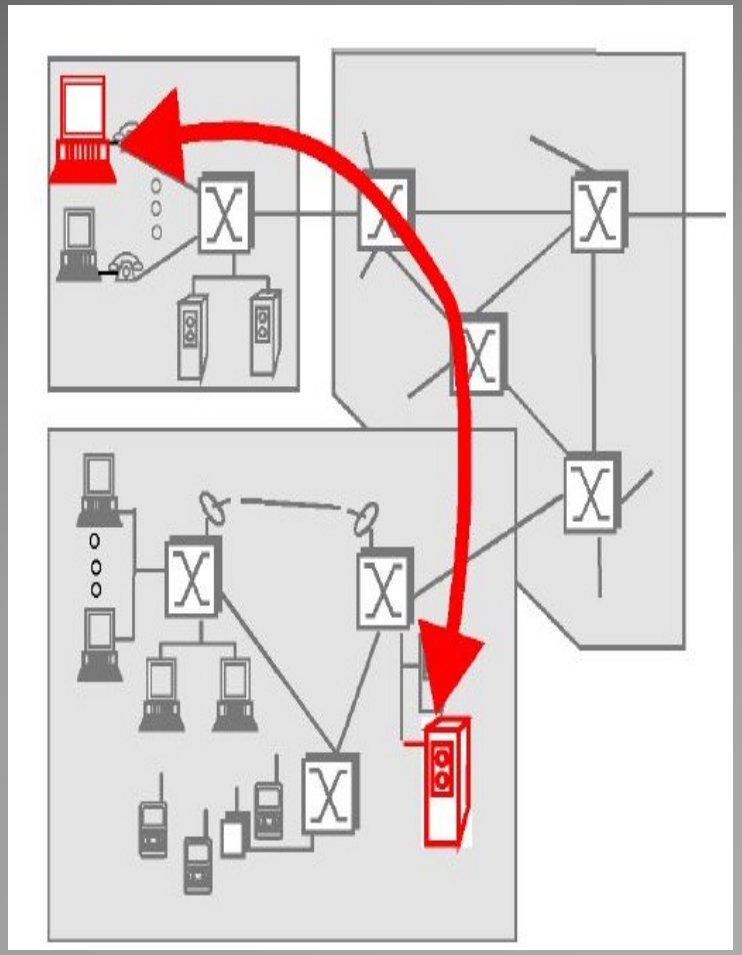


Networks

- Two or more nodes connected by a physical link.
- Two or more networks connected by a node.

The Network Edge

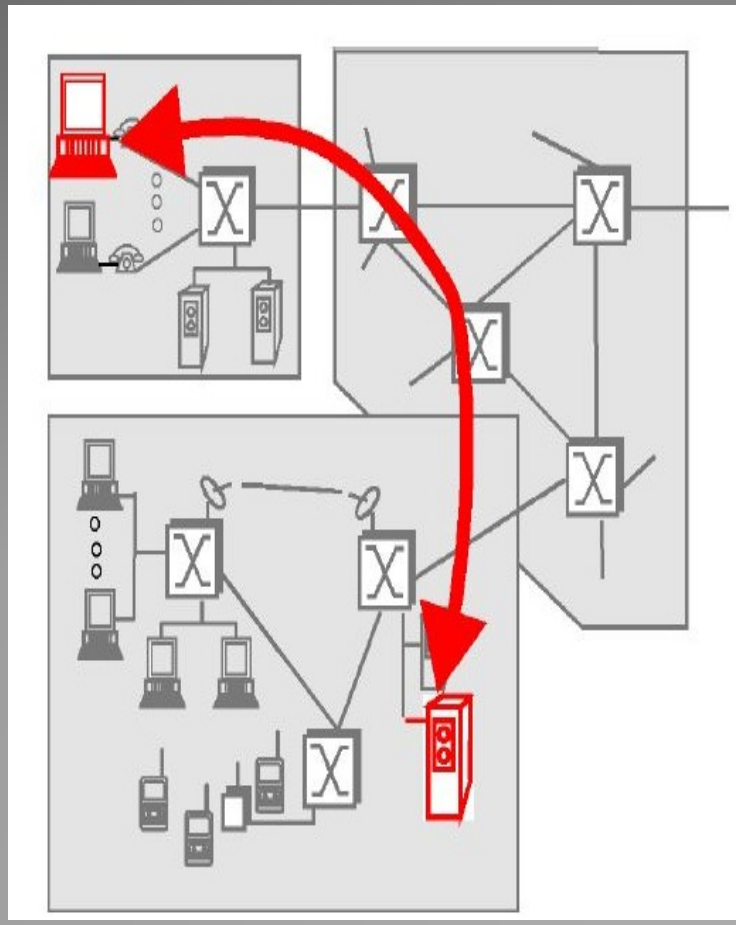
- Components which we use on daily basis
- Hosts
  - Computers that we use on a daily basis are referred to as hosts or end systems.
  - The run application level programs-Eg web browser, server program, email etc…
  - They sit at the “edge” of the Internet.
  - Two categories:
    - Client
    - Server



The Network Edge

- Clients
  - Often tends to be PCs, workstation, etc…
- Server
  - Often tends to be more powerful machines.

- A client program running on one end system requests and receives information from a server running on another end system



Connectionless & Connection-Oriented Services

- A developer creating an Internet application (e.g., an email application, a file transfer application, a Web application or an Internet phone application) must program the application to use one of these two services:
  - Connectionless
  - Connection Oriented

Connection Oriented

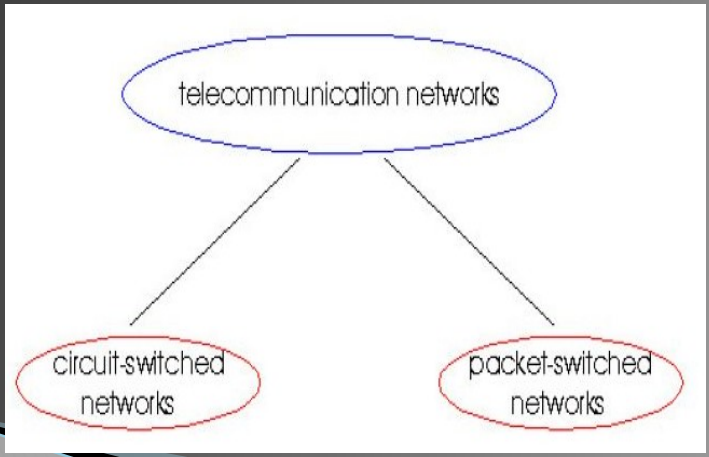
- When an application uses the connection-oriented service, the client and the server (residing in different end systems) send control packets to each other before sending packets with real data. Eg:e-mail etc.
- Handshaking procedure-alerts the client and server, allowing them to prepare for movement of packets
- Only the end systems themselves are aware of this connection.
- The components within the Internet are completely unaware to the connection
- The internet's connection oriented service-TCP(Transmission Control Protocol)

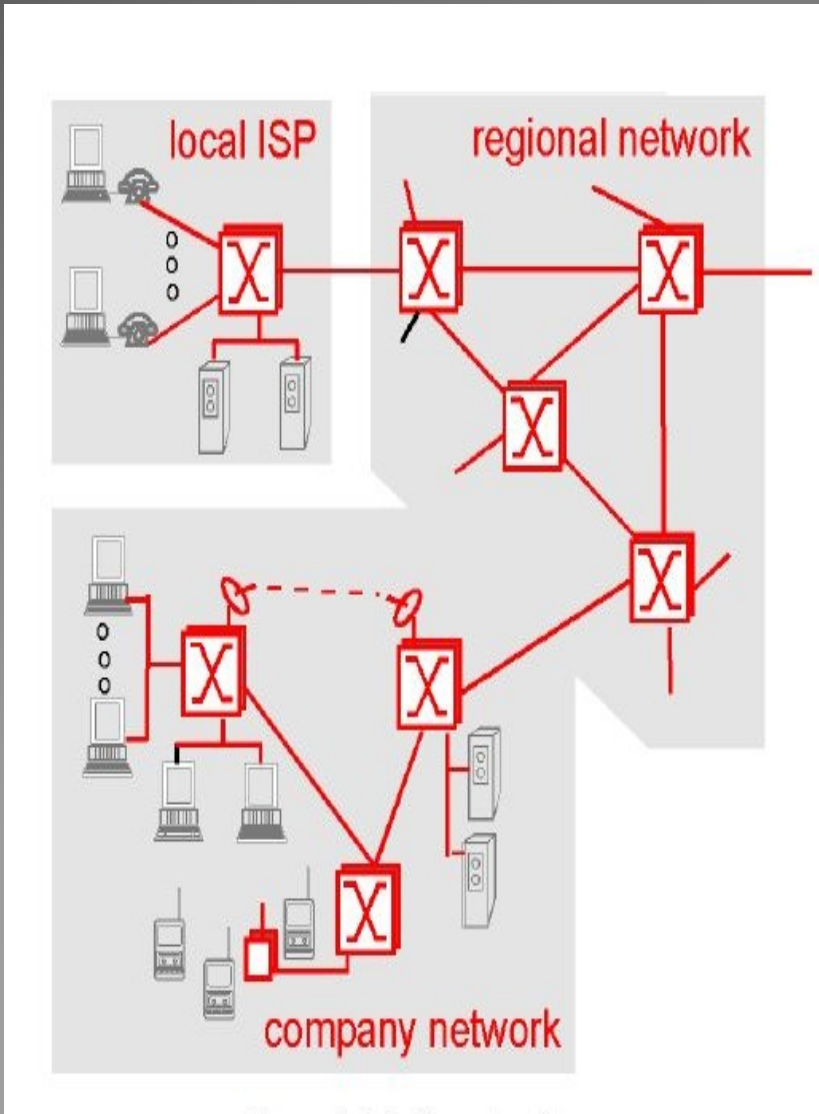
Connectionless

- One side of an application wants to send packets to another side of an application, the sending application simply sends the packets.
- Eg: Telnet(remote login), HTTP etc…
- No handshaking procedure.
- Hence faster delivery of data.
- The internet's connectionless service- UDP (User Datagram Protocol)

Network Core

- Inside of a network
- Two fundamental approaches in building network core
  - Circuit Switching
  - Packet Switching





Circuit Switching Vs Packet Switching

Circuit Switching	Packet Switching
In circuit switching there are 3 phases i) Connection Establishment. ii) Data Transfer. iii) Connection Released.	In Packet switching directly data transfer takes place.
In circuit switching, each data unit know the entire path address which is provided by the source	In Packet switching, each data unit just know the final destination address intermediate path is decided by the routers.
In Circuit switching, data is processed at source system only	In Packet switching, data is processed at all intermediate node including source system.
Delay between data units in circuit switching is uniform.	Delay between data units in packet switching is not uniform.

Circuit Switching Vs Packet Switching

Circuit Switching	Packet Switching
Delay between data units in circuit switching is uniform.	Delay between data units in packet switching is not uniform.
Resource reservation is the feature of circuit switching because path is fixed for data transmission.	There is no resource reservation because bandwidth is shared among users.
Wastage of resources are more in Circuit Switching	Less wastage of resources as compared to Circuit Switching
It is not a store and forward technique.	It is a store and forward technique.
Circuit switching is more reliable	Packet switching is less reliable.
Congestion can occur during connection establishment time	Congestion can occur during data transfer phase

Network Architecture

- A computer network must provide general, cost effective, fair, and robust connectivity among a large number of computers.
- To help deal with this complexity, network designers have developed general Blueprints usually called *network architectures* that *guide the design and implementation* of networks.

Layered Architecture

- A communication subsystem is a complex piece of Hardware and software.
- for implementing the software for such subsystems were based on a single, complex, unstructured program with many interacting components.
- The resultant software was very difficult to test and modify.
- To overcome such problem, the ISO has developed a layered approach.
- Networking concept is divided into several layers, and each layer is assigned a particular task.
  - networking tasks depend upon the layers.

Layered Architecture

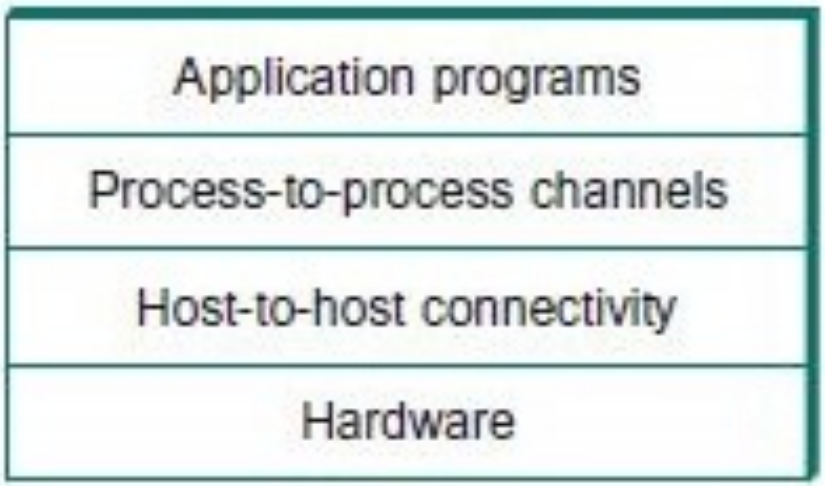
- The main aim of the layered architecture is to divide the design into small pieces.
- Each lower layer adds its services to the higher layer to provide a full set of services to manage communications and run the applications.
- It provides modularity and clear interfaces, i.e., provides interaction between subsystems.
- It ensures the independence between layers by providing the services from lower to higher layer without defining how the services are implemented. Therefore, any modification in a layer will not affect the other layers.

Layered Architecture

- The number of layers, functions, contents of each layer will vary from network to network.
- However, the purpose of each layer is to provide the service from lower to a higher layer and hiding the details from the layers of how the services are implemented.
- The basic elements of layered architecture are services, protocols, and interfaces.
  - **Service:** It is a set of actions that a layer provides to the higher layer.
  - **Protocol:** It defines a set of rules that a layer uses to exchange the information with peer entity. These rules mainly concern about both the contents and order of the messages used.
  - **Interface:** It is a way through which the message is transferred from one layer to another layer.
- In a layer n architecture, layer n on one machine will have a communication with the layer n on another machine and the rules used in a conversation are known as a layer-n protocol.

Features of layering

- Decompose the problem of building a network into more manageable components.
- Provides modular design-for adding new features/services, need to modify the functionality at one layer.



Protocols

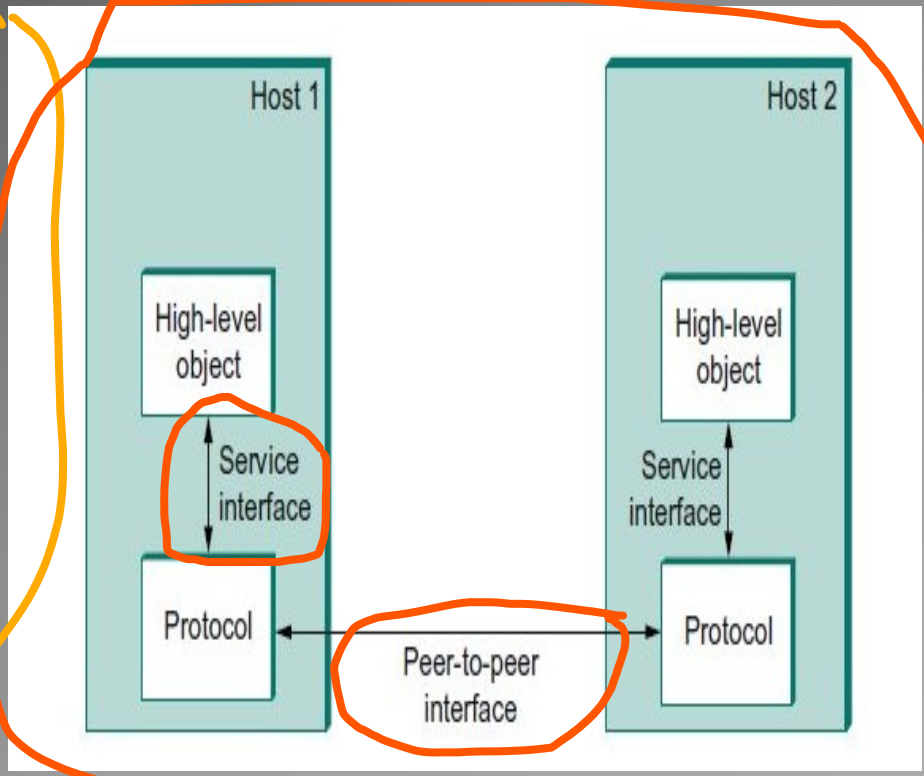
- A **protocol** is a standard set of rules that allow electronic devices to communicate with each other.
- These rules include what type of data may be transmitted, what commands are used to send and receive data, and how data transfers are confirmed.
- Provides a communication service that higher-level objects use to exchange messages.

Two interfaces of each protocols

- **Service interface**
  - Defines interface to other objects on the same computer that want to use its communication services.
  - Defines the operations that local objects can perform on the protocol.
- **Peer interface**
  - Defines interface to its counterpart(peer) on another machine.
  - Defines the form and meaning of messages exchanged between protocol peers to implement the communication service.

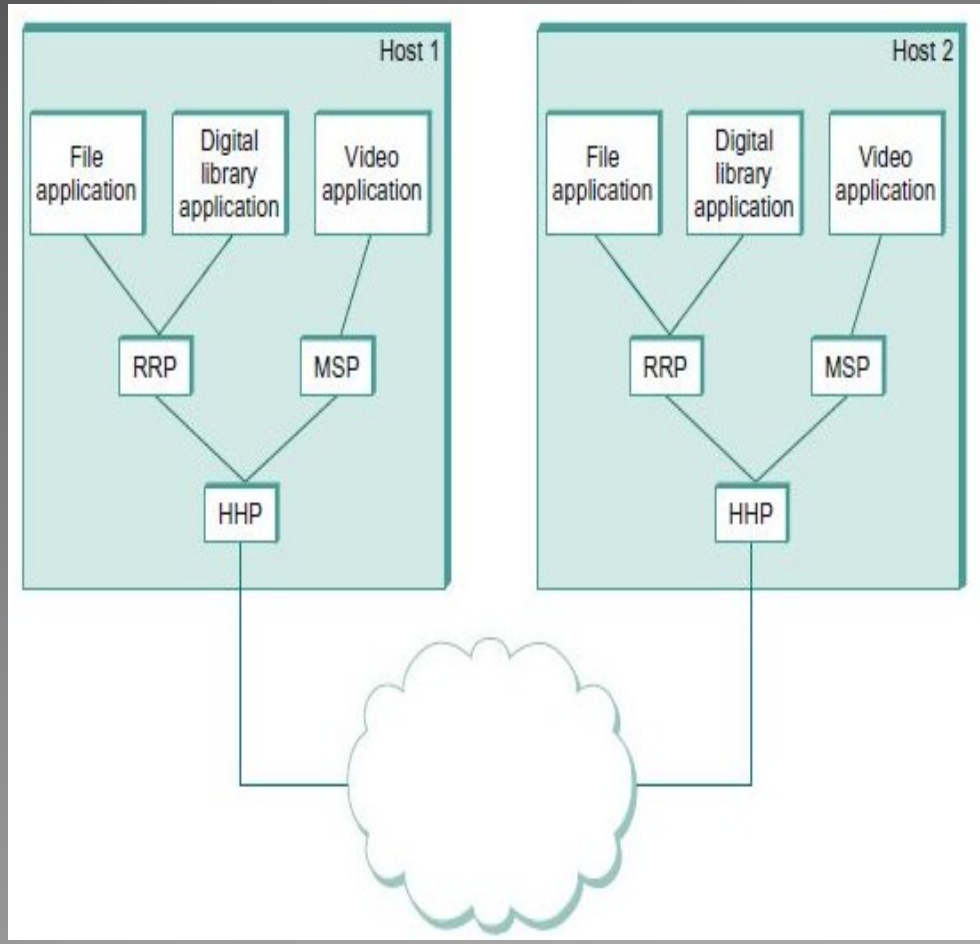
Two interfaces of each protocols

- Protocol defines a communication service that it exports locally(service interface) along with a set of rules governing the messages that the protocol exchanges with its peers to implement the service(peer interface)



Protocol Graph

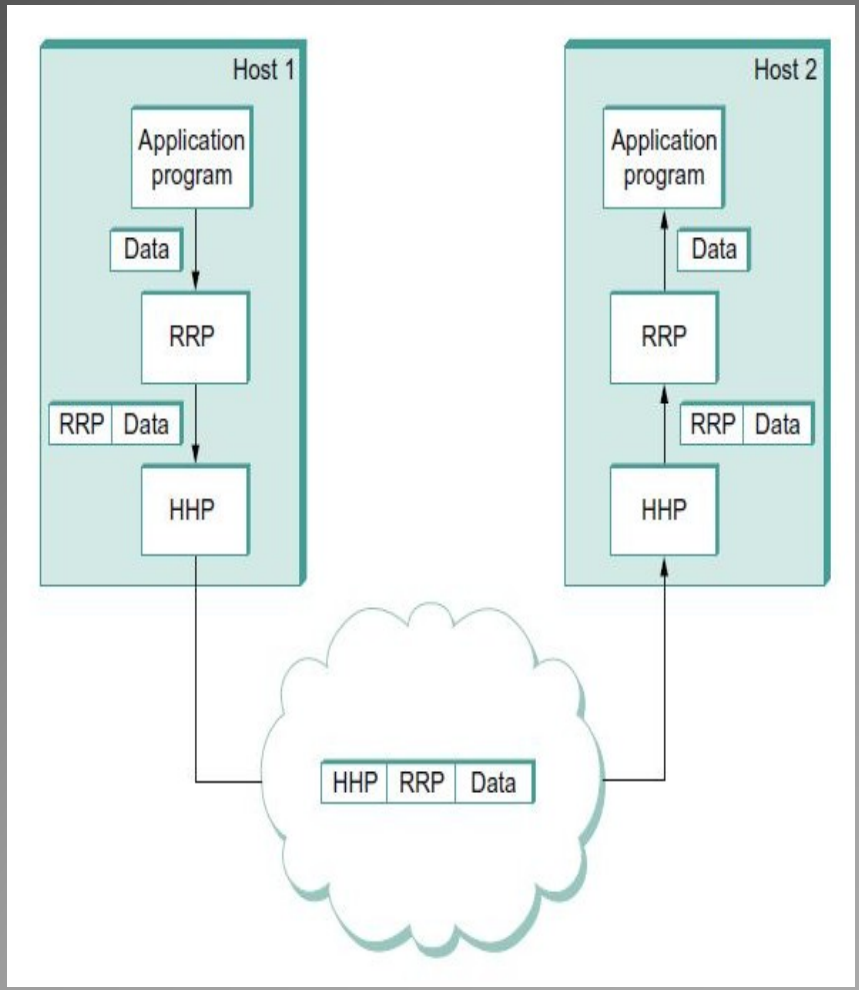
- Represent the suite of protocols that make up a network system with a *protocol graph*
- Each protocol communicates with its peer by passing messages to some lower level protocol, which in turn delivers the message to *its* peer.
- The nodes of the graph correspond to protocols,
- The edges represent a *depends on* relation



RRP- Request/Reply Protocol
MSP-Message Stream Protocol
HHP-Host-to-Host Protocol
*The application is said to employ the services of the *protocol stack* RRP/HHP.

Protocol Graph

- Consider application programs sends a message to its peer through RRP and HHP.
- information to its peer, instructing it how to handle the message when it is received.
- RRP attaches header to the message exact format for the header attached by RRP is defined by its protocol specification application's data is encapsulated in the new message created by RRP.
  - HHP then encapsulates RRP's message by attaching a header of its own
  - At destination, HHP first interprets the HHP header at the front of the message passes the body of the message up to RRP.



RRP takes whatever action is indicated by the RRP header that its peer attached and passes the body of the message up to the application program.

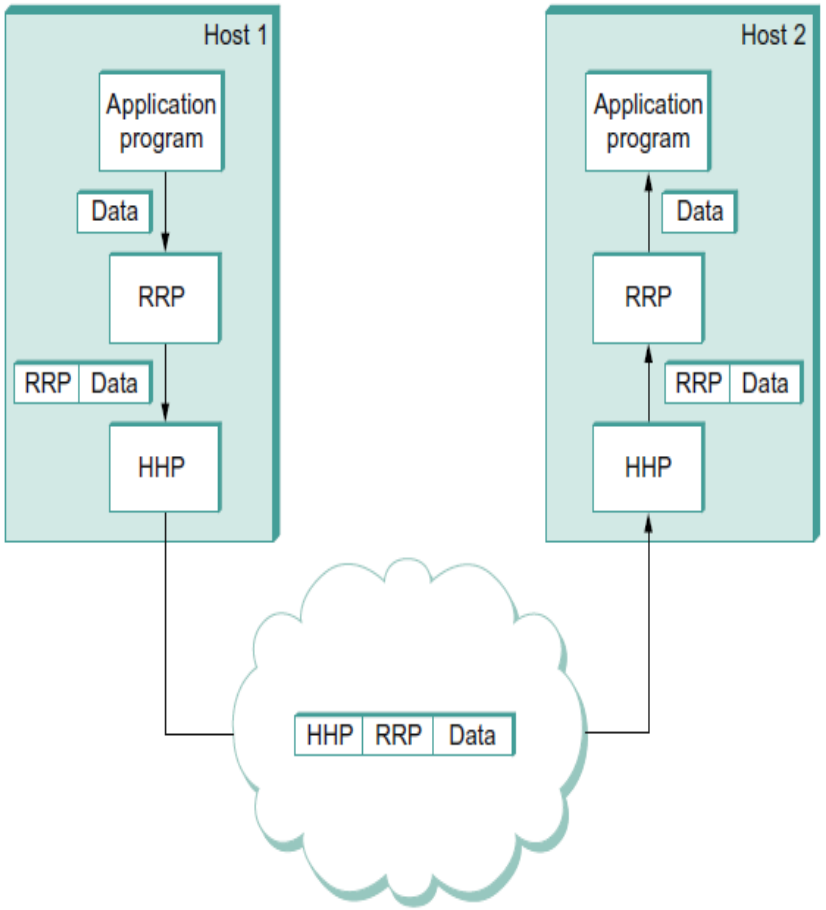
Encapsulation

Consider application programs sends a message to its peer through RRP and HHP.

•At source, RRP must communicate control information to its peer, instructing it how to handle the message when it is received RRP attaches *header* to the message exact format for the header attached by RRP is defined by its protocol specification application's data is *encapsulated* in the new message created by RRP.

• HHP then encapsulates RRP's message by attaching a header of its own At destination, HHP first interprets the HHP header at the front of the message passes the body of the message up to RRP.

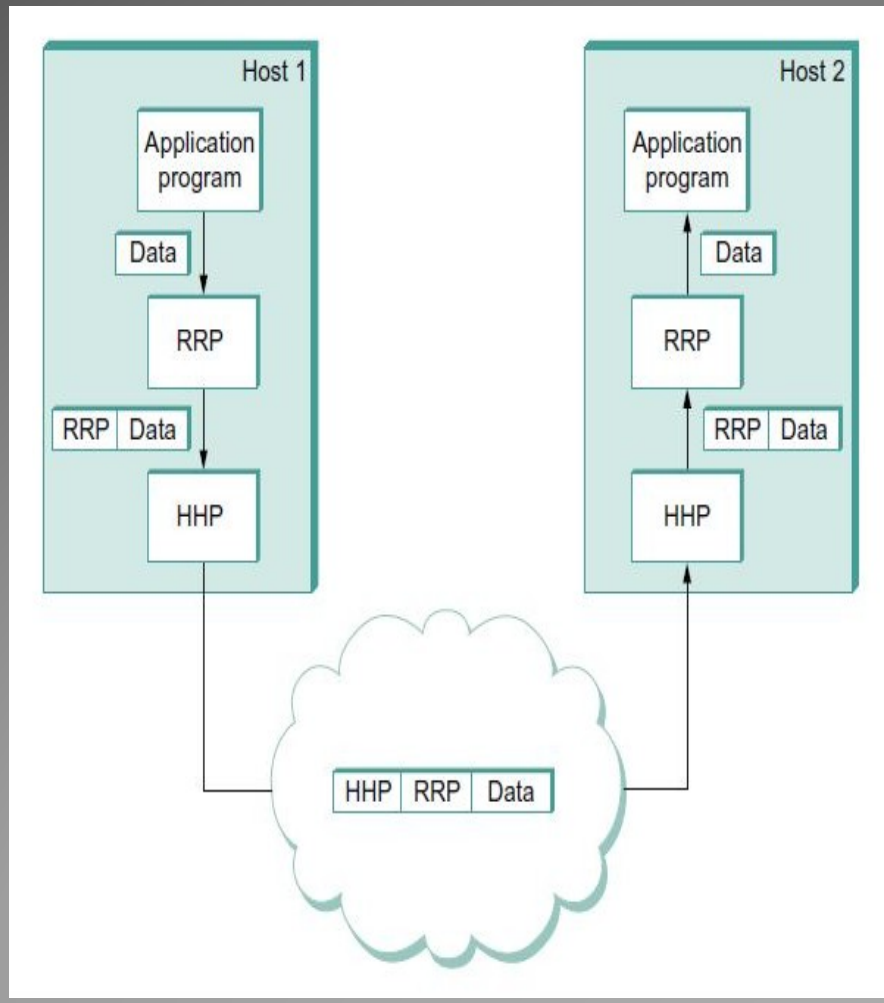
•RRP takes whatever action is indicated by the RRP header that its peer attached and passes the body of the message up to the application program.



Encapsulation

A header is a small data structure(few bytes) is used among
peers to communicate with each other. Is also attached to the front of a message.

- In some cases, control information is sent at the end of the message, in which case it is called a *trailer*.
- The data being transmitted on behalf of the application is called the message's *body* or *payload*
- Encapsulation is the operation, performed by a lower-level protocol, of attaching a protocol-specific header and/or trailer to a message passed down by a higher-level protocol.



Why do we require Layered architecture?

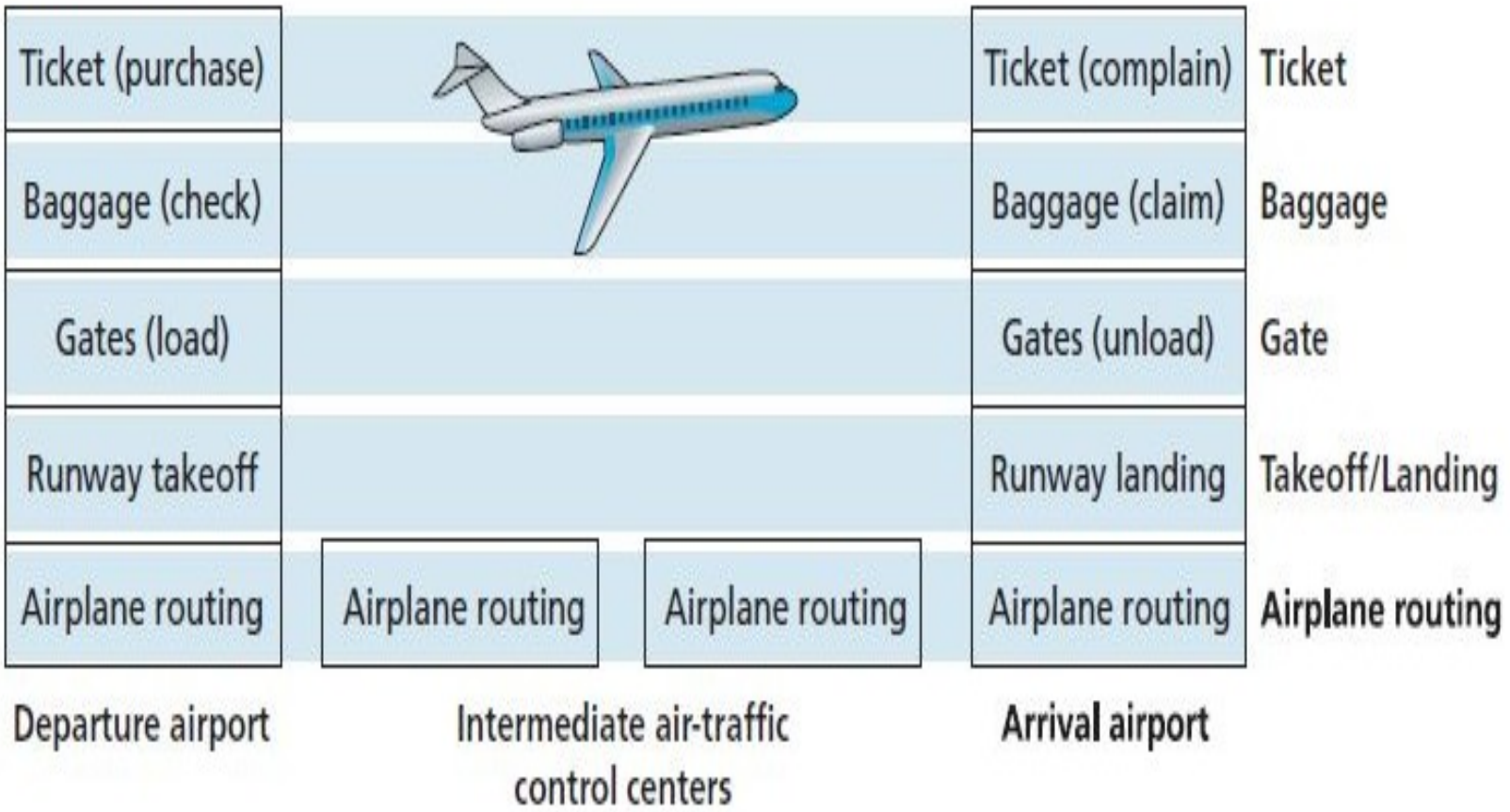
Divide-and-conquer approach: Divide-and-conquer approach makes a design process in such a way that the unmanageable tasks are divided into small and manageable tasks. In short, we can say that this approach reduces the complexity of the design.

Modularity: Layered architecture is more modular. Modularity provides the independence of layers, which is easier to understand and implement.

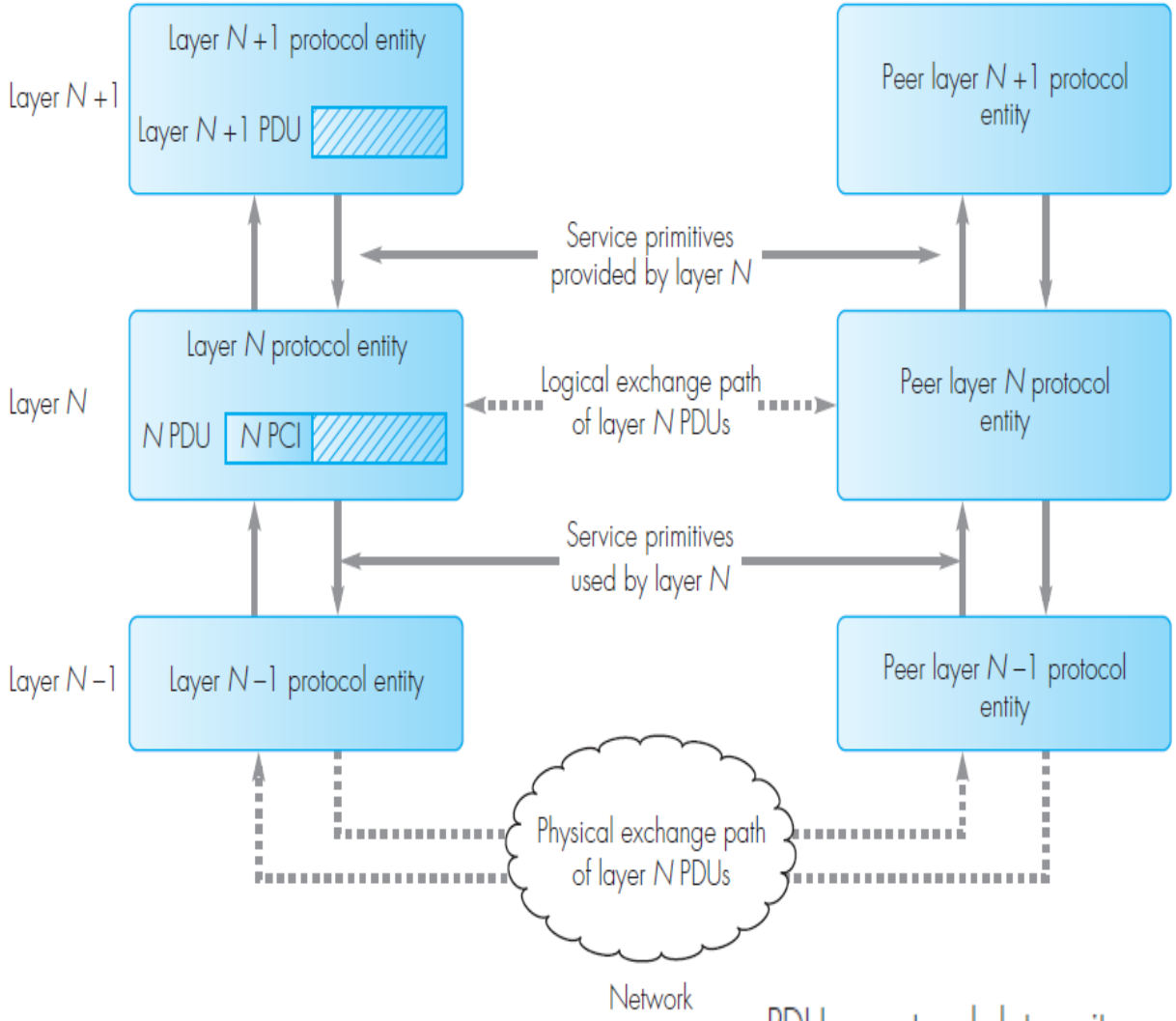
Easy to modify: It ensures the independence of layers so that implementation in one layer can be changed without affecting other layers.

Easy to test: Each layer of the layered architecture can be analyzed and tested individually.

Layered Approach!!!



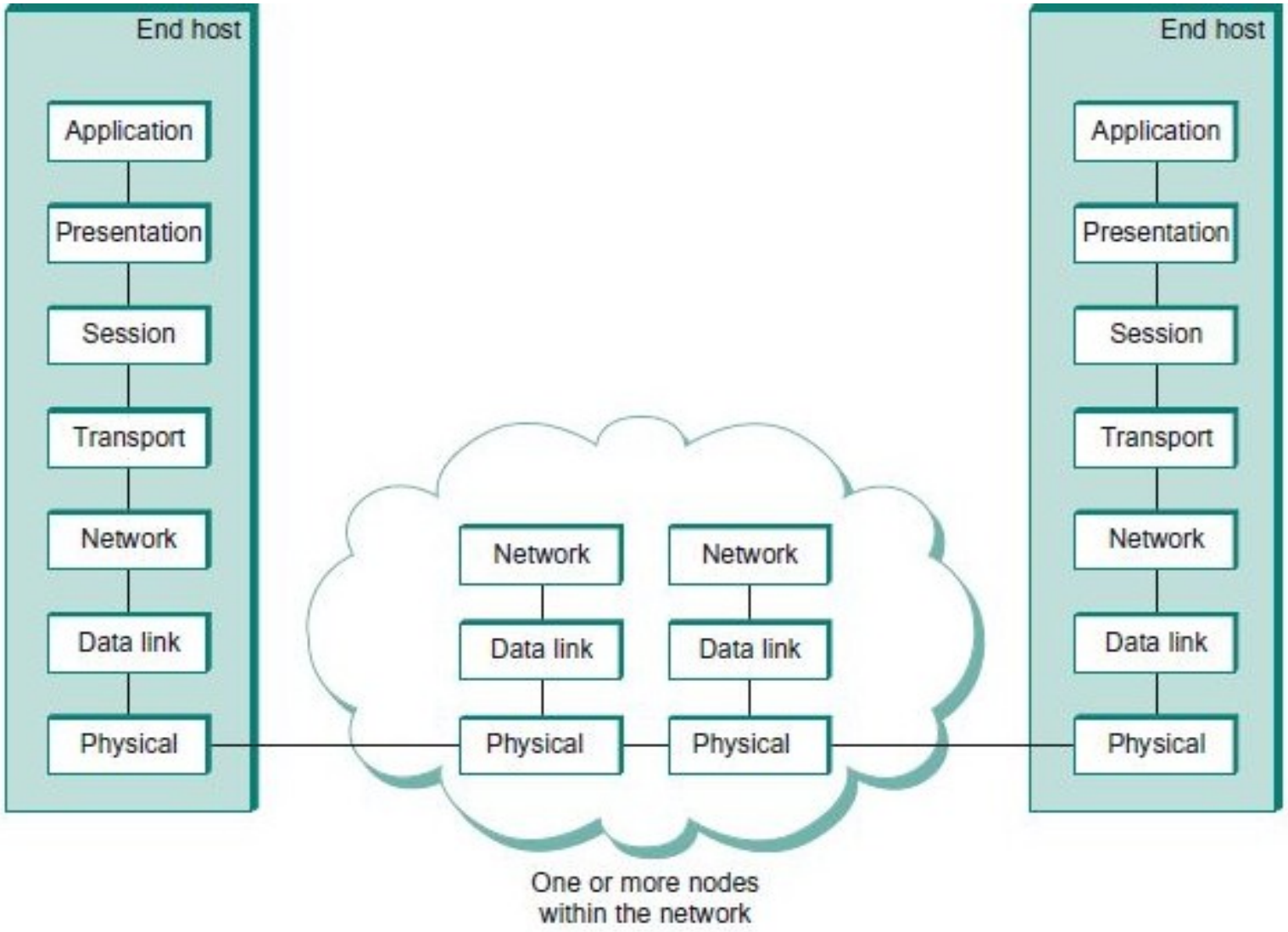
Layer Interactions



PDU = protocol data unit

PCI = protocol control information

7-Layer Model-Open Systems Interconnection(OSI) Architecture.



Layer 1: Physical layer

- 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.
- It also defines the procedures and functions that physical devices and interfaces have to perform for transmission to occur.

"The physical layer is responsible for movements of individual bits from one hop (node) to the next"

Layer 1: Physical layer

Services Provided

Physical characteristics of interfaces and medium.

✓The physical layer defines the characteristics of the interface between the devices and the transmission medium.

✓It also defines the type of transmission medium.

Representation of bits.

✓ The physical layer data consists of a stream of bits (sequence of 0s or 1s) with no interpretation.

✓ To be transmitted, bits must be encoded into signals--electrical or optical.

✓The physical layer defines the type of encoding (how 0s and 1s are changed to signals).

Data rate.

The transmission rate: the number of bits sent each second.

(ie) the physical layer defines the duration of a bit, which is how long it lasts.

Synchronization of bits.

The sender and receiver not only must use the same bit rate but also must be synchronized at the bit level.

(ie)the sender and the receiver clocks must be synchronized.

Physical layer

Line configuration.

The physical layer is concerned with the connection of devices to the media.

- ✓ **point-to-point configuration**: Two devices are connected through a dedicated link.
- ✓**multipoint configuration**: a link is shared among several devices.

Physical topology.

The physical topology defines how devices are connected to make a network.

- ✓**mesh topology:** every device is connected to every other device
- ✓**star topology**: devices are connected through a central device
- ✓**ring topology**: each device is connected to the next, forming a ring
- ✓**bus topology:** every device is on a common link
- ✓**hybrid topology**: this is a combination of two or more topologies

Transmission mode

The physical layer also defines the direction of transmission between two devices

- ✓**simplex mode:** only one device can send; the other can only receive(a one-way communication).
- ✓**half-duplex mode**: two devices can send and receive, but not at the same time.
- ✓**full-duplex** (or simply duplex) mode: two devices can send and receive at the same time.

Layer 2: Data link layer

The data link layer transforms the physical layer, a raw transmission facility, to a reliable link.

✓Collects a stream of bits into a larger aggregate called a frame. Frames are actually delivered to the host.

It makes the physical layer appear **error-free** to the upper layer (network layer).

✓Network Adaptors along with device drivers running in the node's operating system implement the data link level.

✓Oversees the delivery of the data units between two systems on the same network (links)

"The data link layer is responsible for moving frames from one hop (node) to the next"

Services Provided

Framing

✓ The data link layer divides the stream of bits received from the network layer into manageable data units called frames.

Data link layer

Services Provided

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.
- ✓ If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects the network to the next one.

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.

Error control.

- ✓ The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames.
- ✓ It also uses a mechanism to recognize duplicate frames.

Error control is normally achieved through a trailer added to the end of the frame.

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.

Layer 3: 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.
- 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 unit of data exchanged among nodes is typically called packet.

"The network layer is responsible for the delivery of individual packets from the source host to the destination host"

Network layer

Services Provided

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, among other things, includes the logical addresses of the sender and receiver.

Routing.

✓When independent networks or links are connected to create *intenetworks* (network of networks) or a large network, the connecting devices (called *routers* or *switches)* route or switch the packets to their final destination.

✓ The network layer gets each packet to the correct computer.

Layer 4: 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.

  It treats each one independently, as though each piece belonged to a separate message, whether or not it does.
- 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.

  "The transport layer is responsible for the delivery of a message from one process to another."

Services Provided:

Service-point addressing

✓Computers often run several programs at the same time.

✓source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on one computer to a specific process (running program) on the other.

✓The transport layer header must therefore include a type of address called **a service-point address (or port address)**.

✓Gets the entire message to the correct process on a computer.

Transport layer

Segmentation and reassembly.

✓ A message is divided into transmittable segments, with each segment containing a **sequence number.**

✓These numbers enable the transport layer **to reassemble the message** correctly upon arriving at the destination and to identify and replace packets that were lost in transmission.

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.

Flow control.

Like the data link layer, the transport layer is responsible for flow control. Flow control at this layer is performed **end to end rather than across a single link**.

Error control.

✓ error control at this layer is performed **process-to-process** rather than across a single link.

✓The sending transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, or duplication).

✓Error correction is usually achieved through retransmission.

Layer 5: Session layer

- The services provided by the first three layers (physical, data link, and network) are not sufficient for some processes.
- The session layer is the network *dialog controller.*
- It establishes, maintains, and synchronizes the interaction among communicating systems.

"The session layer is responsible for dialog control and synchronization."

Services Provided:

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.

Session layer

Synchronization.

✓The session layer allows a process to **add checkpoints**, or 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.

✓ Managing an audio and a video stream that are being combined in a teleconferencing application.

Layer 6: Presentation layer

The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems.

(ie) Concerned with the format of data exchanged between peers.

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

Services Provided:

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. Because different computers use different encoding systems, the presentation layer is responsible for **interoperability between these different encoding methods.**
- ✓The presentation layer at the sender changes the information from its sender-dependent format into a common format.
- ✓The presentation layer at the receiving machine changes the common format into its receiver-dependent format.

Eg:

whether integer is 16,32 or 64 bits long;
whether the most significant byte is transmitted first of last how the video streaming is formatted

Presentation layer

Services Provided

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.

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.

Layer 7: Application layer

- ✓ The application layer enables the user, whether human or software, to access the network.
- ✓ It provides user interfaces and support for services such as electronic mail, remote file access and transfer, shared database management, and other types of distributed information services.

"The application layer is responsible for providing services to the user."

Services Provided:

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.

✓To do so, the application creates a software emulation of a terminal at the remote host.

✓The user's computer talks to the software terminal which, in turn, talks to the host, and vice versa.

✓The remote host believes it is communicating with one of its own terminals and allows the user to log on.

The lower three layers are implemented on all network nodes.

The transport and the higher layers run only on end hosts

Application layer

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,

✓to manage or control files in a remote computer locally.

Mail services.

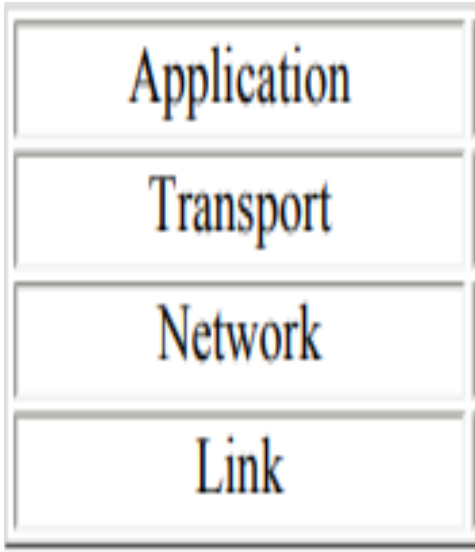
This application provides the basis for e-mail forwarding and storage.

Directory services.

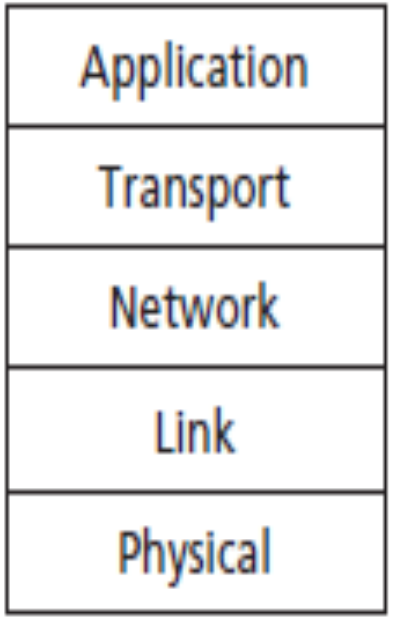
This application provides distributed database sources and access for global information about various objects and services.

TCP/IP Protocol Suite

- The TCP/IP protocol suite was developed prior to the OSI model.
- The layers in the TCP/IP protocol suite do not exactly match those in the OSI model.
- The original TCP/IP protocol suite was defined as having four layers.
- Internet's TCP/IP protocol suite is made of five layers
- TCP/IP is a hierarchical protocol made up of interactive modules, *each of which* provides a specific functionality;.
  - ✓ The modules are not necessarily interdependent.
  - ✓The term *hierarchical* means that each upper-level protocol is supported by one or more lower-level protocols.



•The layers of the TCP/IP protocol suite contain relatively independent protocols that can be mixed and matched depending on the needs of the system.



Physical and Data Link Layers

- At the physical and data link layers, *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.

Network Layer

- ARP, Internetworking Protocol(IP), is a host-to-host protocol, meaning that it can deliver a packet from one physical device to another.
- At the network layer (internetwork layer), *TCP/IP supports* the Internetworking Protocol(IP), in turn, uses four supporting protocols:
  - ✓RARP,
  - ✓ICMP,
  - ✓IGMP.

Network Layer

Internetworking Protocol (IP)

The Internetworking Protocol (IP) is the transmission mechanism used by the TCP/IP protocols.

- It is an unreliable and connectionless protocol with a best-effort delivery service.

  The term *best effort* *means* *that IP provides* *no error checking or tracking*. *IP assumes* the unreliability of the underlying layers and does its best to get a transmission through its destination, but **with no guarantees**.

- IP transports data in packets called *datagrams, each of which is transported separately.*
- Datagrams can travel along different routes and can arrive out of sequence or be duplicated.
- IP does not keep track of the routes and has no facility for reordering datagrams once they arrive at their destination.
- IP provides bare-bones transmission functions that free the user to add only those facilities necessary for a given application and thereby allows for maximum efficiency.

Network Layer

Address Resolution Protocol

The Address Resolution Protocol (ARP) is used to associate a logical address with a physical address.

On a typical physical network, such as a LAN, each device on a link is identified by a physical or station address, usually imprinted on the network interface card (NIC).

ARP is used to find the physical address of the node when its Internet address is known

Reverse Address Resolution Protocol

The Reverse Address Resolution Protocol (RARP) allows a host to discover its Internet address when it knows only its physical address.

It is used when a computer is connected to a network for the first time or when a disk of computer is booted.

Network Layer

Internet Control Message Protocol

The Internet Control Message Protocol (ICMP) is a mechanism used by hosts and gateways to send notification of datagram problems back to the sender.

ICMP sends query and error reporting messages.

Internet Group Message Protocol

The Internet Group Message Protocol (IGMP) is used to facilitate the simultaneous transmission of a message to a group of recipients.

Transport Layer

Traditionally the transport layer was represented in *TCP/IP by two protocols:*

TCP

UDP

UDP and TCP are transport level protocols responsible for delivery of a message from a process (running program) to another process.

A new transport layer protocol) , has been devised to meet the needs of some newer applications.

Transport Layer

User Datagram Protocol

The User Datagram Protocol (UDP) is the simpler of the two standard TCP/IP transport protocols.

It is a process-to-process protocol that adds only port addresses, checksum error control, and length information to the data from the upper layer

Transmission Control Protocol

The Transmission Control Protocol (TCP) provides full transport-layer services to applications.

TCP is a reliable stream transport protocol.

The term *stream, in this context,* means connection-oriented:

A connection must be established between both ends of a transmission before either can transmit data.

At the sending end of each transmission, TCP divides a stream of data into smaller units called *segments.*

Each segment includes a sequence number for reordering after receipt, together with an acknowledgment number for the segments received.

Segments are carried across the internet inside of IP datagrams.

At the receiving end, TCP collects each datagram as it comes in and reorders the transmission based on sequence numbers.

Stream Control Transmission Protocol

The Stream Control Transmission Protocol (SCTP) provides support for newer applications such as voice over the Internet.

It is a transport layer protocol that combines the best features of UDP and TCP.

Application Layer

The application layer in TCP/IP is equivalent to the combined session, presentation, and application layers in the OSI model Many protocols are defined at this layer.

Egs: The Internet's application layer includes many protocols:

HTTP protocol - which provides for Web document request and transfer,

SMTP - which provides for the transfer of e-mail messages,

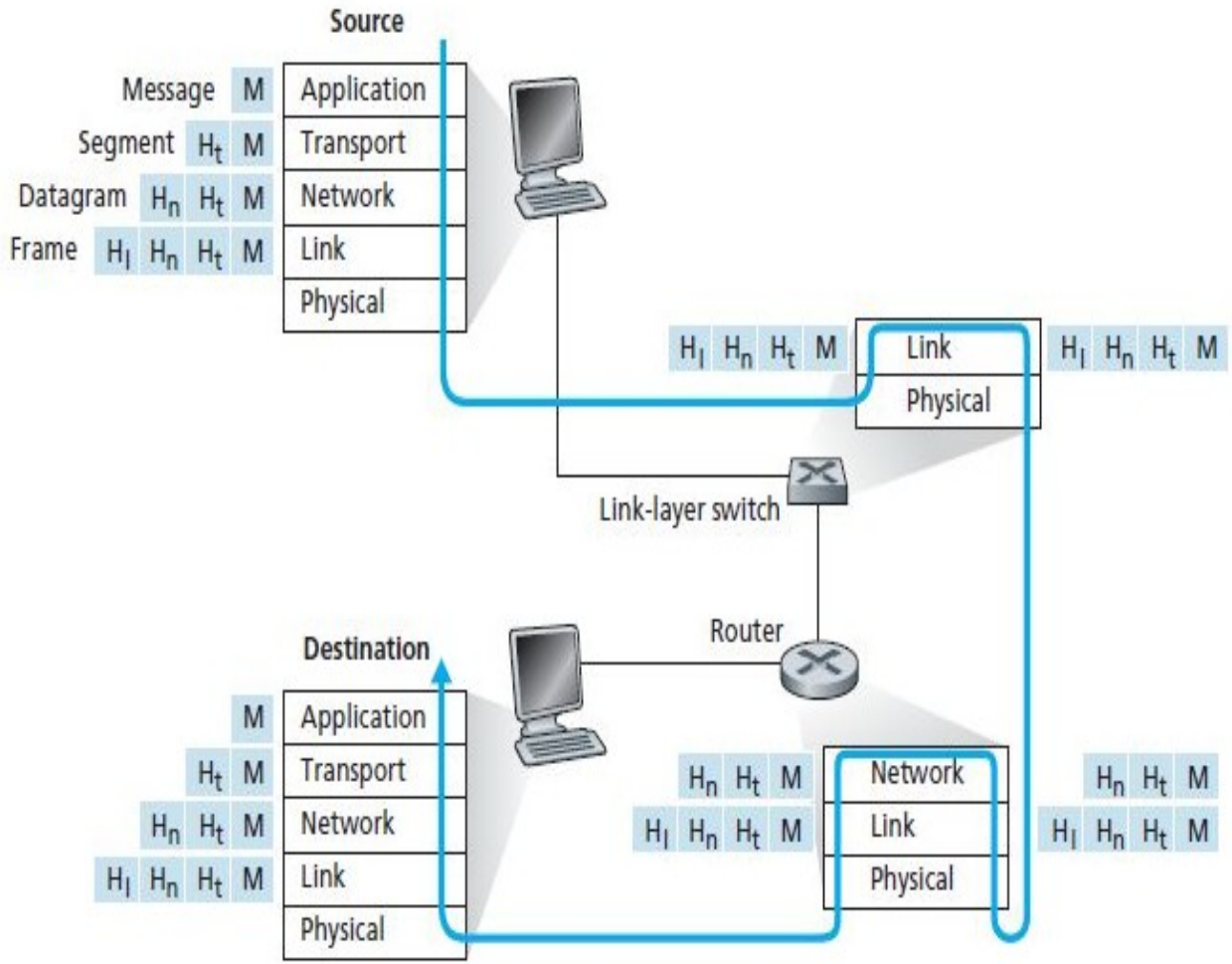
FTP - which provides for the transfer of files between two end systems

TCP/IP Architecture (Encapsulation)

- Suppose Alice, who is in one branch office, wants to send a memo to Bob, who is in another branch office.
  - The ***memo*** is analogous to the *application-layer message*.
  - Alice puts the memo in an interoffice envelope with Bob's name and department written on the front of the envelope.
  - The ***interoffice envelope*** is analogous to a *transport-layer segment*—it contains header information (Bob's name and department) and it encapsulates the application-layer message (the memo).
  - When the sending branch-office mailroom receives the interoffice envelope, it puts the interoffice envelope inside yet another envelope, which is suitable for sending through the public postal service.
  - The sending mailroom also writes the postal address of the sending and receiving branch offices on the postal envelope.
  - Here, the ***postal envelope*** is analogous to the ***datagram***—it encapsulates the transport-layer segment (the interoffice envelope), which encapsulates the original message (the memo).
  - The postal service delivers the postal envelope to the receiving branch office mailroom. There, the process of ***de-encapsulation*** is begun.
  - The mailroom extracts the interoffice memo and forwards it to Bob.
  - Finally, Bob opens the envelope and removes the memo.

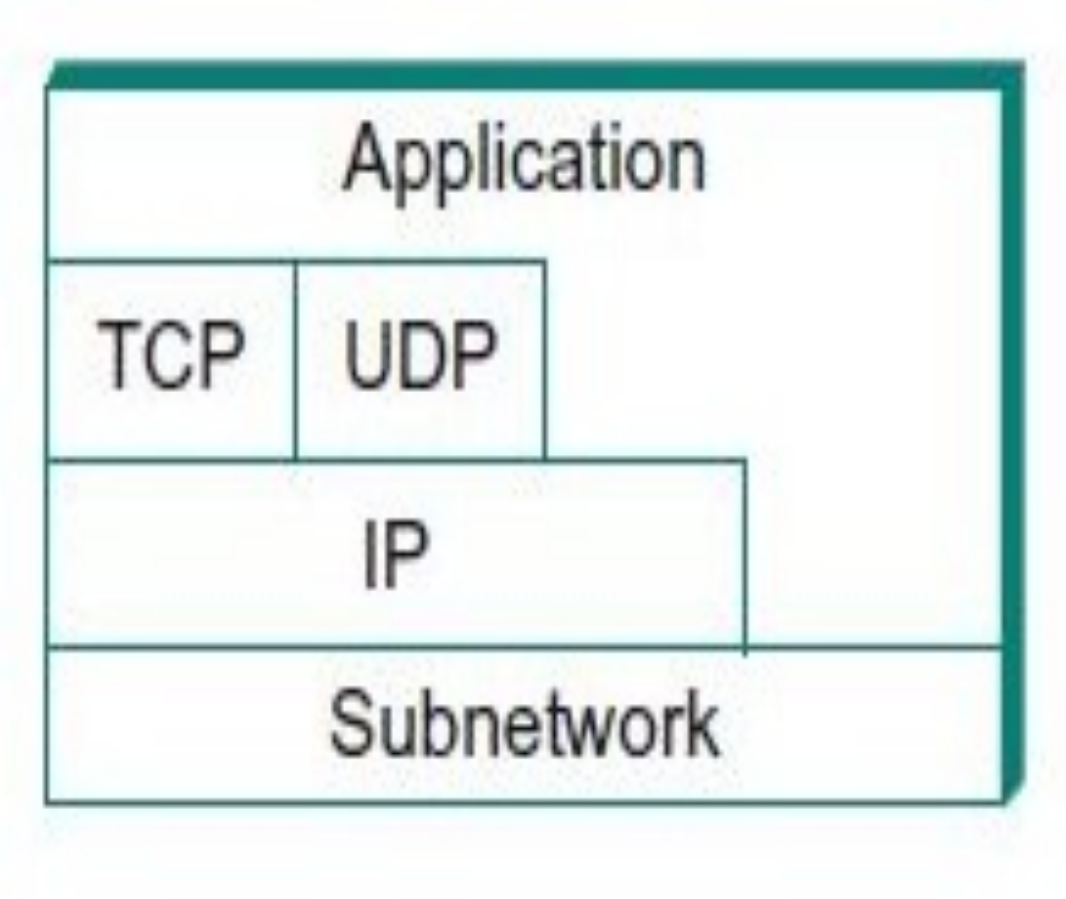
Encapsulation - Information is binded as a header
De-encapsulation - Information from header is extracted

TCP/IP Architecture



Older model of TCP/IP often show only four layers, combining the physical and data link layers.

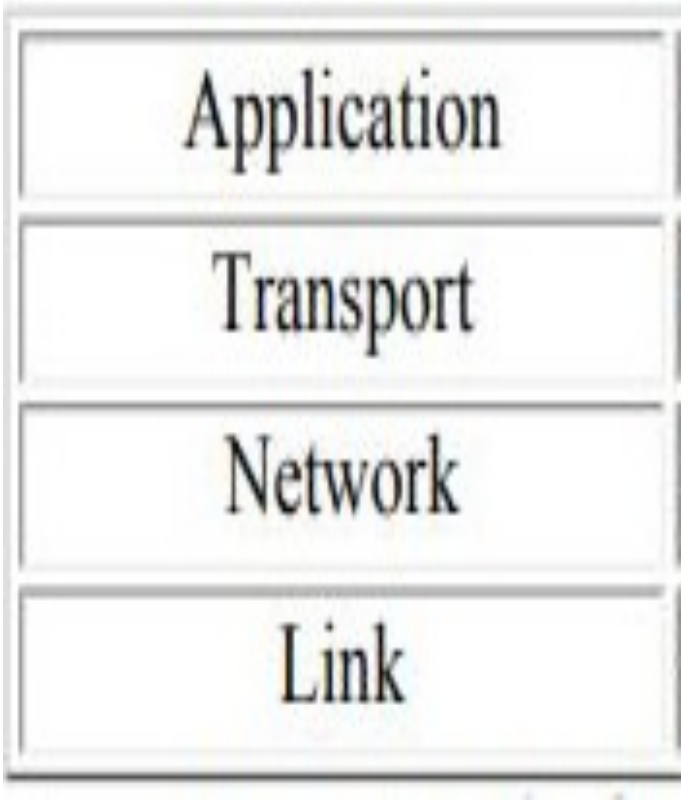
4-Layer Model-TCP/IP Architecture



Features of Internet Architecture

- Does not imply strict layering
  - Eg: application is free to bypass the defined transport layers and to directly use IP or one of the underlying networks.
  - Programmers are free to define new channel abstractions or applications that run on top of any of the existing protocols.
- IP servers as the focal point for the architecture
  - Defines a common method for exchanging packets among a wide collection of networks.
- In order for a new protocol to be officially included in the architecture, there must be both a protocol specification and at least one representative implementations of the specification.

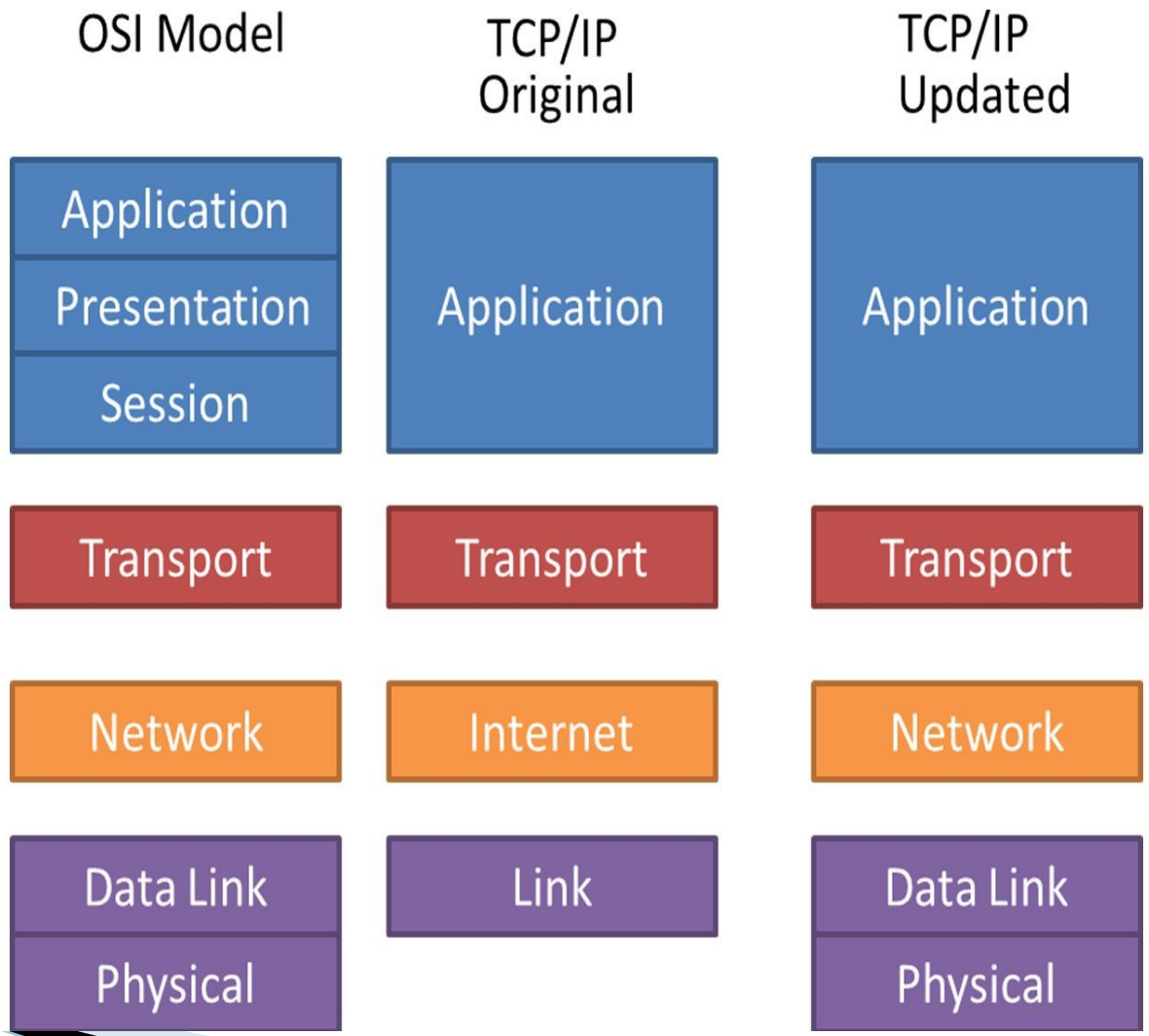
4-Layer Model-TCP/IP Architecture



TCP/IP Architecture

- **Subnetwork**
  - Implemented by the combination of hardware(network adaptor) and software(network device drivers)
  - Eg protocols: Ethernet or wireless protocols
- **Internet Protocol(IP)**
  - Supports the interconnection of multiple networking technologies into a single, logical internetwork.
- **TCP/UDP**
  - Contains two main protocols-TCP,UDP
  - TCP-provides a reliable byte-stream channel
  - UDP-provides an unreliable datagram delivery channel.
  - TCP & UDP-sometimes called end-to-end protocols/transport protocols
- **Application Layer**
  - Application Protocols-HTTP,FTP, Telnet, SMTP etc that enable the interoperation of popular applications.

OSI Vs TCP/IP


OSI Model
TCP/IP Original
TCP/IP Updated
Application
Presentation
Session
Application
Application
Transport
Transport
Transport
Network
Internet
Network
Data Link
Physical
Link
Data Link
Physical

TCP/IP and OSI

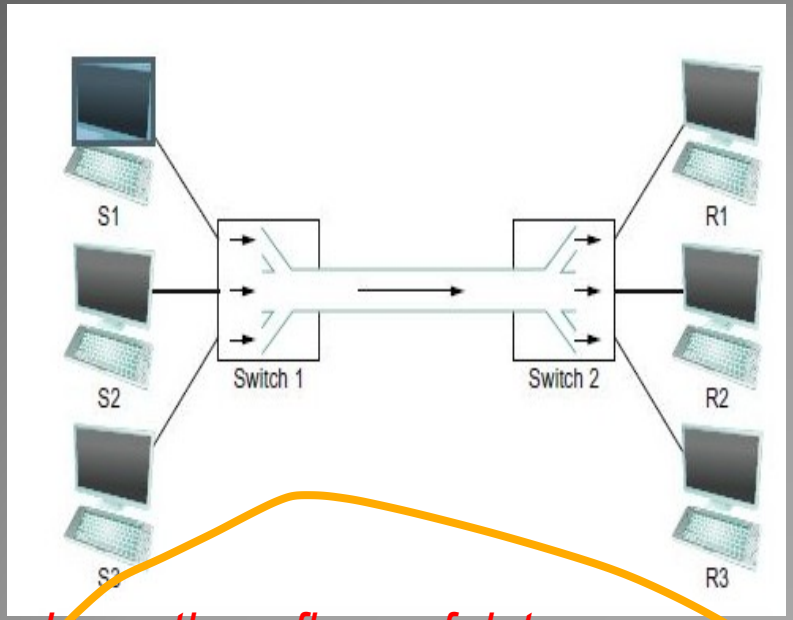
- TCP/IP architecture has come to dominate.
- There are a number of reasons for this outcome.
  - Perhaps the most important is that the key TCP/IP protocols were mature and well tested at a time when similar OSI protocols were in the development stage.
  - When businesses began to recognize the need for interoperability across networks, only TCP/IP was available and ready to go.
  - Another reason is that the OSI model is unnecessarily complex, with seven layers to accomplish what TCP/IP does with fewer layers

TCP/IP Model

Nos	Layer Name	Protocol	Data chunk	Addressing
5	Application layer	HTTP, SMTP etc	Messages	N/A
4	Transport layer	TCP/UDP	Segments	Port address
3	Network layer	IP	Datagrams	IP Address
2	Datalink layer	Ethernet, Wi-fi	Frames	MAC Address
1	Physical layer	10 base 2	Bits	N/A

How hosts share a network?

- Multiplexing
  - System resource is shared among multiple users.
  - Data being sent by multiple users can be multiplexed over the physical link that make up a network.



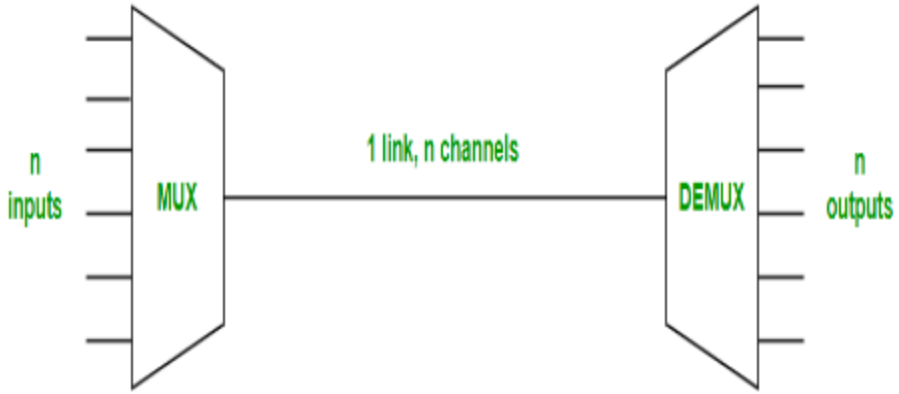
Here, three flows of data, corresponding to three hosts are multiplexed by 'switch 1' and then demultiplexed back into separate flow by 'switch 2'.

Multiplexing

Multiplexing is the sharing of a medium or bandwidth. It is the process in which multiple signals coming from multiple sources are combined and transmitted over a single communication/physical line.

Multiplexing divides the high capacity medium into low capacity logical medium which is then shared by different streams.

All mediums are capable of multiplexing.

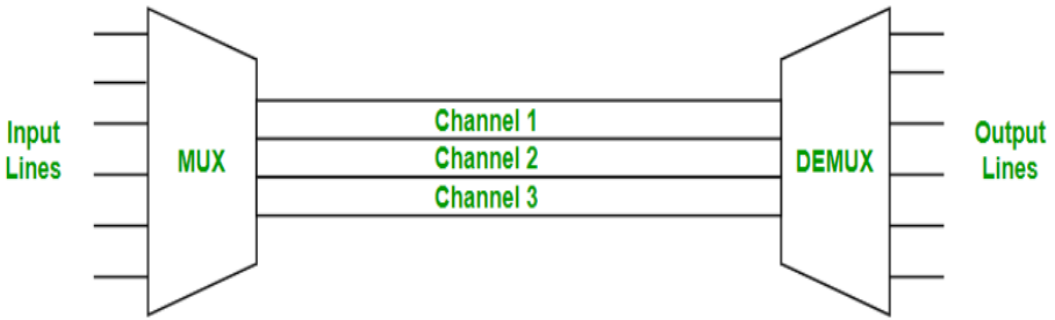


Types of Multiplexing

- ✓ Frequency Division Multiplexing (FDM)
- ✓ Time-Division Multiplexing (TDM)
- ✓ Wavelength Division Multiplexing (WDM)

Frequency Division Multiplexing

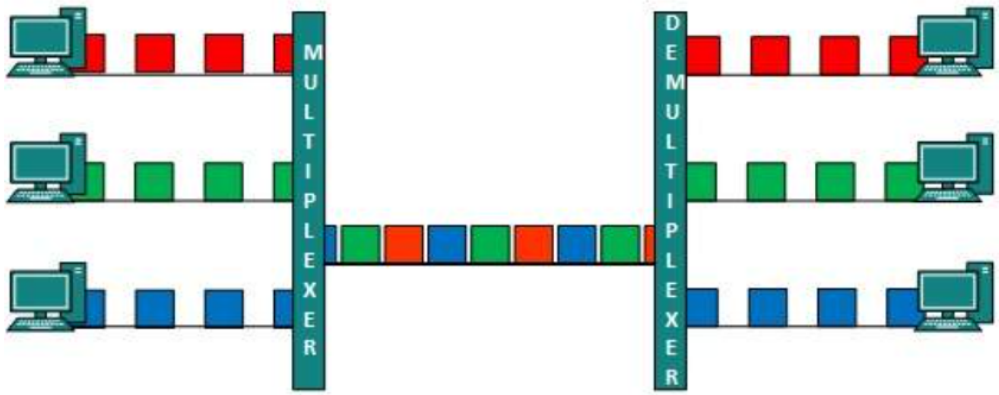
- Separation of the whole spectrum into **smaller frequency bands**
- A channel gets a certain band of the spectrum for the whole time
- Frequency Division Multiplexing is used in radio and television transmission.
- **Advantages**:
- no dynamic coordination necessary
- **Disadvantages:**
  - waste of bandwidth if the traffic is distributed unevenly
  - Inflexible
  - guard spaces



Time Division Multiplexing

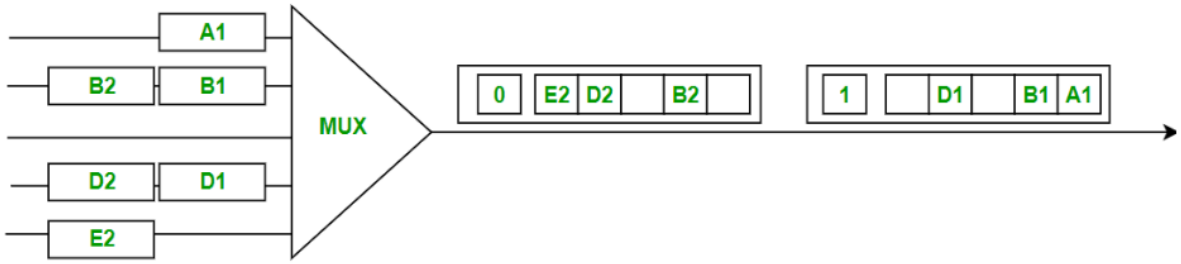
- A channel gets the whole spectrum for a certain amount of time
- A more flexible multiplexing scheme for **typical mobile**
- **communications**
- **Advantages**:
  - only one carrier in the medium at any time
  - throughput high even for many users
- **Disadvantages:**

If two transmissions overlap in time, this is called co-channel interference (precise synchronization necessary)



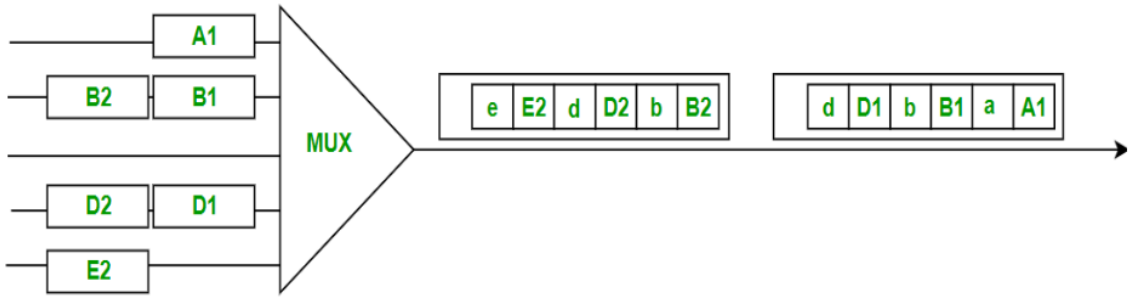
Methods of TDM

- Synchronous Time Division Multiplexing(STDM)
  - Divide time into different size quanta and in round robin fashion, give each flow a chance to send its data over the physical link.
- Limitations of STDM and FDM
  - If one of the flows(host) does not have any data to send, its share of the physical link(time quantum or frequency) remains idle even if one of the other hosts has data to transmit.



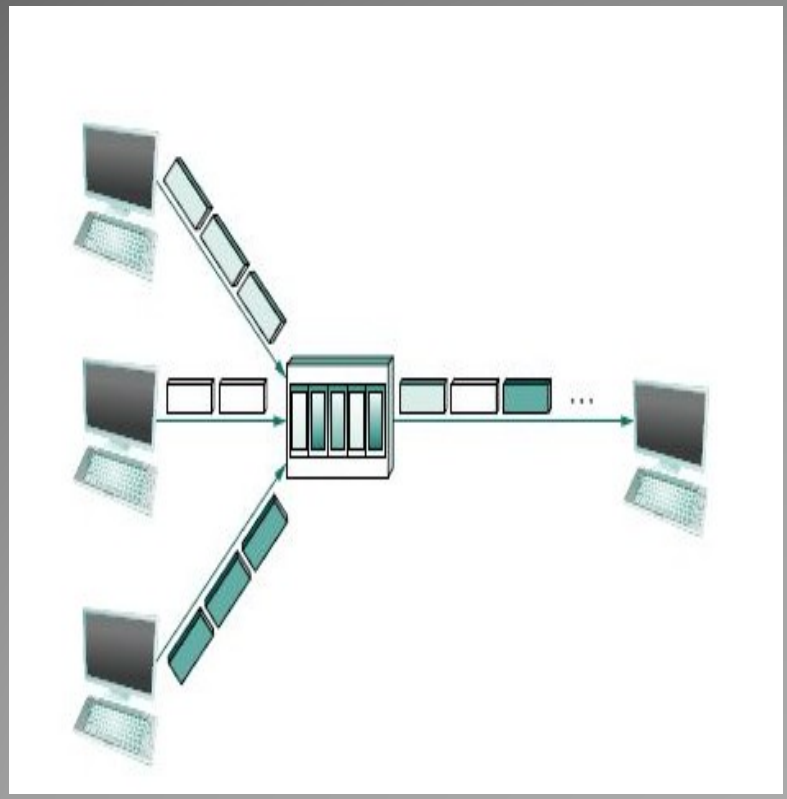
Methods of multiplexing

- Statistical Multiplexing
  - Physical link is shared over time(like STDM)
  - Data is transmitted from each flow on demand rather than using a predetermined time slot(unlike STDM)
  - Hence, the avoidance of idle time gives packet switching its efficiency.
  - To ensure that all the flows eventually get their turn to transmit over the physical link, it defines an upper bound on the size of the block of data that each flow is permitted to transmit at a given time.



Packet

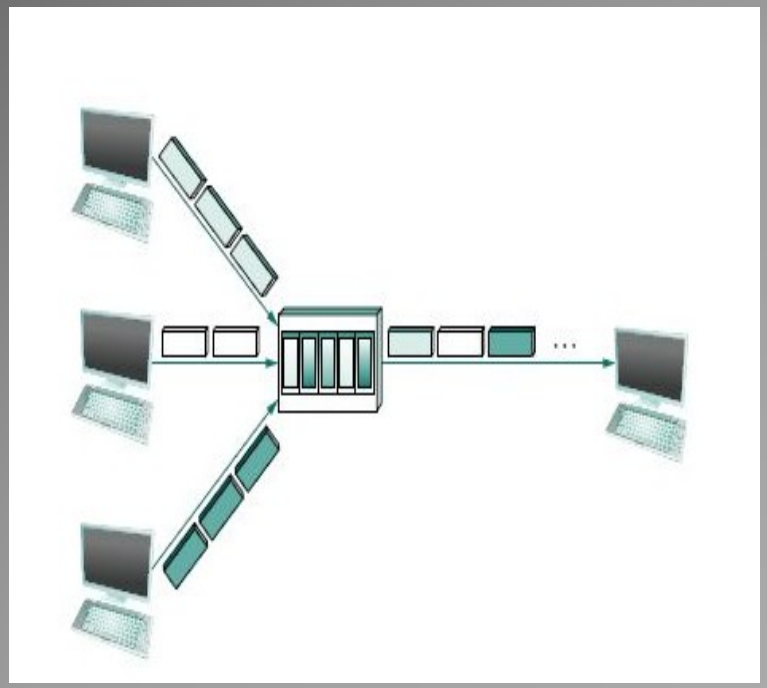
- The limited-size block of data is typically referred to as a ***packet***(unlike, large ***message*** that an application program might want to transmit.)
- The host sends a sequence of packets over the physical link.



Switch multiplexing packets!!

Which packet to send next(by a switch) on a shared link?

- Each switch in a packet-switched network makes this decision independently on a packet-by-packet basis.
- Approaches:
  - First-In-First-Out(FIFO)
  - Round-Robin
- Switch can receive packets faster than the shared link can accommodate. Hence the packets are buffed in switch's memory.
- If the switch run out of buffer space, some packets are dropped. This operating state is called ***Congested***.



Switch multiplexing packets!!

Communication modes:

Simplex

The data associated with the application flows in one direction only.

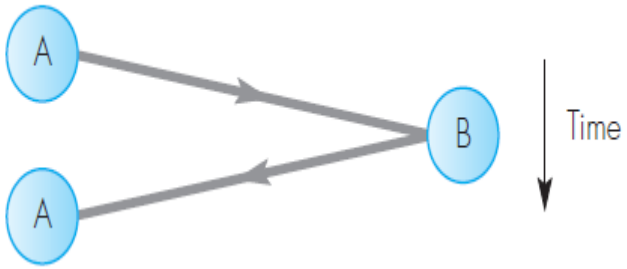
Eg: transmission of photographic images from a deep-space probe at predetermined times since this involves just a unidirectional flow of data from the probe to an earth station;

Half-Duplex

The data flows in both directions but alternately.

This mode is also known as two-way alternate

Eg: user making a request for some data from a remote server and the latter returning the requested data;



Duplex:

The data flows in both directions simultaneously.

It is also known as two-w

ay simultaneous

Eg: the two-way flow of digitized speech associated with a telephony application;

Addressing

- Address - Byte string that identifies a node.
- Network can use a node's address to distinguish it from the other nodes connected to the network.
- Types of Addressing
  - Physical
  - Logical
  - Port

Physical Address

- A link-layer address is variously called a **LAN address**, a **physical address,** or a **MAC address**
- MAC address is 6 bytes long, giving 2^{48} possible MAC addresses
- Expressed in hexadecimal notation, with each byte of the address expressed as a pair of hexadecimal numbers.
- MAC addresses were designed to be permanent- MAC address has a flat structure and doesn't change no matter where the adapter is.
  - analogous to a person's social security number.
- IEEE manages the MAC address space
- IEEE allocates the chunk of 2^{24} addresses by fixing the first 24 bits of a MAC address and letting the company create unique combinations of the last 24 bits for each adapter.
- Example MAC Address:

  49-BD-D2-C7-56-2A

Logical Address

- In the Internet, the host is identified by its **IP address**.
- an IP address is a 32-bit(four bytes) quantity that we can uniquely identifying the host.
- In an IP address, each period separates one of the bytes expressed in decimal notation from 0 to 255.
- An IP address is hierarchical because as we scan the address from left to right, we obtain more and more specific information about where the host is located in the Internet (ie, within which network, in the network of networks)
- host's IP addresses needs to be changed when the host moves, i.e, changes the network to which it is attached.
  - An IP address is analogous to a person's postal address
- Example IP Address:

  222.222.222.220

Port Number

Find a particular application/process in a node

- Message sending process must identify the receiving process running in the host because in general, a host could be running many network applications.
- This is done by Port Number
- Port Number is a 16 bit number.
- Types:
  - System Ports/Well Known Ports - 0 to 1023 (assigned by IANA)
  - User Ports/Registered Ports - 1024 to 49151 (assigned by IANA)
  - Dynamic Ports/Private or Ephemeral Ports – 49152 to 65535 (never assigned)

  Eg:
  - Web server is identified by port number 80.
  - A mail-server process is identified by port number 25.

Who assigns the IP addresses and port numbers?

Internet Assigned Numbers Authority

(IANA)

https://www.iana.org

Bits & Bytes

- 1 bit = 0 or 1
- 1 Byte = 8 bits
- 1 Nibble = 4 bits

Bits & Bytes

Name	Abbr.	Size
Kilo	K	2^10 = 1,024
Mega	M	2^20 = 1,048,576
Giga	G	2^30 = 1,073,741,824
Tera	T	2^40 = 1,099,511,627,776
Peta	P	2^50 = 1,125,899,906,842,624
Exa	E	2^60 = 1,152,921,504,606,846,976
Zetta	Z	2^70 = 1,180,591,620,717,411,303,424
Yotta	Y	2^80 = 1,208,925,819,614,629,174,706,176

Bits & Bytes

Prefix	Symbol	Multiplier	
exa	E	10^{18}	1,000,000,000,000,000,000
peta	P	10^{15}	1,000,000,000,000,000
tera	T	10^{12}	1,000,000,000,000
giga	G	10^{9}	1,000,000,000
mega	M	10^{6}	1,000,000
kilo	k	10^{3}	1,000
hecto	h	10^{2}	100
deka	da	10^{1}	10
deci	d	10^{-1}	0.1
centi	c	10^{-2}	0.01
milli	m	10^{-3}	0.001
micro	μ	10^{-6}	0.000,001
nano	n	10^{-9}	0.000,000,001
pico micro micro	p $\mu\mu$	10^{-12}	0.000,000,000,001
femto	f	10^{-15}	0.000,000,000,000,001
atto	a	10^{-18}	0.000,000,000,000,000,001

Numeric value	Description	Zeros
1	unit	0
10	ten	1
100	hundred	2
1,000	thousand	3
10,000	ten thousand	4
100,000	100 thousand	5
1,000,000	1 mil	6
10,000,000	10 mil	7
100,000,000	100 mil	8
1,000,000,000	1 bil	9
10,000,000,000	10 bil	10
100,000,000,000	100 bil	11
1,000,000,000,000	1000 bil	12

Bits & Bytes

Quantities of bytes						
Common prefix				Binary prefix		
Name	Symbol	Decimal SI	Binary JEDEC	Name	Symbol	Binary IEC
kilobyte	KB/kB	10^3	2^{10}	kibibyte	KiB	2^{10}
megabyte	MB	10^6	2^{20}	mebibyte	MiB	2^{20}
gigabyte	GB	10^9	2^{30}	gibibyte	GiB	2^{30}
terabyte	TB	10^{12}	2^{40}	tebibyte	TiB	2^{40}
petabyte	PB	10^{15}	2^{50}	pebibyte	PiB	2^{50}
exabyte	EB	10^{18}	2^{60}	exbibyte	EiB	2^{60}
zettabyte	ZB	10^{21}	2^{70}	zebibyte	ZiB	2^{70}
yottabyte	YB	10^{24}	2^{80}	yobibyte	YiB	2^{80}

- $10^3=1000$
- $2^{10}= 1024$

- $10^6=1000000$
- $2^{20}=1048576$

Network Performance

- Two measures
  - *Bandwidth/throughput*

    The bandwidth of a network is given by the number of bits that can be transmitted over the network in a certain period of time
  - *latency/delay*

    how long it takes a message to travel from one end of a network to the other

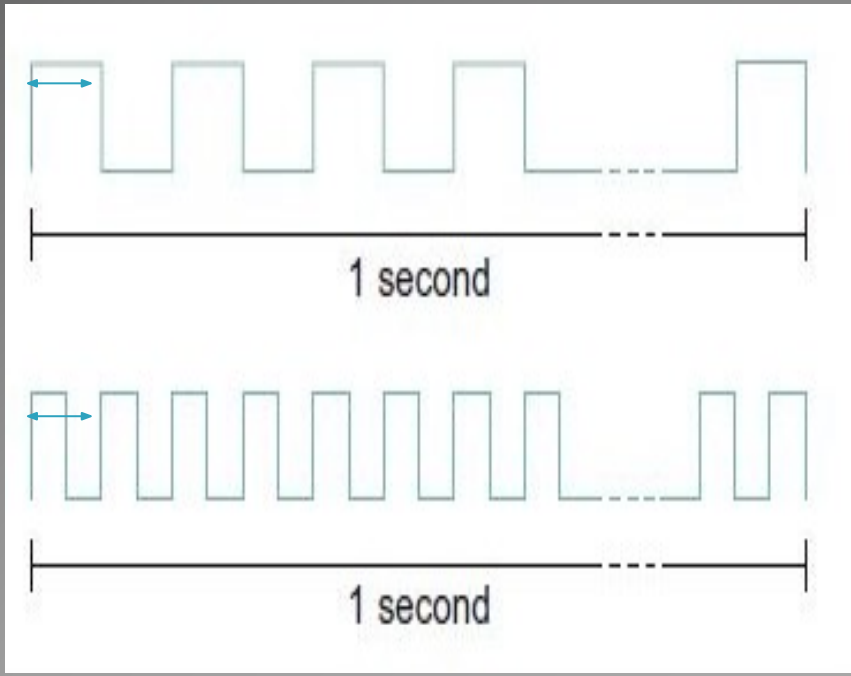
Bandwidth/Throughput

- The bandwidth of a network is given by the number of bits that can be transmitted over the network in a certain period of time
  - Bandwidth = No of bits / Time
  - Eg: Bandwidth of 10 million bits/second (Mbps), meaning that it is able to deliver 10 million bits every second.

Bandwidth/Throughput – Example

- Bandwidth in terms of how long it takes to transmit each bit of data.
  - Eg: On a 10-Mbps network, it takes 0.1 microsecond (µs) to transmit each bit.
- Egs:
  - Each bit on a 1-Mbps link is 1 µs wide
  - Each bit on a 2-Mbps link is 0.5 µs wide

*Thus, the narrower each bit can become the higher is the bandwidth!



Bandwidth Vs Throughput – How people might use them?

- Word *bandwidth* is literally a measure of the width of a frequency band. (probably refers to the range of signals that can be accommodated)
  - Eg: a voice-grade telephone line supports a frequency band ranging from 300 to 3300 Hz - it is said to have a bandwidth of 3300 Hz−300 Hz = 3000 Hz
- Word bandwidth in communication link is the number of bits per second (data rate) that can be transmitted on the link.
  - Eg: bandwidth of an Ethernet link is 10 Mbps.

Bandwidth Vs Throughput – How people might use them?

- Word *throughput* generally refers to the measured performance of a system.
- Eg: Because of various inefficiencies of implementation, a pair of nodes connected by a link with a bandwidth of 10 Mbps might achieve a throughput of only 2 Mbps. This would mean that an application on one host could send data to the other host at 2 Mbps.

Latency/Delay

- Latency is measured strictly in terms of time.
- How long it takes a message to travel from one end of a network to the other.
- Eg: a transcontinental network might have a latency of 24 milliseconds(ms)
- *round-trip time* (RTT)
  - send a message from one end of a network to the other and back.

Latency/Delay - Components

Propagation Delay (speed-of-light propagation delay)

- The time required to propagate from the beginning of the link to end is the propagation delay. The bit propagates at the propagation speed of the link.
- light travels across different media at different speeds
- Eg: It travels at
  - 3.0×10^8 m/s in a vacuum,
  - 2.3×10^8 m/s in a copper cable,
  - 2.0×10^8 m/s in an optical fiber.

Latency/Delay - Components

- Transmission Delay (amount of time taken to transmit a unit of data)
  - function of the network bandwidth and the size of the packet in which the data is carried
- Queuing Delay (Delays inside the network) packet switches generally need to store packets for some time before forwarding them on an outbound link.
- Processing Delay: is the time it takes routers to process the packet header. processing delays in high-speed routers are typically on the order of microseconds or less.

Propagation Delay Vs Transmission Delay

- The transmission delay is the amount of time required for the (router) to push out the packet.
- it is a function of the packet's length and the transmission rate of the link
- Transmit = Size / Bandwidth
- The propagation delay, on the other hand, is the time it takes a bit to propagate from one (router) to the next
- it is a function of the distance between the two (routers).
- Propagation = Distance / Speed-of-Light

Latency/Delay

- Total Latency = Propagation + Transmit + Queue
  - Propagation = Distance / Speed-of-Light
  - Transmit = Size / Bandwidth
- Distance- length of the wire over which the data will travel
- Speed-of-Light - the effective speed of light over that wire,
- Size - the size of the packet
- Bandwidth-the bandwidth at which the packet is transmitted.

*if the message contains only one bit and single link, then the Transmit and Queue terms are not relevant, and latency corresponds to the propagation delay only.

Problem!!!

- How long does it take to transmit x KB over a y-Mbps link? Give your answer as a ratio of x and y.

2^10

- x KB is 8 × 1024 × x bits.

10^6

- y Mbps is y × 10^6 bps;
- The transmission time would be
- 8 × 1024 × x/y × 10^6 sec = 8.192x/y ms.

Problem

- Consider a point-to-point link 4 km in length. At what bandwidth would propagation delay (at a speed of 2×10^8m/s) equal transmit delay for 100-byte packets? What about 512-byte packets?

Sol

1. Prop delay = Trans Delay
   - =4x 10^7 bits/sec
   - = 40 Mbps
2. What about 512-byte packets?

 = 204.8 Mbps

Problems!!!

- Consider a point-to-point link 50 km in length. At what bandwidth would propagation delay (at a speed of 2×10^8m/s) equal transmit delay for 100-byte packets? What about 512-byte packets?
- Solution
- If Size =100B, Bandwidth =3.2 Mbps

  If Size=512B, Bandwidth = 16.4 Mbps

- How "wide" is a bit on a 10-Gbps link? How long is a bit in copper wire, where the speed of propagation is 2.3×10^8 m/s?

solution

10 Gbps = 10^10 bps, meaning each bit is 10^−10 sec (0.1 ns) wide.

- The length in the wire of such a bit is

0.1 ns × 2.3 × 10^8 m/sec = 0.023 m or 23mm

Problems

- Calculate the latency (from first bit sent to last bit received) for the following:

(a) 100-Mbps Ethernet with a single store-and-forward switch in the path and a packet size of 12,000 bits. Assume that each link introduces a propagation delay of 10 μs and that the switch begins retransmitting immediately after it has finished receiving the packet.

(b) Same as (a) but with three switches.

(c) Same as (a), but assume the switch implements "cut-through" switching; it is able to begin retransmitting the packet after the first 200 bits have been received.

Problems!!!

(a) 100-Mbps Ethernet with a single store-and-forward switch in the path and a packet size of 12,000 bits. Assume that each link introduces a propagation delay of 10 μs and that the switch begins retransmitting immediately after it has finished receiving the packet.

Total Latency = 2x Tras.Delay+2x Prop. Delay
= 2x 12000/100x10^6+2x10 μs
= 260 μs

- **(b)** Same as (a) but with three switches.

Total Latency = 4x Tras.Delay+4x Prop. Delay
= 4x 12000/100x10^6+4x10 μs
= 520 μs

- **(c)** Same as (a), but assume the switch implements "cut-through" switching; it is able to begin retransmitting the packet after the first 200 bits have been received.

- Solution

  With cut-through, switch delays the packet by 200 bits= 2 μs

  Delay = 120 μs+2 μs+20 μs

  = 142 μs

Problems

- Calculate the total time required transferring a 1000KB file in the following cases, assuming an RTT of 50ms, a packet size of 1 KB data, and an initial 2×RTT of "handshaking" before data is sent:
- (a)The bandwidth is 1.5 Mbps, and data packets can be sent continuously.
- The bandwidth is 1.5 Mbps, but after we finish sending each data packet we must wait one RTT before sending the next.
- (c) The bandwidth is "infinite," meaning that we take transmit time to be zero, and up to 20 packets can be sent per RTT.
- (d) The bandwidth is infinite, and during the first RTT we can send one packet (2 ^1−1), during the second RTT we can send two packets (2 ^2−1), during the third we can send four (2 ^3−1), and so on.

Solutions

(a)The bandwidth is 1.5 Mbps, and data packets can be sent continuously.

- Total time = 2* RTT + Trans. Time + Pro.delay
- Total time =5.586 sec

(b) The bandwidth is 1.5 Mbps, but after we finish sending each data packet we must wait one RTT before sending the next.

To the above we add the time for 999 RTTs (the number of RTTs between when packet 1 arrives and packet 1000 arrives), for a total of 5.586 + 49.95 = 55.536.

Solutions

(c)The bandwidth is "infinite," meaning that we take transmit time to be zero, and up to 20 packets can be sent per RTT.

This is 49.5 RTTs, plus the initial 2, for 2.575 seconds.

(d)The bandwidth is infinite, and during the first RTT we can send one packet (2 ^1−1), during the second RTT we can send two packets (2 ^2−1), during the third we can send four (2 ^3−1), and so on.

Right after the handshaking is done we send one packet. One RTT after the handshaking we send two packets.

At n RTTs past the initial handshaking we have sent $1 + 2 + 4 + \cdots + 2^n = 2^{(n+1)} - 1$ packets.

At n = 9 we have thus been able to send all 1,000 packets; the last batch arrives 0.5 RTT later.

Total time is 2+9.5 RTTs, or .575 sec.

Performance Characteristics

- Bandwidth and latency combine to define the performance characteristics of a given link or channel.

- Their relative importance, depends on the application.
- For some applications, latency dominates bandwidth and vice versa.

Latency Dominates!

- Consider a client that sends a 1-byte message to a server and receives a 1-byte message in return.
  - Assume that no serious computation is involved in preparing the response.
  - The application will perform much differently on a transcontinental channel with a 100-ms RTT than it will on an across-the-room channel with a 1-ms RTT.
  - Whether the channel is 1 Mbps or 100 Mbps is relatively insignificant, however, since the former implies that the time to transmit a byte (Transmit) is 8 μs and the latter implies Transmit = 0.08 μs.

Bandwidth Dominates!

- Consider a digital library program that is being asked to fetch a 25-megabyte (MB) image.
  the more bandwidth that is available, the faster it will be able to return the image to the user.
- Here, the bandwidth of the channel dominates performance.
- Suppose that the channel has a bandwidth of 10 Mbps. It will take 20 seconds to transmit the image ($25\times10^6 \times 8$ bits $\div 10\times10^6$ Mbps = 20 seconds), making it relatively unimportant if the image is on the other side of a 1-ms channel or a 100-ms channel;
- The difference between a 20.001-second response time and a 20.1-second response time is negligible.

Pause!!!

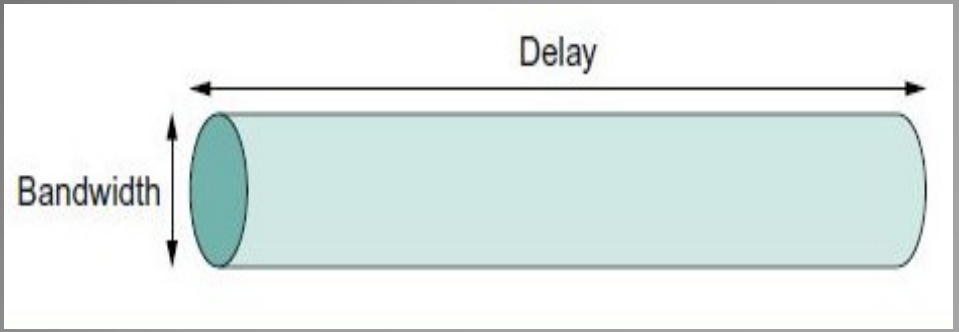
- For each of the following operations on a remote file server, discuss whether they are more likely to be delay sensitive or bandwidth sensitive:
  (a) Open a file.
  (b) Read the contents of a file.
  (c) List the contents of a directory.
  (d) Display the attributes of a file.

(a) Delay-sensitive; the messages exchanged are short.
(b)Bandwidth-sensitive, particularly for large files.
that the underlying protocol uses a large message size or window size.

- (c) **Delay-sensitive;** directories are typically of modest size.
- (d) **Delay-sensitive;** a file's attributes are typically much smaller than the file itself.

Delay X Bandwidth Product

- Consider a channel between a pair of processes as a hollow pipe where,
  - latency - the length of the pipe
  - bandwidth - the diameter of the pipe,
- Then the delay × bandwidth product gives the volume of the pipe.
- It is the maximum number of bits that could be in transit through the pipe at any given instant.



Delay X Bandwidth Product

Example

- A transcontinental channel with a one-way latency of 50 ms and a bandwidth of 45 Mbps is able to hold,

  $50 \times 10^{-3}\text{s} \times 45 \times 10^{6}\text{bits/s}$

  $= 2.25 \times 10^{6}\text{bits}$

  or approximately 280 KB of data.

- Usually, whether the “delay” in “delay × bandwidth” means one-way latency or RTT is made clear by the context

Delay X Bandwidth Product

- The delay × bandwidth product is important to know when constructing high-performance networks because
  - it corresponds to how many bits the sender must transmit before the first bit arrives at the receiver.
  - If the sender is expecting the receiver to somehow signal that bits are starting to arrive, and it takes another channel latency for this signal to propagate back to the sender,
  - then the sender can send up one RTT× bandwidth worth of data before hearing from the receiver that all is well.
  - The bits in the pipe are said to be “in flight,” which means that if the receiver tells the sender to stop transmitting it might receive up to one RTT×bandwidth’s worth of data before the sender manages to respond.

Sample Delay X Bandwidth Products

Link type	Bandwidth (typical)	One-way distance (typical)	Round-trip delay	RTT × Bandwidth
Dial-up	56 kbps	10 km	87 μs	5 bits
Wireless LAN	54 Mbps	50 m	0.33 μs	18 bits
Satellite	45 Mbps	35,000 km	230 ms	10 Mb
Cross-country fiber	10 Gbps	4,000 km	40 ms	400 Mb

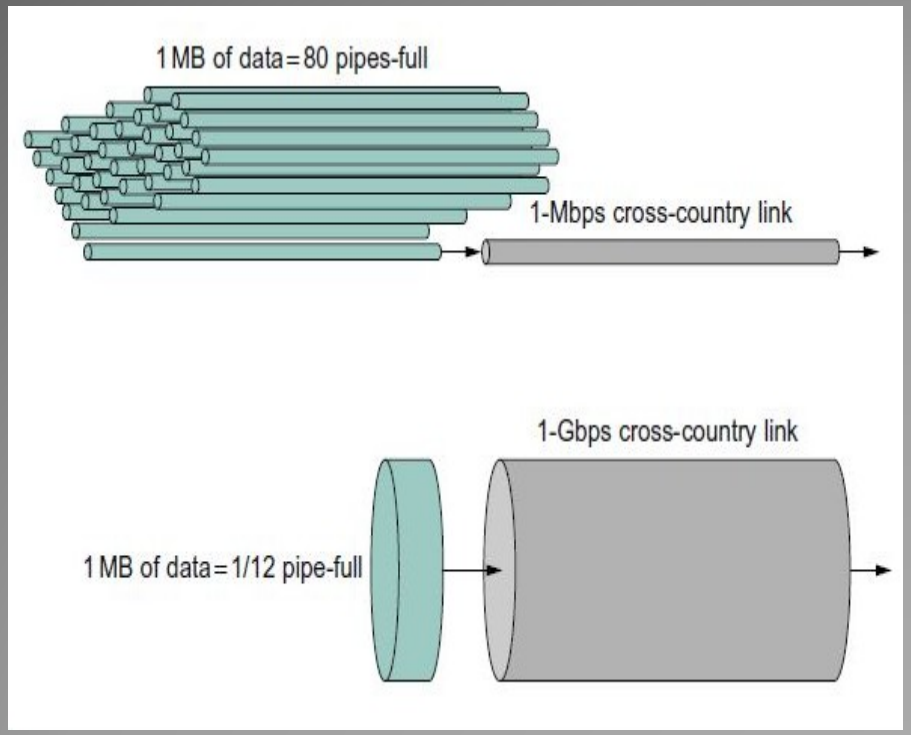
Significance of ever-increasing bandwidth

- Consider transmit of a 1-MB file with RTT of 100 ms
  - If Bandwidth = 1-Mbps
  - If Bandwidth = 1-Gbps

- If Bandwidth = 1-Mbps, it takes 80 round-trip times to transmit the file; during each RTT, 1.25% of the file is sent.
- If Bandwidth = 1-Gbps 1-MB file doesn't even come close to filling 1 RTT's worth of the 1-Gbps link, which has a delay×bandwidth product of 12.5 MB

Significance of ever-increasing bandwidth

- the 1-MB file looks like a stream of data that needs to be transmitted across a 1-Mbps network,
- while it looks like a single packet on a 1-Gbps network.



Problems

- Suppose a 1-Gbps point-to-point link is being set up between the Earth and a new lunar colony. The distance from the moon to the Earth is approximately 385,000 km, and data travels over the link at the speed of light 3 × 10^8 m/s.
- (a) Calculate the minimum RTT for the link.
- (b) Using the RTT as the delay, calculate the delay × bandwidth product for the link.
- What is the significance of the delay × bandwidth product computed in (b)?
- (d) A camera on the lunar base takes pictures of the Earth and saves them in digital format to disk. Suppose Mission Control on Earth wishes to download the most current image, which is 25 MB. What is the minimum amount of time that will elapse between when the request for the data goes out and the transfer is finished?

Solutions

(a) minimum RTT for the link.

The minimum RTT = 2x propagation delay

$2 \times 385,000,000 \text{ m} / 3 \times 10^8 \text{ m/s}$

$= 2.57$ seconds.

(b) The delay×bandwidth product is 2.57 s×1 Gbps

$= 2.57\text{Gb} = 321$ MB.

(c)This represents the amount of data the sender can send before it would be possible to receive a response.

Solutions

(d) A camera on the lunar base takes pictures of the Earth and saves them in digital format to disk. Suppose Mission Control on Earth wishes to download the most current image, which is 25 MB. What is the minimum amount of time that will elapse between when the request for the data goes out and the transfer is finished?

We require at least one RTT from sending the request before the first bit of the picture could begin arriving at the ground (TCP would take longer). 25 MB is 200Mb.

Assuming bandwidth delay only, it would then take 200Mb/ 1000Mbps = 0.2 seconds to finish sending, for a total time of 0.2+ 2.57 = 2.77 sec until the last picture bit arrives on earth.

5-Layer Model-TCP/IP Architecture

TCP/IP Architecture

- **Application Layer**
  - The application layer is where network applications and their application-layer protocols reside.
  - An application-layer protocol is distributed over multiple end systems, with the application in one end system using the protocol to exchange packets of information with the application in another end system.
  - packet of information at the application layer is **message**.
  - Egs: The Internet's application layer includes many protocols:
    - HTTP protocol - which provides for Web document request and transfer,
    - SMTP - which provides for the transfer of e-mail messages,
    - FTP - which provides for the transfer of files between two end systems

TCP/IP Architecture

- **Transport Layer**
- Concerned with process-to-process delivery of information.
  - The Internet's transport layer transports application-layer messages between application endpoints.
  - Transport-layer packet called as a **segment**.
  - Eg: In the Internet there are two transport protocols, TCP and UDP, either of which can transport application-layer messages.
    - TCP provides a connection-oriented service to its applications.
    - This service includes guaranteed delivery of application-layer messages to the destination and flow control and congestion-control mechanism.
    - The UDP protocol provides a connectionless service to its applications.
    - This is a no-frills service that provides no reliability, no flow control, and no congestion control.

TCP/IP Architecture

- **Network Layer**
  - The Internet's network layer is responsible for moving network-layer packets known as **datagrams** from one host to another
  - The network layer then provides the service of delivering the datagram to the transport layer in the destination host
  - The Internet's network layer includes the celebrated **IP Protocol**, which defines the fields in the datagram as well as how the end systems and routers act on these fields.
  - There is only one IP protocol, and all Internet components that have a network layer must run the IP protocol.
  - The Internet's network layer also contains routing protocols that determine the routes that datagrams take between sources and destinations.

TCP/IP Architecture

Data Link Layer

- Organizes the bit stream into a data unit called a "**frame**" and delivers the frame to an adjacent system.
- The services provided by the link layer depend on the specific link-layer protocol that is employed over the link.
- For example, some link-layer protocols provide reliable delivery, from transmitting node, over one link, to receiving node.
- Datagrams typically need to traverse several links to travel from source to destination, a datagram may be handled by different link-layer protocols at different links along its route.
- Eg: Ethernet, WiFi, and the cable access network's DOCSIS(Data Over Cable Service Interface Specification) protocol.

TCP/IP Architecture

- **Physical Layer**
  - The functions needed to carry the **bit stream** over a physical medium to another system.
  - The job of the physical layer is to move the ***individual bits*** from one node to the next
  - The protocols in this layer are link dependent and further depend on the actual transmission medium of the link (for example, twisted-pair copper wire, fiber optics).
  - Eg: Ethernet has many physical-layer protocols: one for twisted-pair copper wire, another for coaxial cable, another for fiber, and so on. In each case, a bit is moved across the link in a different way.