ADVANCED COMPUTER NETWORKS

20IMCAT404

MODULE I

Syllabus
Introduction to Networks

- Basic communications model - Protocol layers and service models
- Basic definitions - OSI model - Internet protocols - Role of standards organizations
- Security in the internet - Concept of quality of service (QoS)
- Application layer protocols - Client-server as a key mode
- Network application architecture – Web – HTTP – FTP – SMTP - POP3 - DNS

What is a Computer Network?

Computer Network

- A computer network is a system in which **multiple computers are connected to each other to share information and resources.**

- Networked computing devices exchange data with each other using a data link.

- The *connections* between nodes are established using either **cable media or wireless media.**

Characteristics of a Computer Network

- **Share Resources** from one computer to another
- Create files and store them in one computer, **access those files from the other computer(s)** connected over the network
- **Connect a printer, scanner, or a fax machin**e to one computer within the network and let other computers of the network use the machines available over network.

Basic communications model

- Communication is **the exchange of messages between a sender and a receiver via a medium**. This is the basic communication model.
- The receiver's response or feedback is part of the model and consideration of the context(s) in which communication takes.
- The figure below shows a simple and most basic communication model.

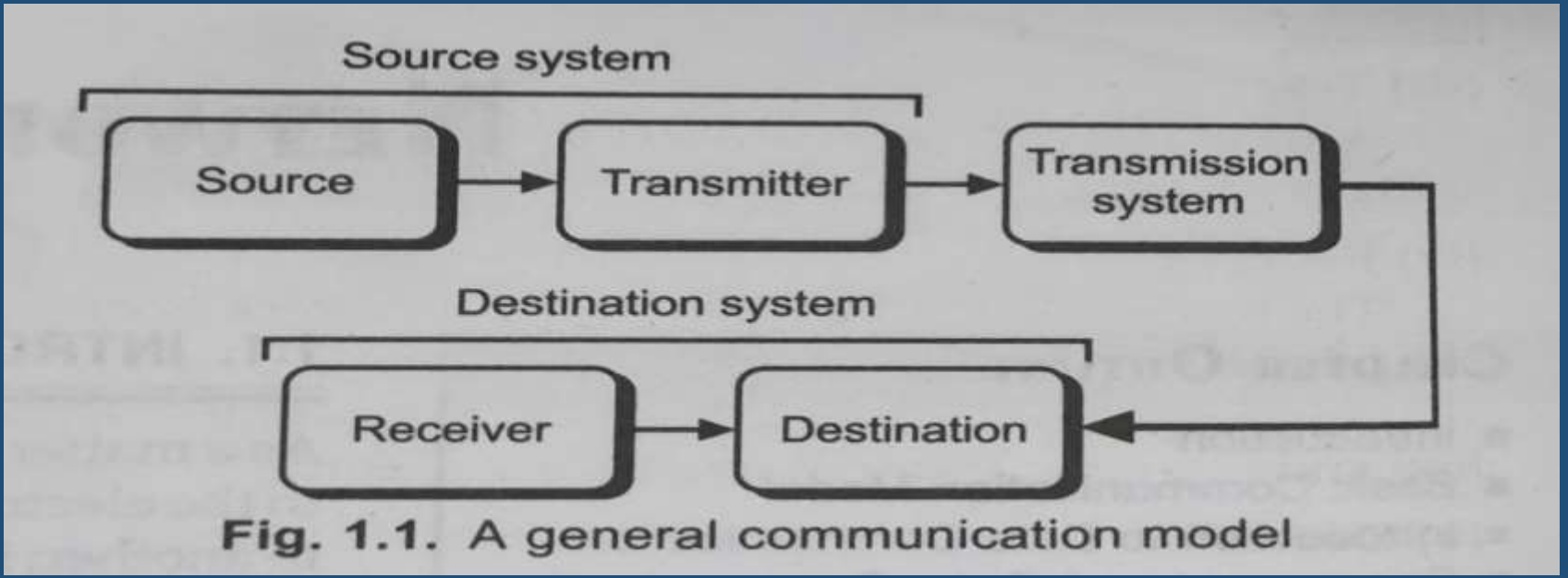


Fig. 1.1. A general communication model

Basic communications model

- The **key elements** of basic communication model are:
  - **i.** **Source:** The device which generates the data to be transmitted. *Eg: personal computers.*
  - **ii.** **Transmitter**: A transmitter transforms and encodes the information in such a manner so as to produce electromagnetic waves or signals. These electromagnetic signals can be transmitted across some sort of transmission systems. *Eg: A modem takes a digital bit stream from computer and transforms into analog signal which can be handled by the telephone network.*
  - **iii.** **Transmission System:** A transmission system can be a single transmission line or a complex network connecting source and destination.
  - **iv.** **Receiver:** Accepts the signal from the transmission system and converts it into a form which can be handled by the destination device. Eg: Modem
  - **v.** **Destination:** The receiver passes information to the destination for processing. A destination takes the incoming data from the receiver.

Basic communications model

- End systems are connected together by a network of **communication links and packet switches (**Packet switches in today's Internet are **routers and link-layer switches).**

- There are many types of communication links, which are made up of different types of physical media, including coaxial cable, copper wire, optical fiber etc.

- Different links can transmit data at different rates, with the **transmission rate of a link measured in bits/second.**

- When one end system has data to send to another end system, the sending end system segments the data and adds header bytes to each segment.
- The resulting packages of information, known as **packets, are then** sent through the network to the destination end system, where they are reassembled into the original data.

Basic communications model

- **Packet-switched networks (which transport packets) are in many ways similar to transportation networks of highways, roads, and intersections (which transport vehicles).**
- For example, a factory that needs to **move a large amount of cargo to some destination** warehouse located thousands of kilometers away.
  - At the factory, the cargo is segmented and loaded into a fleet of trucks. Each of the trucks then independently travels through the network of highways, roads, and intersections to the destination warehouse.
  - At the destination warehouse, the cargo is unloaded and grouped with the rest of the cargo arriving from the same shipment.
- Thus, in many ways, **packets are analogous to trucks, communication links are analogous to highways and roads, packet switches are analogous to intersections, and end systems are analogous to buildings. Just as a truck takes a path through the transportation network, a packet takes a path through a computer network.**

What is a Protocol?

Network Protocol

- *A **protocol defines the format and the order of messages exchanged between** two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other event.*

- A network protocol is an established set of rules that determine how data is transmitted between different devices in the same network.

- It allows connected devices to communicate with each other, regardless of any differences in their internal processes, structure or design.

Network Protocol

- All activity in the Internet that involves two or more communicating remote entities is **governed by a protocol**.

- For example, **Hardware-Implemented Protocols** in **two physically connected computers control the flow of bits on the "wire" between the two network interface cards;**

- **Congestion-Control Protocols in end systems control the rate at which packets are transmitted between sender and receiver;**

- **Protocols in routers determine a packet's path from source to destination**.

Network Protocol – Protocol Layering

- A **protocol** is a set of rules and standards that primarily outline a language that devices will use to communicate. There are an excellent range of protocols in use extensively in networking, and that they are usually implemented in numerous layers.
- It provides a communication service where the process is used to exchange the messages. When the communication is simple, we can use only one simple protocol.
- **When the communication is complex, we must divide the task between different layers, so, we need to follow a protocol at each layer, this technique we used to call PROTOCOL LAYERING. This layering allows us to separate the services from the implementation.**
- ***Each layer needs to receive a set of services from the lower layer and to give the services to the upper layer. The modification done in any one layer will not affect the other layers.***

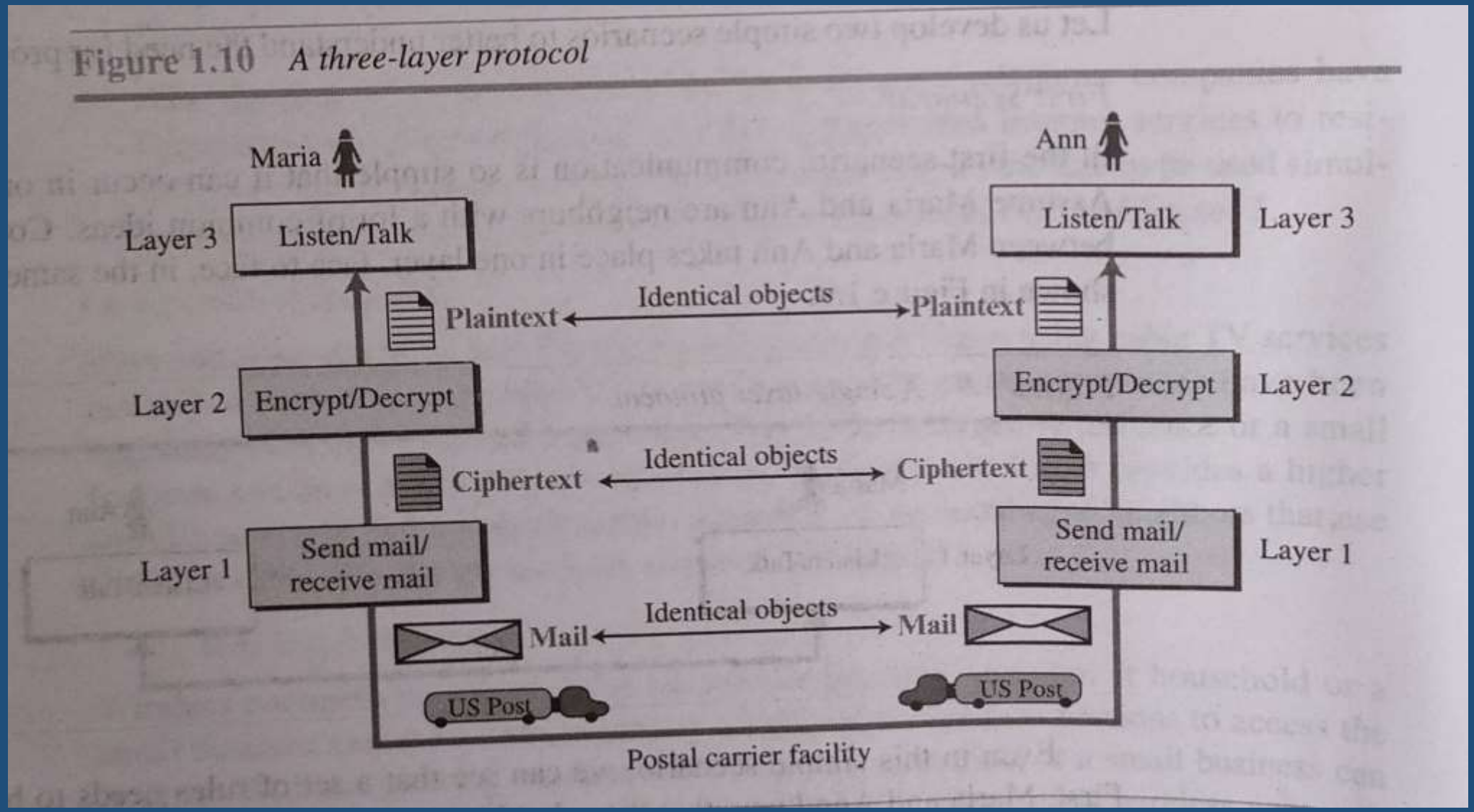
Basic Elements of Layered Architecture

- The **basic elements** of the layered architecture are as follows –
  - **Service** – Set of actions or services provided from one layer to the higher layer.
  - **Protocol** – It defines a set of rules where a layer uses to exchange the information with its peer entity. It is concerned about both the contents and order of the messages used.
  - **Interface** – It is a way through that the message is transferred from one layer to another layer.

Basic Elements of Layered Architecture

- Reasons
  - The reasons for using layered protocols are explained below –
    - Layering of protocols provides well-defined interfaces between the layers, so that ***a change in one layer does not affect an adjacent layer - Modularity***.
    - The protocols of a network are extremely complicated and designing them in layers makes their implementation more feasible.

Basic Elements of Layered Architecture



Protocol Layering

- Network protocol layering is a system of service hierarchy used in networked computer communication.

- Each layer offers a set of guaranteed services to the layer above such that higher-level abstractions can be built while making assumptions about lower-level transport services.

Principles of Protocol Layering

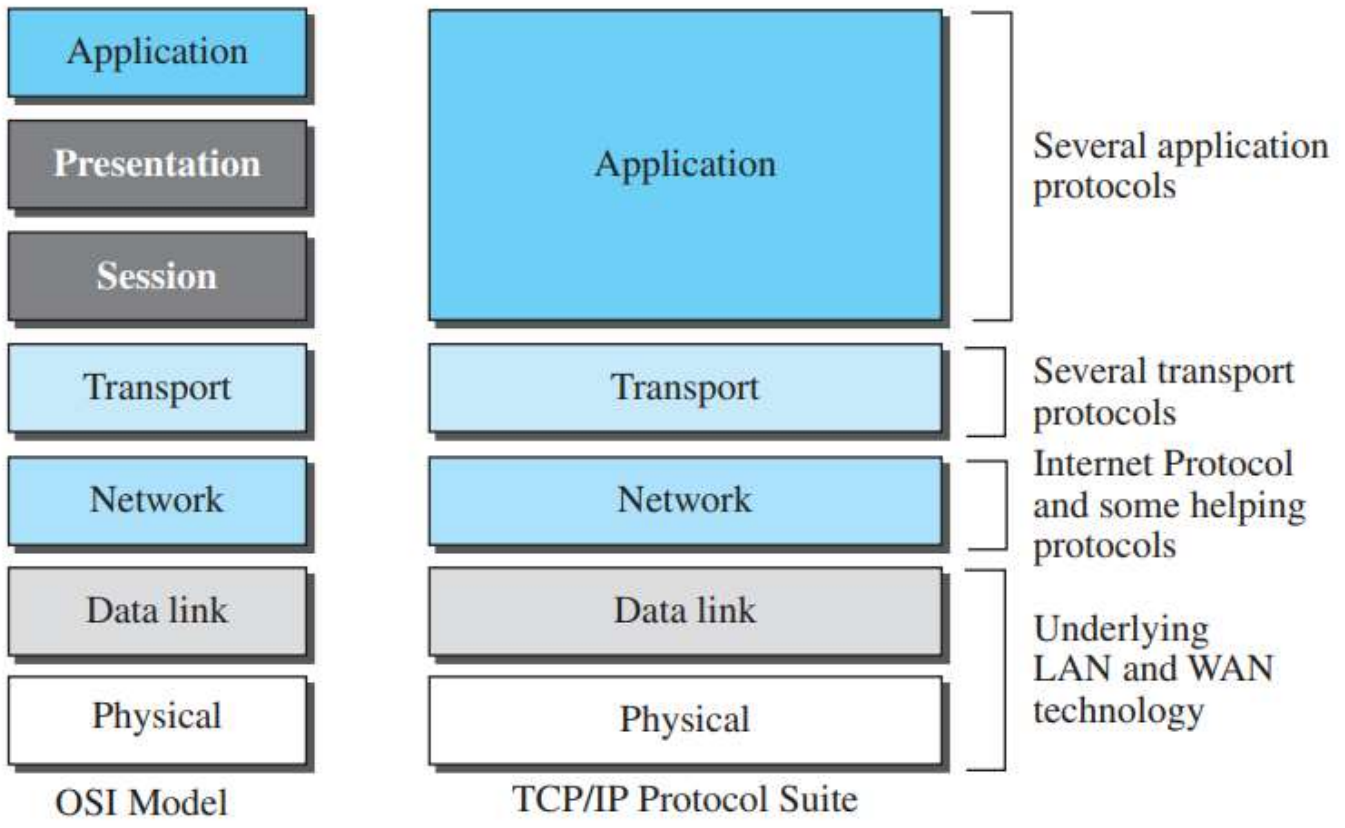
1. Since bidirectional communication is needed, need to make each layer capable to perform two opposite tasks. Eg: able to decrypt/encrypt, send/receive

2. The two objects under each layer at both sites should be identical.

Different Layer Models

- The two most popular network layer models as the OSI Stack and TCP/IP Stack.
- These two models are very similar but have some differences.

Figure 1.20 *TCP/IP and OSI model*



The OSI Model

- In the late 1970s, the International Organization for Standardization (ISO) proposed that computer networks be organized around seven layers, called the Open Systems Interconnection (OSI) model.

- The ISO was one of the first organizations to formally define a common way to connect computers. Their architecture, called the *Open Systems Interconnection (OSI) architecture* defines a partitioning of network functionality into seven layers, where one or more protocols implement the functionality assigned to a given layer.

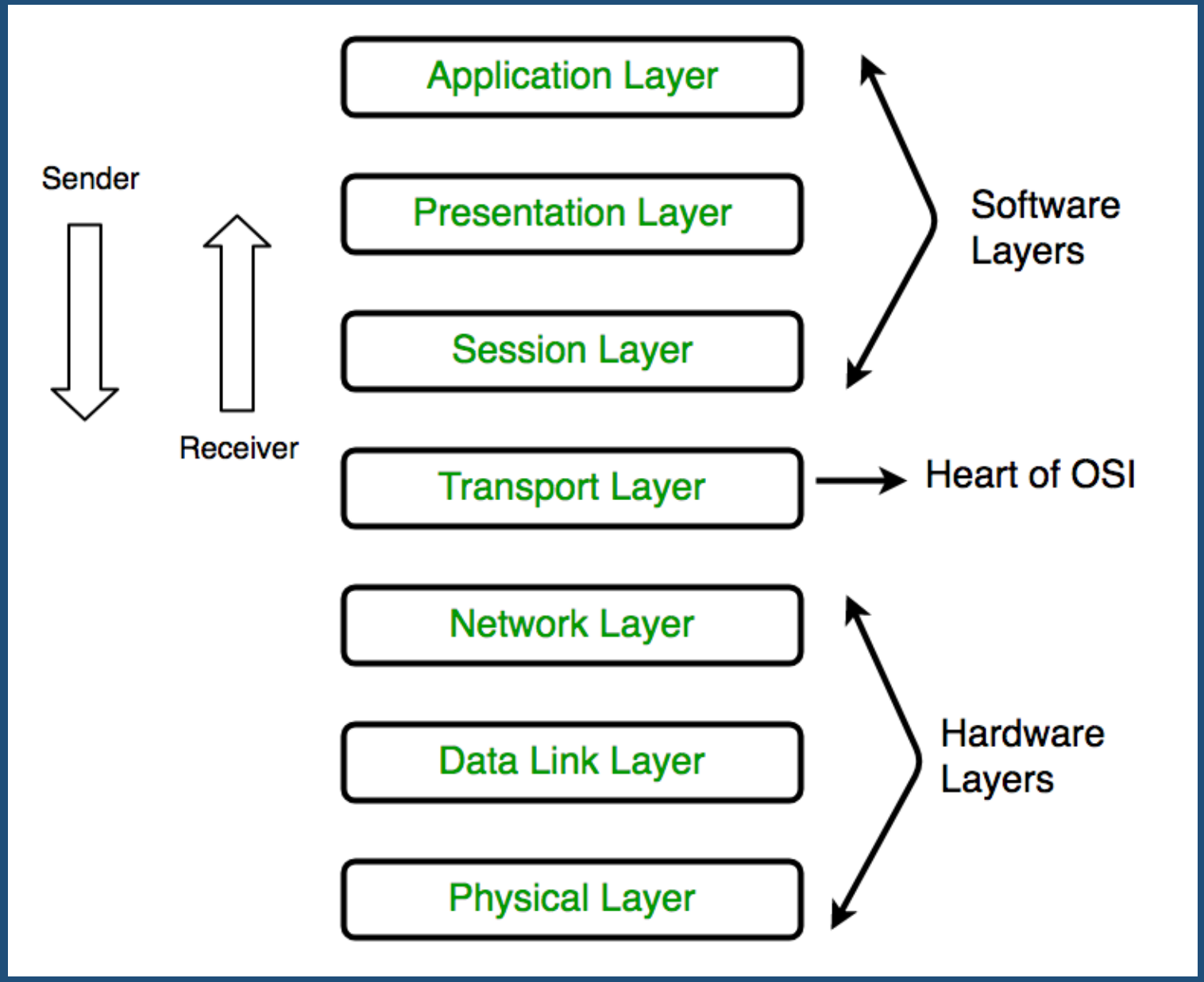
The OSI Model

- OSI stands for **Open Systems Interconnection**.
- It has been developed by ISO – '**International Organization for Standardization**', in the year 1984.
- It is a 7 layer architecture with each layer having specific functionality to perform.
- All these 7 layers work collaboratively to transmit the data from one person to another across the globe.

The OSI Model

- The seven layers of the OSI reference model are:
    - Application layer,
    - Presentation layer,
    - Session layer,
    - Transport layer,
    - Network layer,
    - Data link layer,
    - Physical layer.

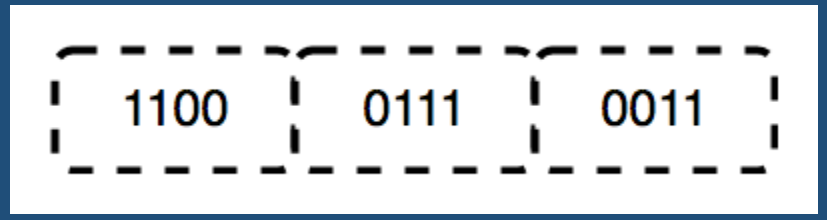
The OSI Model



The OSI Model

1. Physical Layer (Layer 1) :

- The lowest layer of the OSI reference model is the physical layer.
- It is responsible for the actual physical connection between the devices.
- The physical layer contains information in the form of **bits.**
- It is responsible for transmitting individual bits from one node to the next.
- When receiving data, this layer will get the signal received and convert it into 0s and 1s and send them to the Data Link layer, which will put the frame back together.



Hub, Repeater, Modem, Cables are Physical Layer devices.

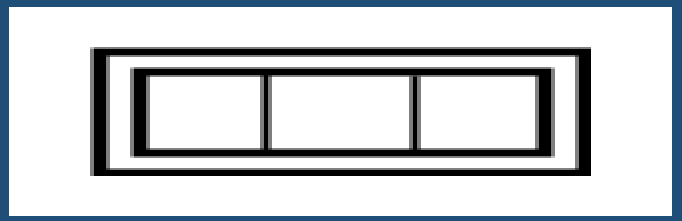
The OSI Model

- The functions of the physical layer are as follows:
  - **Bit synchronization:** The physical layer provides the synchronization of the bits by providing a clock. This clock controls both sender and receiver thus providing synchronization at bit level.
  - **Bit rate control:** The Physical layer also defines the transmission rate i.e. the number of bits sent per second.
  - **Physical topologies:** Physical layer specifies the way in which the different, devices/nodes are arranged in a network i.e. bus, star, or mesh topology.
  - **Transmission mode:** Physical layer also defines the way in which the data flows between the two connected devices. The various transmission modes possible are Simplex, half-duplex and full-duplex.

The OSI Model

2. Data Link Layer (DLL) (Layer 2) :

- The data link layer is responsible for the node-to-node delivery of the message.
- The main function of this layer is to make sure data transfer is error-free from one node to another, over the physical layer.
- When a packet arrives in a network, it is the responsibility of DLL to transmit it to the Host using its MAC address.
- Data Link Layer is divided into two sublayers:
    - Logical Link Control (LLC)
    - Media Access Control (MAC)



- The packet received from the Network layer is further divided into frames depending on the frame size of NIC(Network Interface Card).
- *Packet in Data Link layer is referred to as* **Frame.**
- ** Data Link layer is handled by the NIC (Network Interface Card) and device drivers of host machines. *** **Switch & Bridge** are Data Link Layer devices.

The OSI Model

- The functions of the Data Link layer are :
  - **Framing:** Framing is a function of the data link layer. It provides a way for a sender to transmit a set of bits that are meaningful to the receiver. This can be accomplished by attaching special bit patterns to the beginning and end of the frame.
  - **Physical addressing:** After creating frames, the Data link layer adds physical addresses (MAC address) of the sender and/or receiver in the header of each frame.
  - **Error control:** Data link layer provides the mechanism of error control in which it detects and retransmits damaged or lost frames.
  - **Flow Control:** The data rate must be constant on both sides else the data may get corrupted thus, flow control coordinates the amount of data that can be sent before receiving acknowledgement.
  - **Access control:** When a single communication channel is shared by multiple devices, the MAC sub-layer of the data link layer helps to determine which device has control over the channel at a given time.

The OSI Model

3. Network Layer (Layer 3) :

- The network layer works for the **transmission of data from one host to the other located in different networks.**
- It also takes care of **packet routing** i.e. selection of the shortest path to transmit the packet, from the number of routes available.
- The **sender & receiver's IP addresses** are placed in the header by the network layer.
- The functions of the Network layer are :
  - **Routing:** The network layer protocols determine which route is suitable from source to destination. This function of the network layer is known as routing.
  - **Logical Addressing:** In order to identify each device on internetwork uniquely, the network layer defines an addressing scheme. The sender & receiver's IP addresses are placed in the header by the network layer. Such an address distinguishes each device uniquely and universally.
- *Segment* in Network layer is referred to as **Packet**.
- Network layer is implemented by networking devices such as routers.

The OSI Model

4. Transport Layer (Layer 4) :

- The transport layer provides services to the application layer and takes services from the network layer.
- The data in the transport layer is referred to as *Segments*.
- It is responsible for the *End to End Delivery* of the complete message.
- The transport layer also provides the a*cknowledgement* of the successful data transmission and re-transmits the data if an error is found.
- **At sender's side:** Transport layer receives the formatted data from the upper layers, performs *Segmentation,* and also implements *Flow & Error control* to ensure proper data transmission. It also adds Source and Destination port numbers in its header and forwards the segmented data to the Network Layer.
- **At receiver's side:** Transport Layer reads the port number from its header and forwards the Data which it has received to the respective application. It also performs sequencing and reassembling of the segmented data.

The OSI Model

- The functions of the transport layer are as follows:

  - **Segmentation and Reassembly:** This layer accepts the message from the (session) layer, and breaks the message into smaller units. Each of the segments produced has a header associated with it. The transport layer at the destination station reassembles the message.

  - **Service Point Addressing:** In order to deliver the message to the correct process, the transport layer header includes a type of address called service point address or port address. Thus by specifying this address, the transport layer makes sure that the message is delivered to the correct process.

    - *Data in the Transport Layer is called as **Segments**.*
    - *Transport Layer is called as **Heart of OSI** model.*

The OSI Model

5. Session Layer (Layer 5) :

- This layer is responsible for the establishment of connection, maintenance of sessions, authentication, and also ensures security.

- The functions of the session layer are :
  - **Session establishment, maintenance, and termination:** The layer allows the two processes to establish, use and terminate a connection.
  - **Synchronization:** This layer allows a process to add checkpoints which are considered synchronization points into the data. These synchronization points help to identify the error so that the data is re-synchronized properly, and ends of the messages are not cut prematurely and data loss is avoided.
  - **Dialog Controller:** The session layer allows two systems to start communication with each other in half-duplex or full-duplex.

The OSI Model

6. Presentation Layer (Layer 6):

- The presentation layer is also called the **Translation layer**. The data from the application layer is extracted here and manipulated as per the required format to transmit over the network.

- The functions of the presentation layer are :
  - **Translation:** For example, ASCII to EBCDIC.
  - **Encryption/ Decryption:** Data encryption translates the data into another form or code. The encrypted data is known as the ciphertext and the decrypted data is known as plain text. A key value is used for encrypting as well as decrypting data.
  - **Compression:** Reduces the number of bits that need to be transmitted on the network.

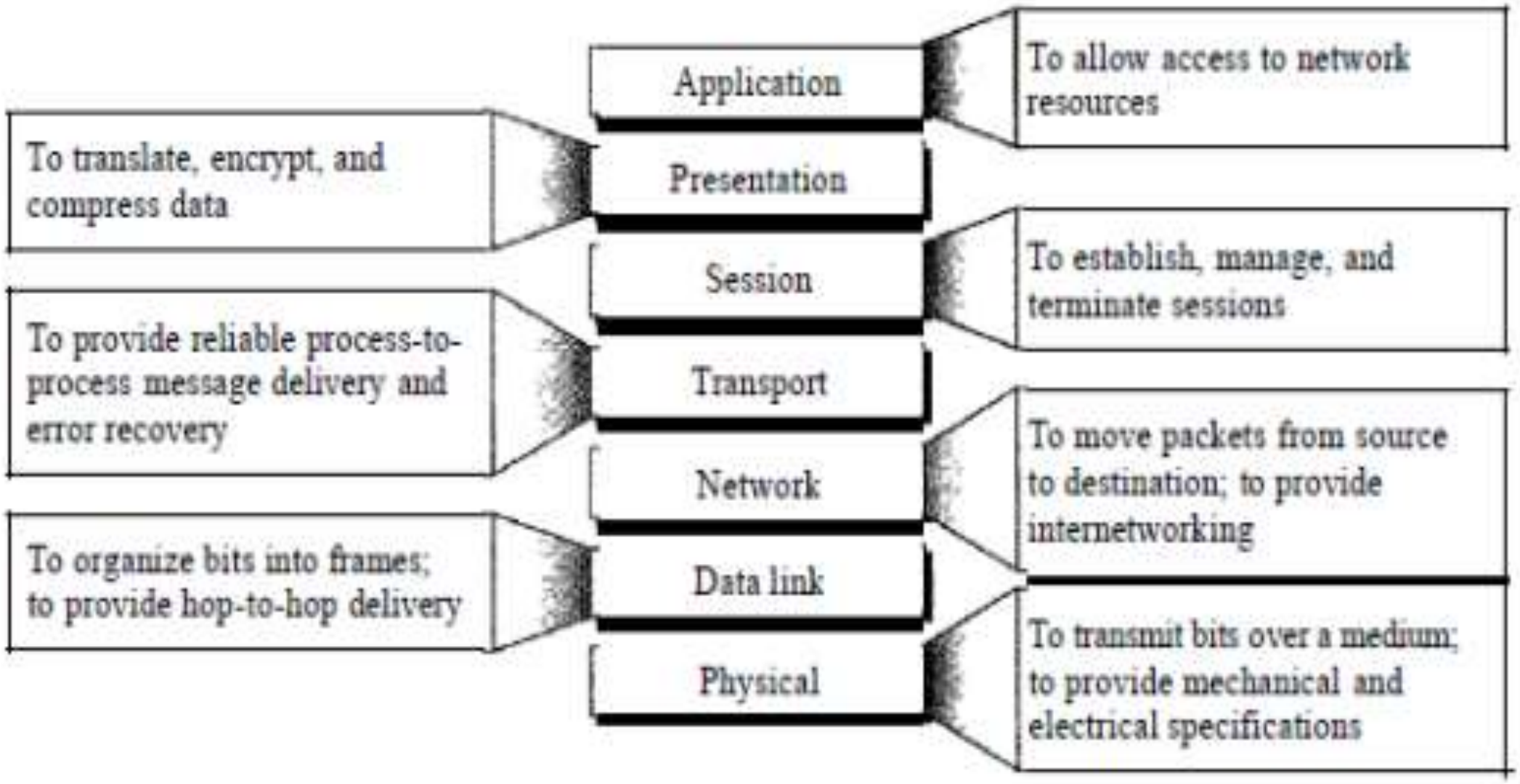
The OSI Model

7. Application Layer (Layer 7) :

- At the very top of the OSI Reference Model stack of layers, we find the Application layer which is implemented by the network applications. These applications produce the data, which has to be transferred over the network. This layer also serves as a window for the application services to access the network and for displaying the received information to the user.
- Example: Application – Browsers, Skype Messenger, etc.
- *Application Layer is also called Desktop Layer.*

- The **functions** of the Application layer are :
  - Network Virtual Terminal: A virtual terminal allows a PC to connect to a remote server, usually to perform a file transfer or run an application.
  - FTAM-File transfer access and management
  - Mail Services
  - Directory Services

SUMMARY OF LAYERS



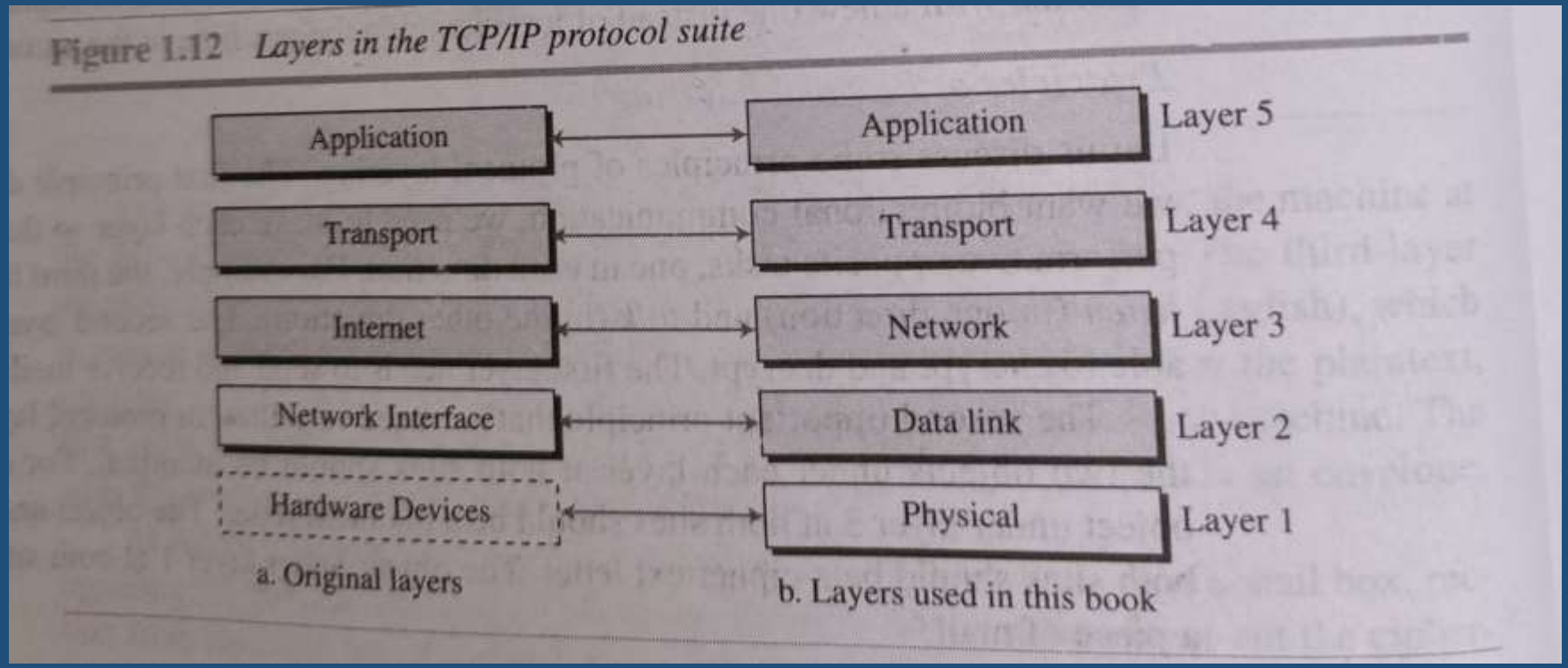
OSI MODEL

No.	Layer Name	Responsibility	Information Form (Data Unit)	Device
7	**Application Layer**	Helps in identifying the client and synchronize communication	Message	-
6	**Presentation Layer (Translation Layer)**	Data from application layer is extracted and manipulated as required format for transmission	Message	-
5	**Session Layer**	Establishes connection, maintenance, authentication and ensure security	Message	Gateway
4	**Transport Layer (HEART of OSI)**	Take service from network layer and provide it to application layer	Segment	Firewall
3	**Network Layer**	Transmission of data from one host to other. Located in different network	Packet	Router
2	**Data Link Layer**	Node to node delivery of messages	Frame	Switch, Bridge
1	**Physical Layer**	Establishing physical connection between devices	Bits	Hub, Repeater, Modem, Cables

INTERNET PROTOCOL

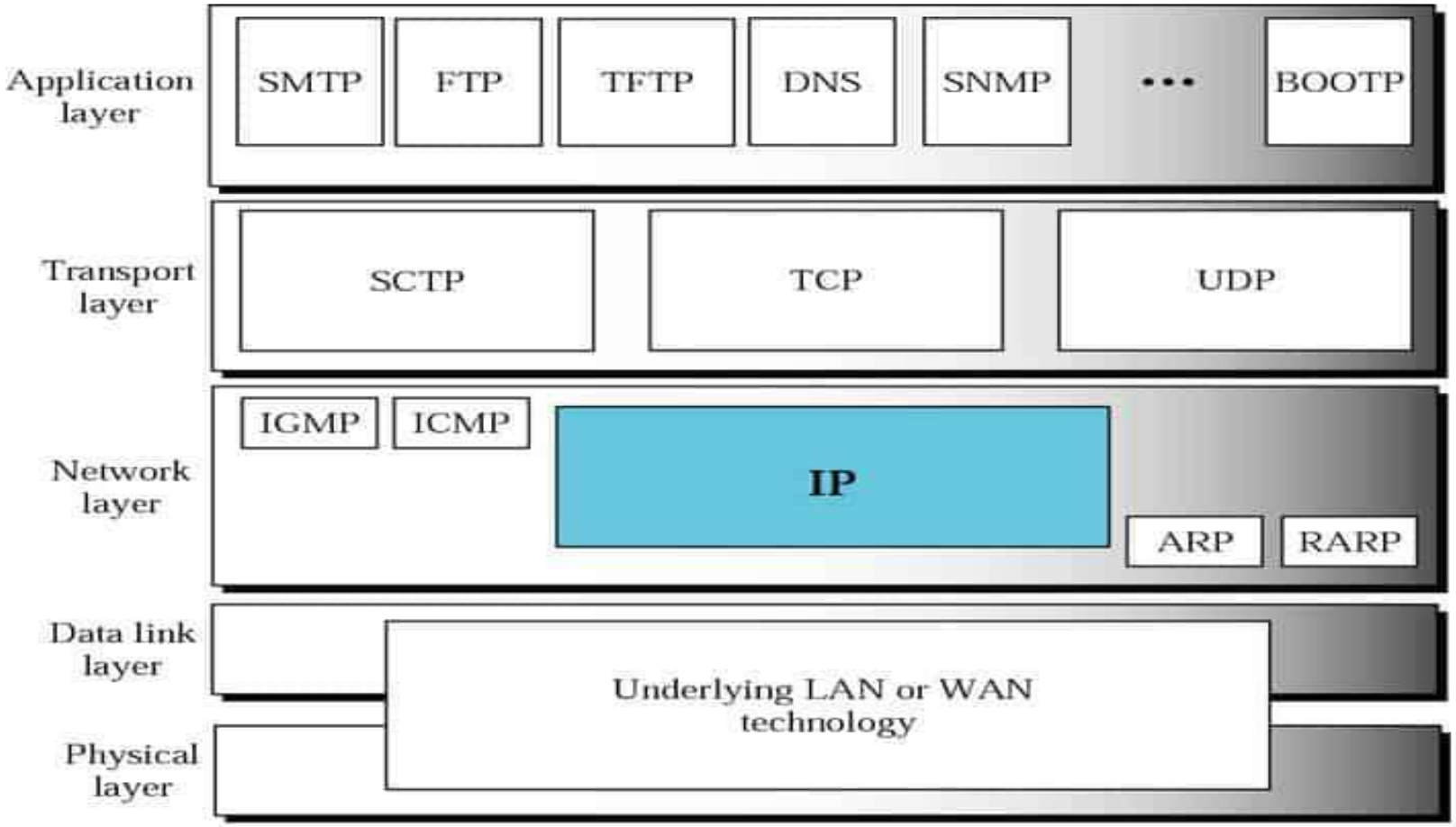
TCP/IP Protocol Suite

- A set of protocols organized in different layers.
- It is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality.
- Five – layer model



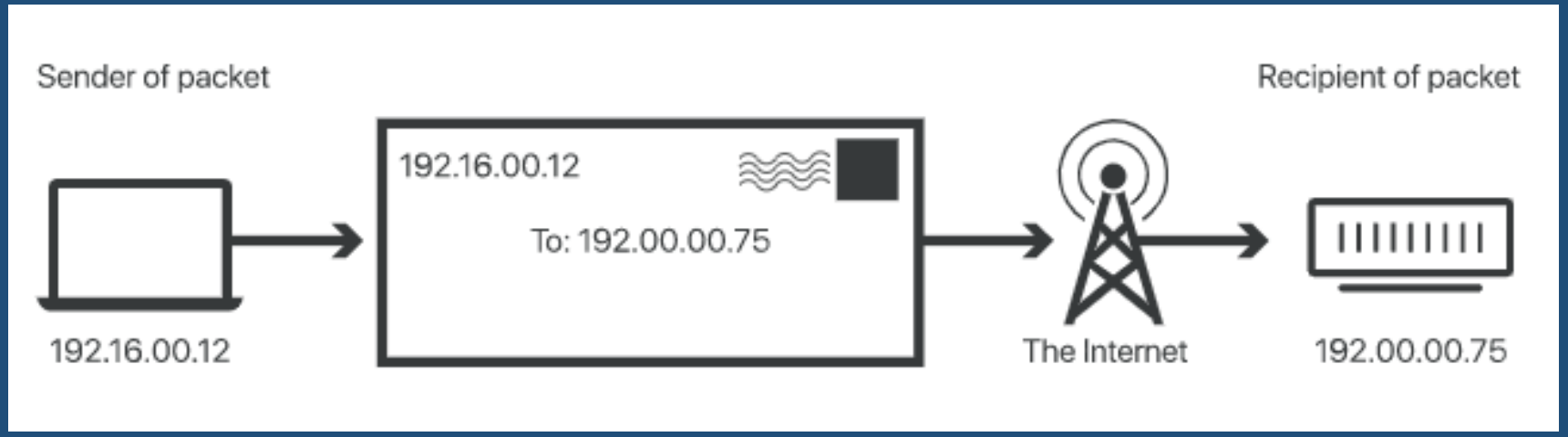
Internet Protocols

Position of IP in TCP/IP protocol suite



What is an IP address? How does IP addressing work?

- An IP address is a unique identifier assigned to a device or domain that connects to the Internet. Each IP address is a series of characters, such as '192.168.1.1'. Via DNS resolvers, which translate human-readable domain names into IP addresses, users are able to access websites without memorizing this complex series of characters. Each IP packet will contain both the IP address of the device or domain sending the packet and the IP address of the intended recipient, much like how both the destination address and the return address are included on a piece of mail.



Internet Protocols

- The **Internet Protocol** (**IP**) is the principal communications protocol in the Internet protocol suite for ***relaying datagram across network boundaries***.
- Its ***routing function*** enables ***internetworking, and essentially establishes the Internet.***

- IP has the task of ***delivering packets*** *from the source host to the destination host solely based on the* ***IP addresses*** in the packet headers. For this purpose, IP defines packet structures that encapsulate the data to be delivered. It also defines addressing methods that are used to label the datagram with source and destination information.

- The Internet Protocol is **responsible for addressing hosts and for routing datagrams (packets) from a source host to a destination host across one or more IP networks.**

- For this purpose, the *Internet Protocol defines the format of packets and provides an addressing system that has two functions:*
  - *Identifying hosts and*
  - *providing a logical location service*

Datagram construction

- Each datagram has two components:
  - **a header and a payload.**
- The **IP header** is tagged with the
  - *source IP address,*
  - *the destination IP address,*
  - *other meta-data needed to route and deliver the datagram*.
- The **Payload**
  - the data that is transported.

This method of nesting the data payload in a packet with a header is called **encapsulation**.

Internet Protocols

- IP specifies
  - the format of packets, and the addressing scheme.
- Most networks combine IP with a higher-level protocol called *Transmission Control Protocol (TCP)*,
  - which establishes a virtual connection between a destination and a source.
- *IP by itself is something like the postal system.*
  - *It allows you to address a package and drop it in the system, but there's no direct link between you and the recipient.*
  - *TCP/IP, on the other hand, establishes a connection between two hosts so that they can send messages back and forth for a period of time.*

Internet Protocols

- Short for **Internet Protocol address**, an **IP** or **IP address** is a number used to
  - indicate the location of a computer or other device on a network using TCP/IP.
- These addresses are similar to those of your house;
  - they allow data to reach the appropriate destination on a network and the Internet.
- As the Internet and technology evolve, there has been an increasing demand for IP addresses.
- To help meet the demand for IP addresses, there are two types of addresses used today, **IPv4 and IPv6**.
- IPv4 address in your local home, school, or small office.
  - Example of an **IPv4** address: (32 Bit format)
    - 45.79.151.23
  - Example of an **IPv6** address: (128 Bit format)
    - 2601:681:4200:c5c0:516:f0bb:ac3b:46bd

The role of standard organizations

- The rise of open standards not owned by any one company has been a great boon to customers of computer and networking products, as well as the manufacturers that sell to them.
- In order to facilitate the development of open standards, organizations are needed that will coordinate the creation and publishing of these *documents*. Generally, these are *non-profit organizations* that specifically take a neutral stance regarding technologies and work for the betterment of the industry as a whole.
- Broadly, an international standards organization develops international standards. There are many international standards organizations. The three largest and most well-established such organizations are the International Organization for Standardization (ISO), the International Electrotechnical Commission (IEC), and the International Telecommunication Union ITU).

The role of standard organizations

- Here are some of the standards organizations that you are likely to encounter when reading about networking and the Internet:

1. **International Organization for Standardization (ISO)**
2. **International Telecommunication Union - Telecommunication Standardization Sector (ITU-T)**
3. **Institute of Electrical and Electronics Engineers (IEEE)**
4. **American National Standards Institute (ANSI)**
5. **Electronic Industries Alliance (EIA)**
6. **Telecommunications Industry Association (TIA)**
7. **Internet Architecture Board (IAB)**
8. **Internet Engineering Task Force (IETF)**
9. **Internet Research Task Force (IRTF)**

The Role of Standard Organizations

1. International Organization for Standardization

- ISO is the international organization for standardization on a wide range of subjects.
- It is comprised mainly of members from the standards committee of various governments throughout the world.
- It is even responsible for developing models which provides high level of system compatibility, quality enhancement, improved productivity and reduced costs.
- The ISO is also responsible for endorsing and coordinating the work of the other standards organizations.

The Role of Standard Organizations

2. International Telecommunications Union-Telecommunication Sector (ITU-T)

- ITU-T is one of the four permanent parts of the International Telecommunications Union based in Geneva, Switzerland.
- It has developed three sets of specifications: the **V series** for modem interfacing and data transmission over telephone lines, the **X series** for data transmission over public digital networks, email and directory services; the **I and Q series 1** for Integrated Services Digital Network (ISDN) and its extension Broadband ISDN.
- ITU-T membership consists of government authorities and representatives from many countries and it is the present standards organization for the United Nations.

The Role of Standard Organizations

3. Institute of Electrical and Electronics Engineers (IEEE)

- IEEE is an international professional organization founded in United States and is compromised of electronics, computer and communications engineers.
- It is currently the world's largest professional society with over 200,000 members.
- It develops communication and information processing standards with the underlying goal of advancing theory, creativity, and product quality in any field related to electrical engineering.

The Role of Standard Organizations

4. American National Standards Institute (ANSI)

- ANSI is the official standards agency for the United States and is the U.S voting representative for the ISO.
- ANSI is a completely private, non-profit organization comprised of equipment manufacturers and users of data processing equipment and services.
- ANSI membership is comprised of people form professional societies, industry associations, governmental and regulatory bodies, and consumer goods.

The Role of Standard Organizations

5. Electronics Industry Association (EIA)

- EIA is a non-profit U.S. trade association that establishes and recommends industrial standards.
- EIA activities include standards development, increasing public awareness, and it is responsible for developing the RS (recommended standard) series of standards for data and communications.

The Role of Standard Organizations

6. Telecommunications Industry Association (TIA)

- TIA is the leading trade association in the communications and information technology industry.
- It facilitates business development opportunities through market development, trade promotion, trade shows, and standards development.
- It represents manufacturers of communications and information technology products and also facilitates the convergence of new communications networks.

The Role of Standard Organizations

7. Internet Architecture Board (IAB)

- IAB earlier known as Internet Activities Board is a committee created by ARPA (Advanced Research Projects Agency) so as to analyze the activities of ARPANET whose purpose is to accelerate the advancement of technologies useful for U.S military.
- IAB is a technical advisory group of the Internet Society and its responsibilities are:

  I. Oversees the architecture protocols and procedures used by the Internet.

  II. Manages the processes used to create Internet Standards and also serves as an appeal board for complaints regarding improper execution of standardization process.

  III. Responsible for administration of the various Internet assigned numbers

  IV. Acts as a representative for Internet Society interest in keeping relationships with other organizations.

  V. Acts as a source of advice and guidance to the board of trustees and officers of Internet Society concerning various aspects of internet and its technologies.

The Role of Standard Organizations

8. Internet Engineering Task Force (IETF)

- The IETF is a large international community of network designers, operators, vendors and researchers concerned with the evolution of the Internet architecture and smooth operation of the Internet.

9. Internet Research Task Force (IRTF)

- The IRTF promotes research of importance to the evolution of the future Internet by creating focused, long-term and small research groups working on topics related to Internet protocols, applications, architecture and technology.

Security in the Internet

- Internet security is a central aspect of cybersecurity, and it includes **managing cyber threats and risks associated with the Internet, web browsers, web apps, websites and networks**.

- Internet security is a term that describes security for activities and transactions made over the internet.

- The primary purpose of Internet security solutions is to protect users and corporate IT assets from attacks that travel over the Internet.

Security in the Internet

Types of Security

- ***Computer Security***

  - generic name for the collection of tools designed to protect data and to prevent hackers

- ***Network Security***

  - measures to protect data during their transmission

- ***Internet Security***

  - measures to protect data during their transmission over a collection of interconnected networks

Security in the Internet

Goals of Security

- **Confidentiality** – prevents unauthorized use or disclosure of information
- **Integrity** - safeguards the accuracy and completeness of information
- **Availability** – authorized users have reliable and timely access to information

Security in the Internet

- Part of a field of study known as cryptology
- **Cryptology includes:**
  - **Cryptography:**
    - Study of methods for secret writing
    - Transforming messages into unintelligible form
    - Recovering messages using some secret knowledge (key)
  - **Cryptanalysis:**
    - Analysis of cryptographic systems, inputs and outputs
    - To derive confidential information

Security in the Internet

Cryptography- Has evolved into a complex science in the field of information security.

- **Encryption** – process of *transforming plaintext to ciphertext using a cryptographic key*

- **Symmetric key cryptography** – *uses a single key to both encrypt and decrypt information. Also known as private key.*

  - Includes DES, 3DES, AES, IDEA, RC5, Blowfish

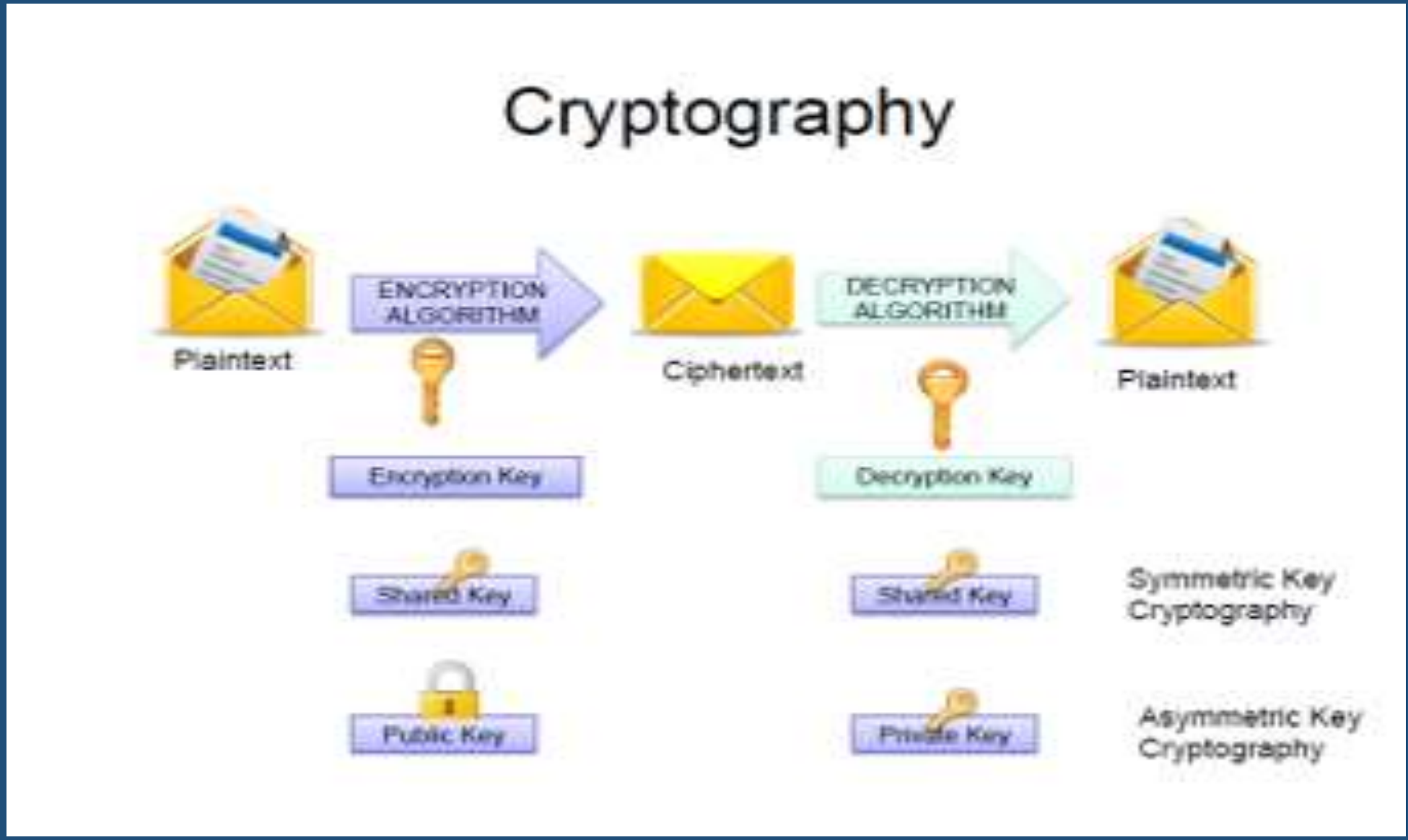
- **Asymmetric key cryptography** – *separate keys for encryption and decryption (public and private key pairs)*

  - Includes RSA, Diffie-Hellman

Security in the Internet

- **Terminology of Cryptography**
  - **Cipher**
    - Cryptographic technique (algorithm) applying a secret transformation to messages
  - **Plaintext / cleartext**
    - Original message or data
  - **Encryption**
    - Transforming plaintext, using a secret key, so meaning is concealed
  - **Ciphertext**
    - Unintelligible encrypted plaintext
  - **Decryption**
    - Transforming ciphertext back into original plaintext
  - **Cryptographic Key**
    - Secret knowledge used by cipher to encrypt or decrypt message

Security in the Internet



Security in the Internet

- **Symmetric Key Algorithm/Secret Key Algorithm**

  Stream ciphers – encrypts bits of the message at a time

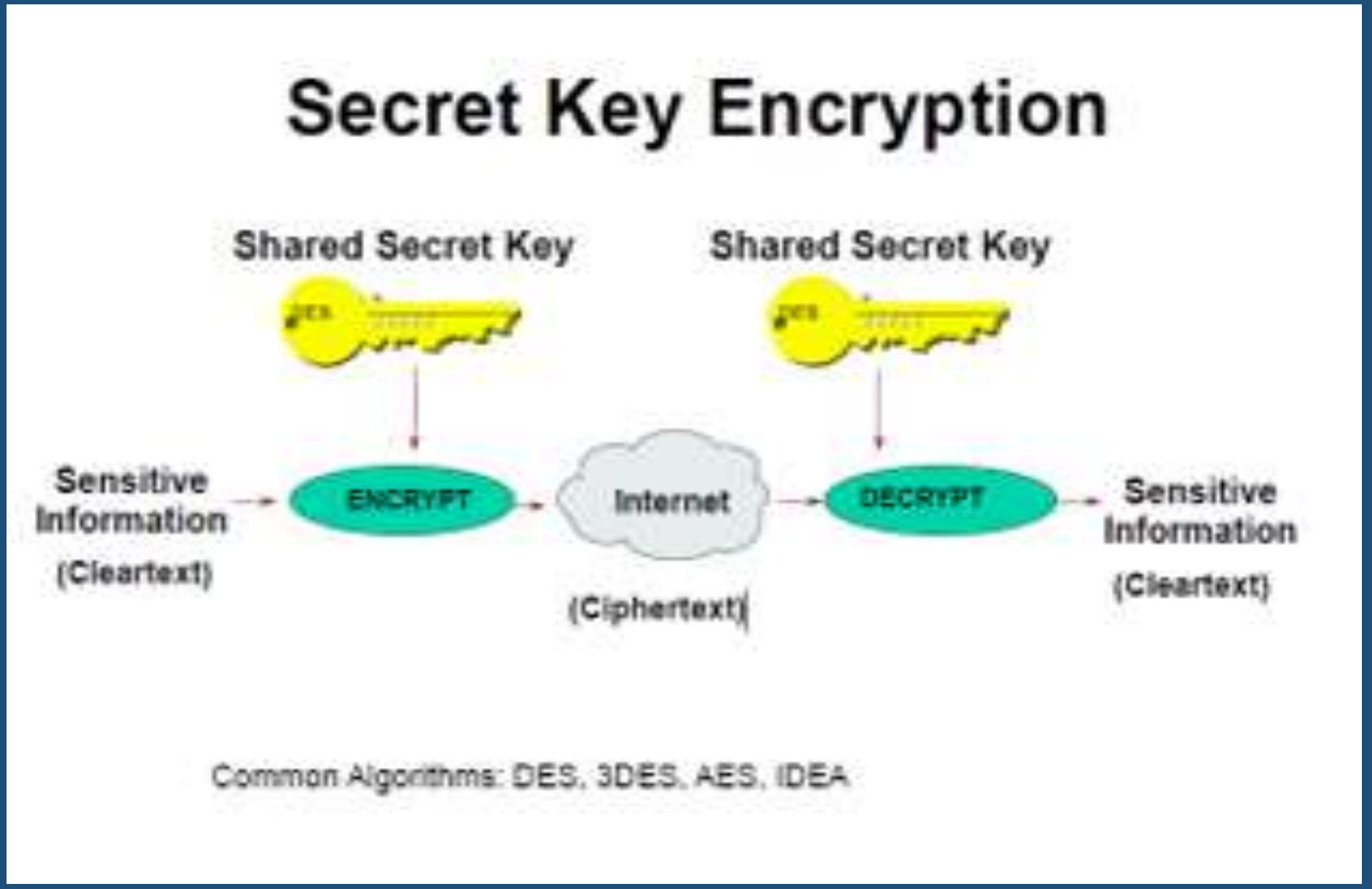
  Block ciphers – takes a block of bits and encrypts them as a single unit

Security in the Internet

Symmetric/Secret Key Algorithms

- DES (Data Encryption Standard) – block cipher using shared key encryption, 56-bit
- 3DES (Triple DES) – a block cipher that applies DES three times to each data block
- AES (Advanced **Encryption** Standard) – replacement for DES; current standard

Security in the Internet



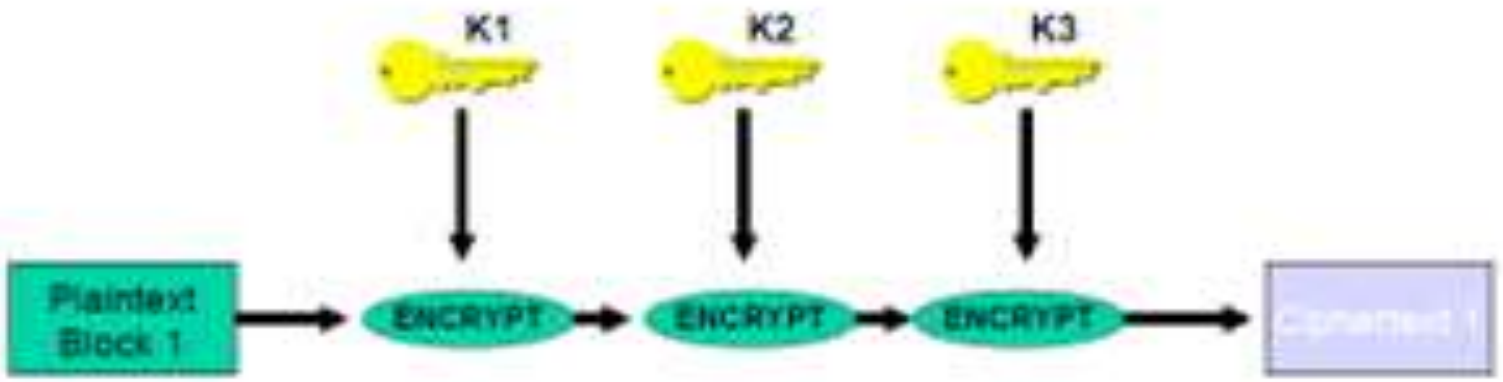
Security in the Internet

Triple DES

- 3DES (Triple DES) – a block cipher that applies DES three times to each data block
- Uses a key bundle comprising of three DES keys (K1, K2, K3), each with 56 bits excluding parity.
- DES encrypts with K1, decrypts with K2, then encrypts with K3
- Disadvantage: *very slow*

Security in the Internet

Triple DES (3DES)



- Many applications use K3=K1, yielding a key length of 112 bits
- Interoperable with conventional DES if K1=K2=K3

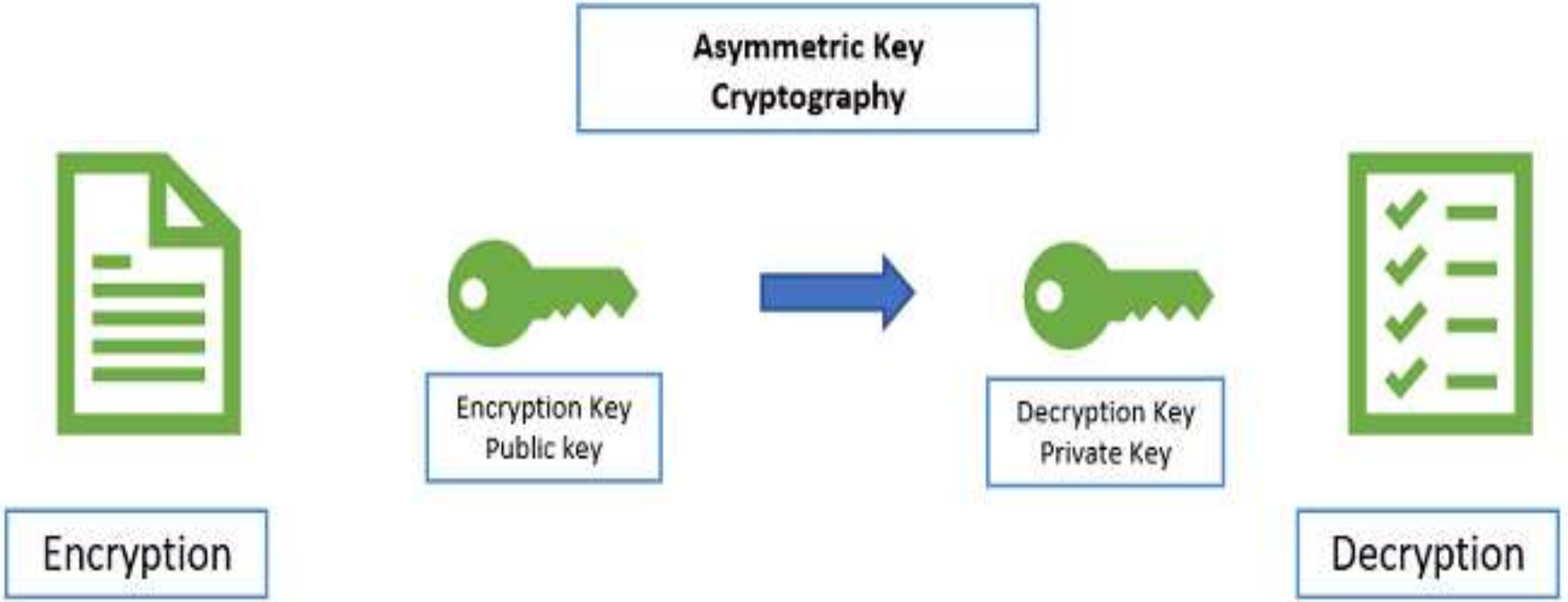
Security in the Internet

- Asymmetric keys are the foundation of Public Key Infrastructure (PKI) **a cryptographic scheme requiring two different keys, one to lock or encrypt the plaintext, and one to unlock or decrypt the cyphertext**.
- The main protocol here used **is Diffie-Hellmann Key Exchange (DHKE) protocol**.

Diffie-Hellman

- Diffie-Hellman Protocol – requires that both the sender and recipient of a message have key pairs.
- Combining one's private key and the other's public key, both parties can compute the same shared secret number.

Security in the Internet



Quality of Service (QoS)

Quality of Service (QoS)

- Quality of Service (QoS) refers to the capability of a network to provide better service to selected network traffic over various technologies.

- Quality of service (QoS) is the use of mechanisms or technologies that work on a network to control traffic and ensure the performance of critical applications with limited network capacity.
- Quality of service (QoS) is an internetworking issue that refers to a set of techniques and mechanisms that guarantees the performance of the network to deliver predictable service to an application program.

- The **primary goal of QoS** is to provide priority including dedicated bandwidth, controlled jitter and latency and improved loss characteristics.
- Jitter and latency are the metrics used to assess the network's performance. The major distinction between jitter and latency is that latency is defined as a delay via the network, whereas jitter is defined as a change in the amount of latency.

- Important is making sure that providing priority for one or more flows does not make other flows fail.

Quality of Service (QoS)

- QoS features provide improved and more predictable network service by providing the following services:
    - Supporting dedicated bandwidth
    - Improving loss characteristics
    - Avoiding and managing network congestion
    - Shaping network traffic
    - Setting traffic priorities across the network

Concept of Quality of Service (QoS)

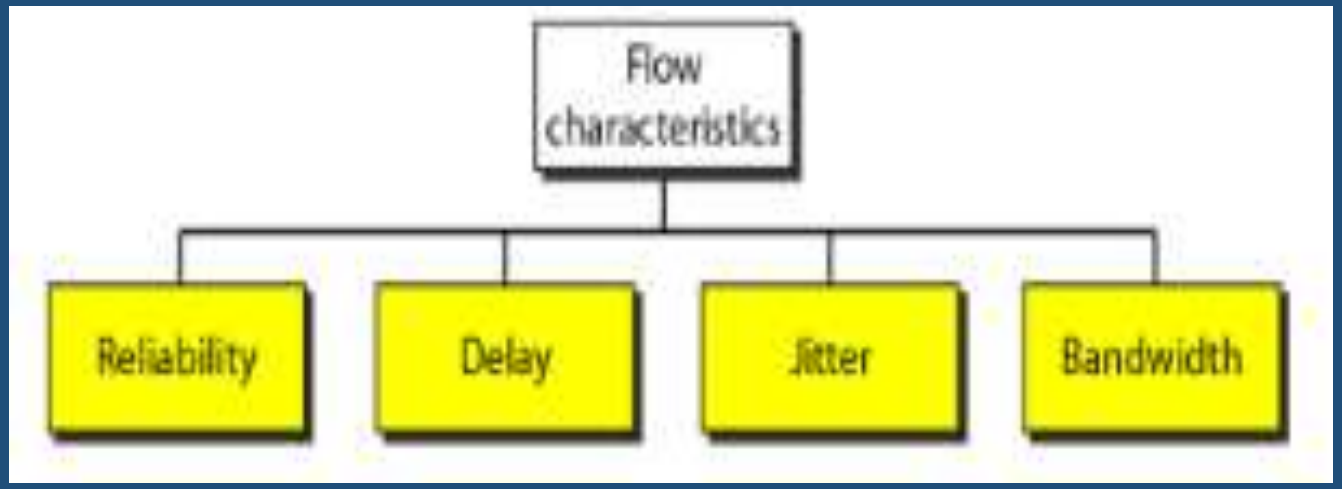
- QoS features throughout a network to provide for **end-to-end** QoS delivery. The following **three components** are necessary to deliver *QoS across a heterogeneous network*:
    - QoS within a single network element, which includes queueing, scheduling, and traffic shaping features.
    - QoS signalling techniques for coordinating QoS for end-to-end delivery between network elements.
    - QoS policing and management functions to control and administer end-to-end traffic across a network.

Data Flow characteristics

- To provide quality of service for an internet application, we need to know what is needed for the application.

- The characteristics attributed to a flow are:

- Data Flow Characteristics are
  - Reliability
  - Delay
  - Jitter
  - Bandwidth



Data Flow characteristics

1.Reliability: characteristic that the flow needs in order to deliver the packets safe and sound to the destination. Lack of reliability means losing a packet or acknowledgement, which causes retransmission.

2.Delay: source to destination delay is another characteristic. Applications can tolerate delay indifferent degrees.

Data Flow characteristics

3. Jitter: is the variation of delay for packets belonging to the same flow. For example, if four packets depart at times 0, 1, 2, 3 and arrive at 20, 21, 22, 23, all have the same delay, 20 units of time.

On the other hand, if the above four packets arrive at 21, 23, 21,and 28, they will have different delays. For applications such as audio and video, the first case is completely acceptable; the second case is not. For these applications, it does not matter if the packets arrive with a short or long delay as long as the delay is the same for all packets. High jitter means the difference between delays is large; low jitter means the variation is small

4.Bandwidth: different applications need different bandwidths.

Techniques that can be used to improve the quality of service.

1.Scheduling

- FIFO Queue
- Priority queuing
- Weighted fair queuing

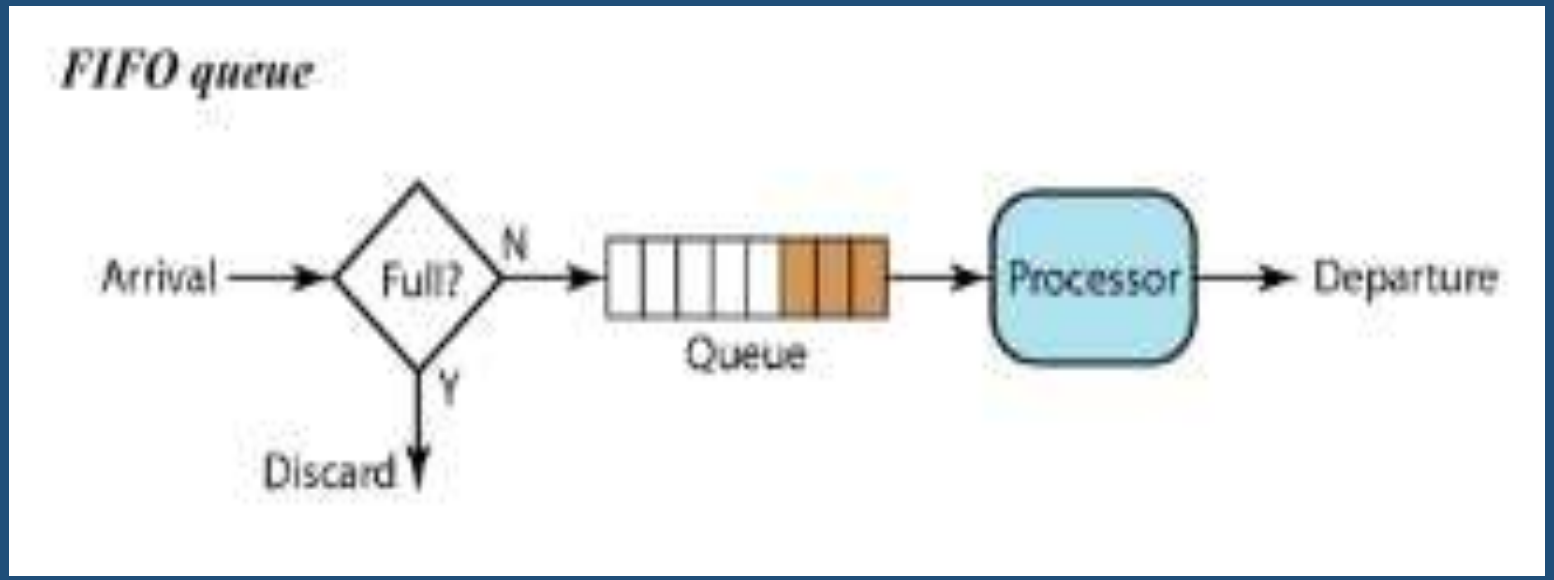
2. Traffic shaping :

- Leaky Bucket
- Token Bucket

1.Scheduling

- FIFO Queue

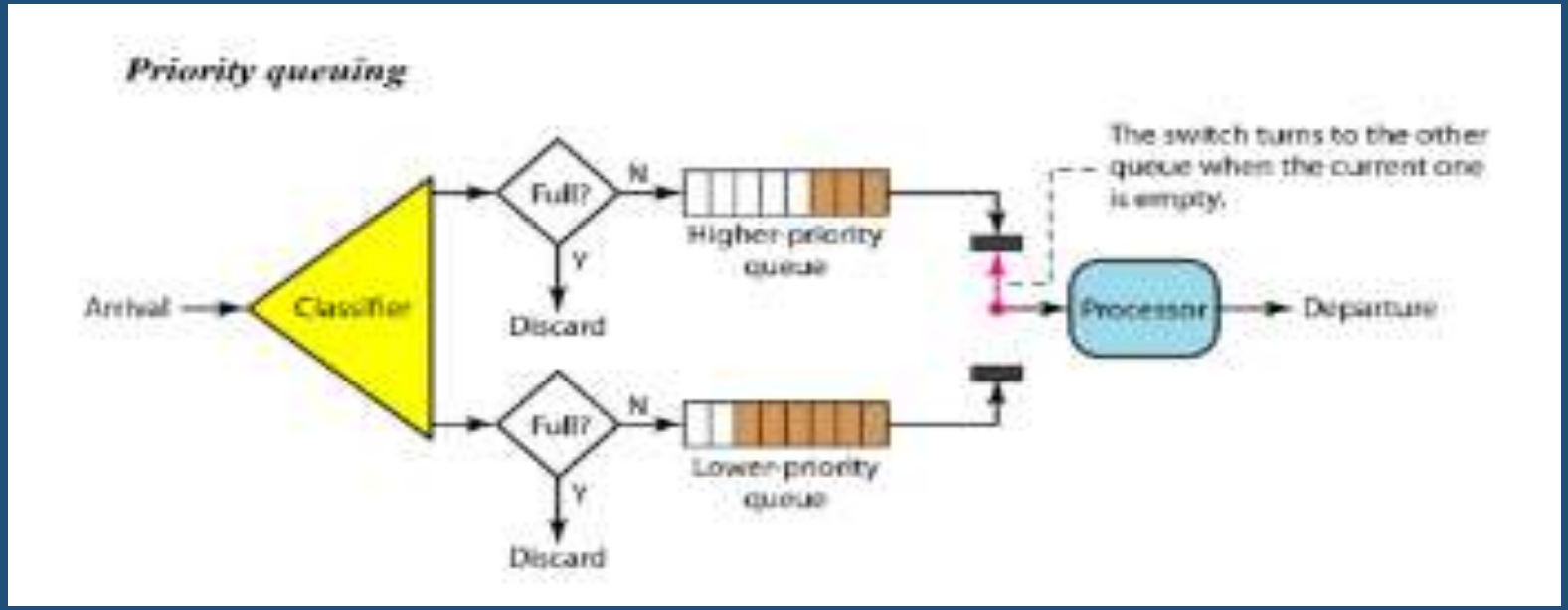
  •In first-in, first-out (FIFO) queuing, packets wait in a buffer (queue) until the node (router or switch) is ready to process them. If the average arrival rate is higher than the average processing rate, the queue will fill up and new packets will be discarded.



1.Scheduling

- Priority queuing

A priority queue can provide better QoS than the FIFO queue because higher priority traffic, such as multimedia, can reach the destination with less delay. However, there is a potential drawback. If there is a continuous flow in a high-priority queue, the packets in the lower-priority queues will never have a chance to be processed. This is a condition called starvation.



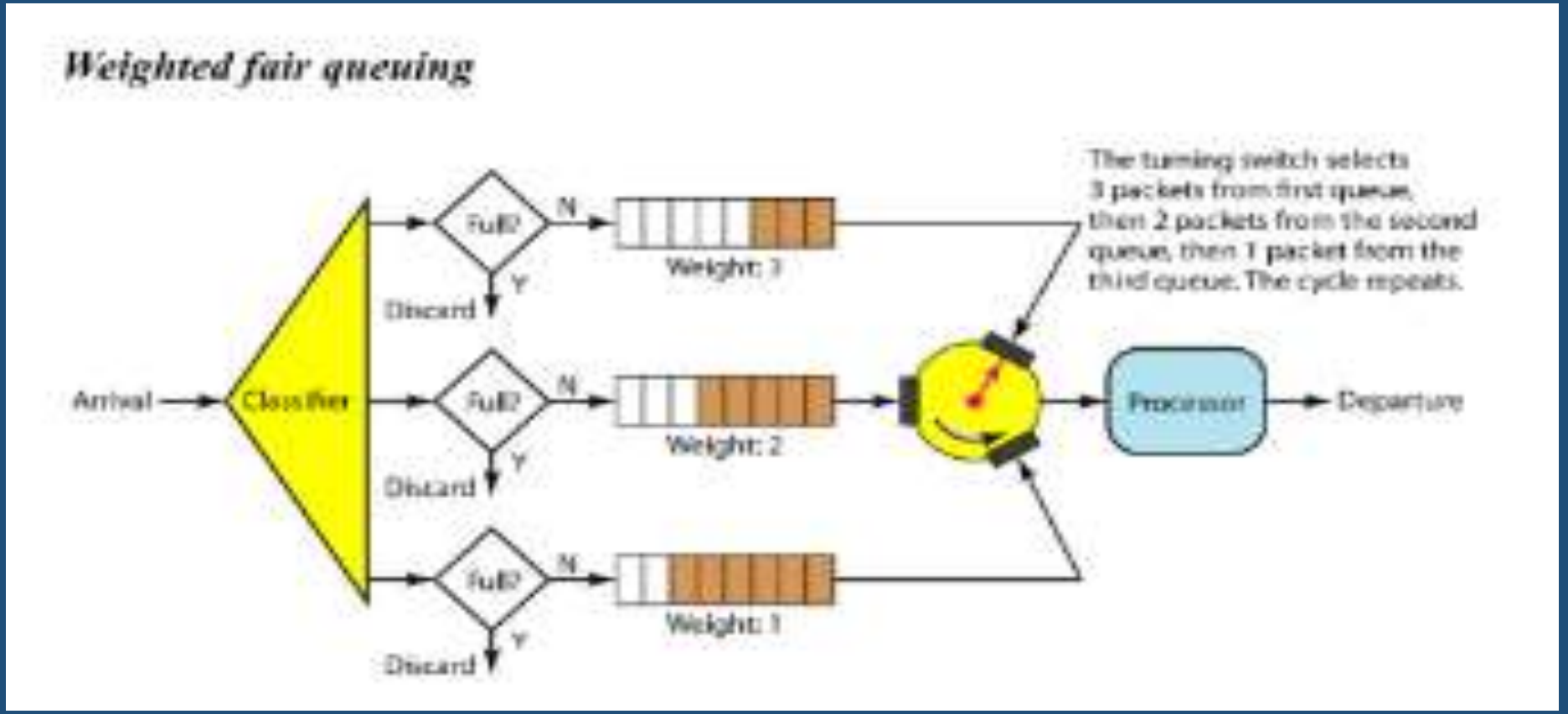
1.Scheduling

- Weighted fair queuing

A better scheduling method is weighted fair queuing. In this technique, the packets are still assigned to different classes and admitted to different queues. The queues, however, are weighted based on the priority of the queues; higher priority means a higher weight. The system processes packets in each queue in a round-robin fashion with the number of packets selected from each queue based on the corresponding weight. For example, if the weights are 3, 2, and 1, three packets are processed from the first queue, two from the second queue, and one from the third queue. If the system does not impose priority on the classes, all weights can be equal. In this way, we have fair queuing with priority.

1.Scheduling

- Weighted fair queuing



2. Traffic shaping

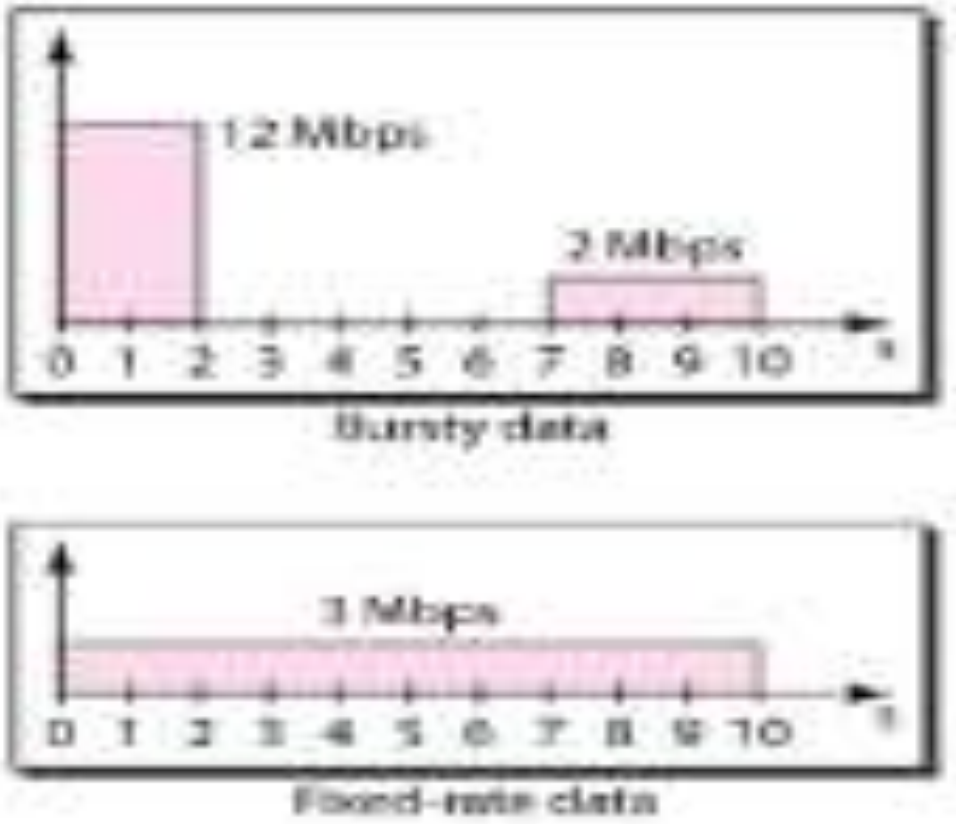
- It is a mechanism to control the amount and the rate of the traffic sent to the network.
- **Two techniques can shape traffic:**
  1. **Leaky bucket**
  2. **Token bucket**

1. Leaky Bucket

- If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rate as long as there is water in the bucket. The rate at which the water leaks does not depend on the rate at which the water is input to the bucket unless the bucket is empty. The input rate can vary, but the output rate remains constant. Similarly, in networking, a technique called leaky bucket can smooth out bursty traffic. Bursty chunks are stored in the bucket and sent out at an average rate.

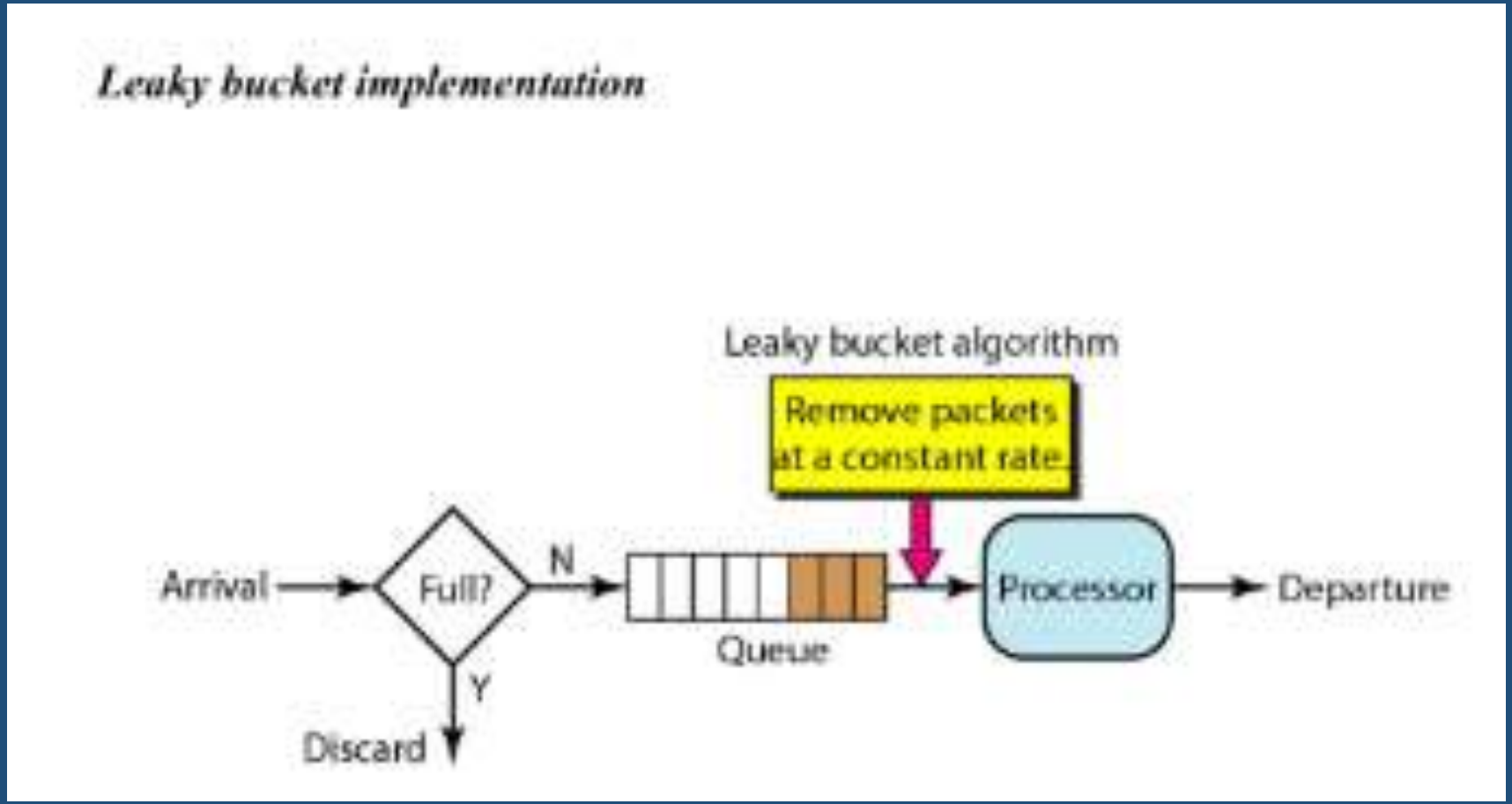
Traffic shaping :

Traffic Shaping or Policing: *Leaky bucket*



2. Traffic shaping

A FIFO queue holds the packets. If the traffic consists of fixed-size packets, the process removes a fixed number of packets from the queue at each tick of the clock. If the traffic consists of variable-length packets, the fixed output rate must be based on the number of bytes or bits.

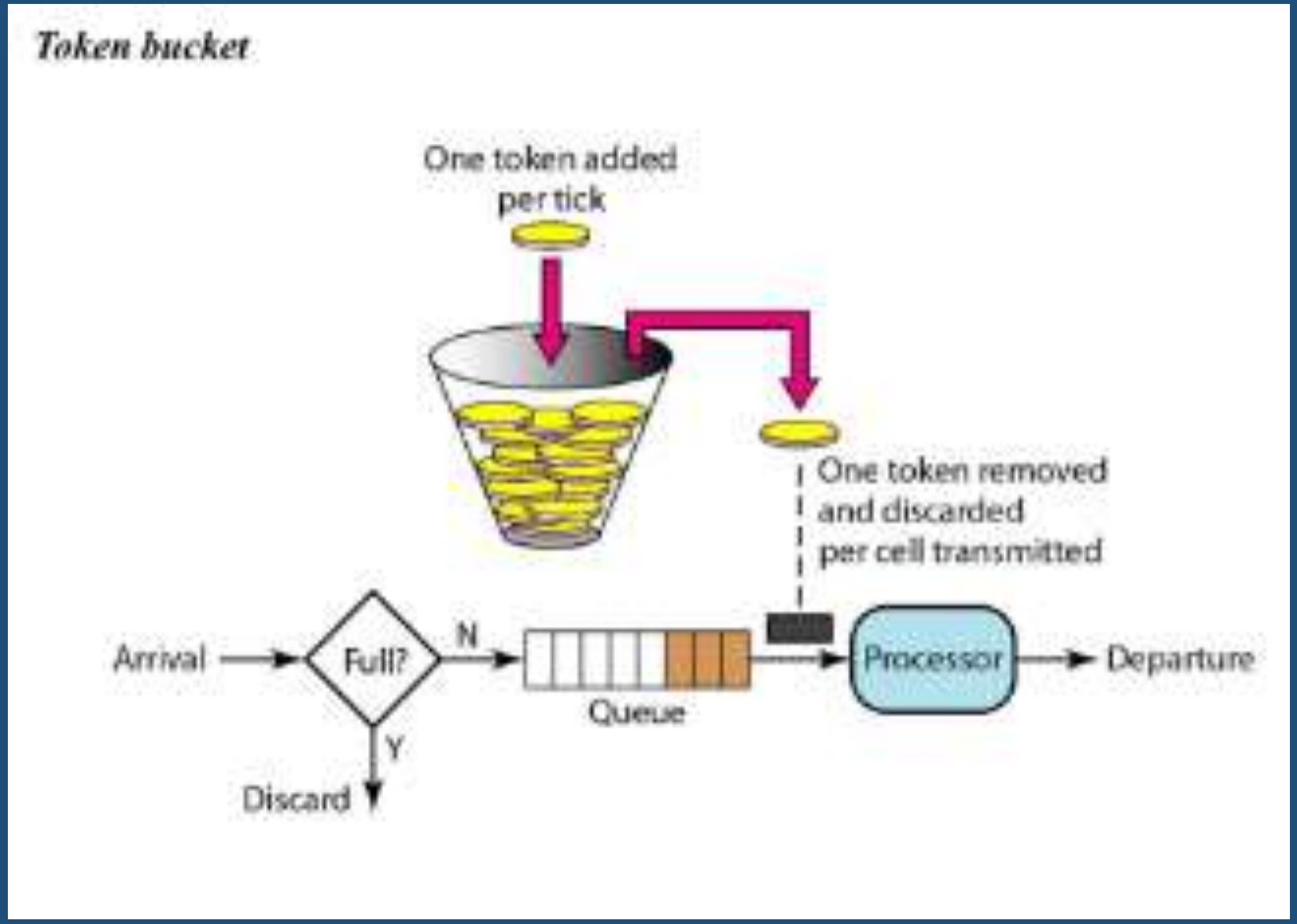


Traffic shaping

2. Token Bucket

- The token bucket algorithm allows idle hosts to accumulate credit for the future in the form of tokens. For each tick of the clock, the system sends n tokens to the bucket.
- The host can send bursty data as long as the bucket is not empty.
- The token bucket can easily be implemented with a counter. The token is initialized to zero. Each time a token is added, the counter is incremented by 1. Each time a unit of data is sent, the counter is decremented by 1. When the counter is zero, the host cannot send data.
- The token bucket allows bursty traffic at a regulated maximum rate.

Traffic shaping



Traffic shaping

DIFFERENCE BETWEEN LEAKY BUCKET AND TOKEN BUCKET ALGORITHM

TOKEN BUCKET	LEAKY BUCKET
Token dependent.	Token independent.
If bucket is full token are discarded, but not the packet.	If bucket is full packet or data is discarded.
Packets can only transmitted when there are enough token	Packets are transmitted continuously.
It allows large bursts to be sent faster rate after that constant rate	It sends the packet at constant rate
It saves token to send large bursts.	It does not save token.

Electronic Mail

- Electronic mail has been around since the beginning of the Internet.
- It was the most popular application when the Internet was in its infancy and has become more and more elaborate and powerful over the years.
- It remains one of the Internet's most important and utilized applications.
- Electronic mail is a method of exchanging messages.
- E-mail systems are based on a **store-and-forward model**.
- E-mail computer server systems accept, forward, deliver and store messages on behalf of users.
- The general architecture of an e-mail system including the **three main components**:
  user agent,
  message transfer agent,
  and message access agent.

Entities in the Email System:

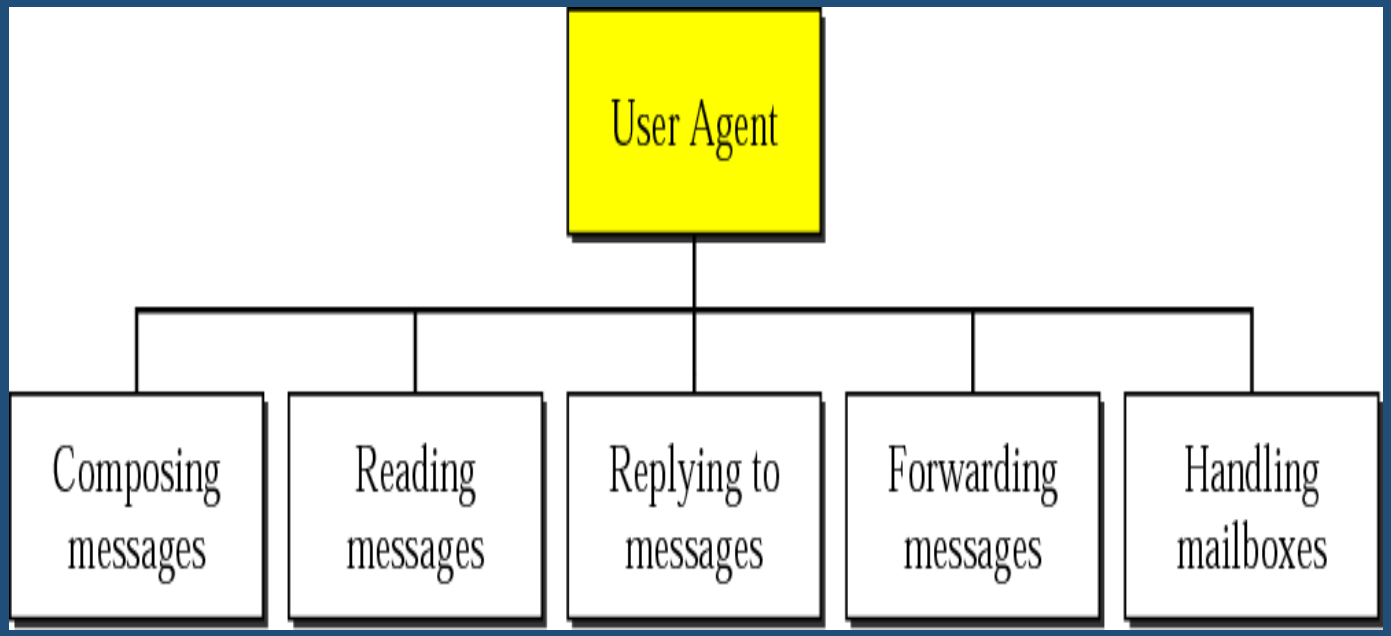
- **Client –** who sends the email (sender)
- **Server –** who receives the email (receiver)
- **Mailbox –** Area where the emails are stored
- **User Agent (Client) –** application used to prepare and send Email
- **User Agent (Server) -** application used to retrieve and read Email

User agent

- User Agent is the first component of email system. It provides service to the user to make the process of **sending and receiving a message** easier.

Services provided by a User Agent:

- A User Agent is a software program that composes, reads, replies to, and forwards messages. It also handles mailboxes.



Mail Transfer Phases

Three phases:

- **Connection Establishment**: Connection between client and server
- **Message Transfer**: Message between one sender and one or more recipient can be exchanged
- **Connection Termination**: After data transfer, Client sends QUIT command to terminate the connection.

Email Protocols

Mail Transfer Agent

- SMTP(Simple Mail Transfer Protocol)

Mail Access Agent

- POP(Post Office Protocol)
- IMAP(Internet Mail Access Protocol)

Message Transfer Agent (MTA): SMTP

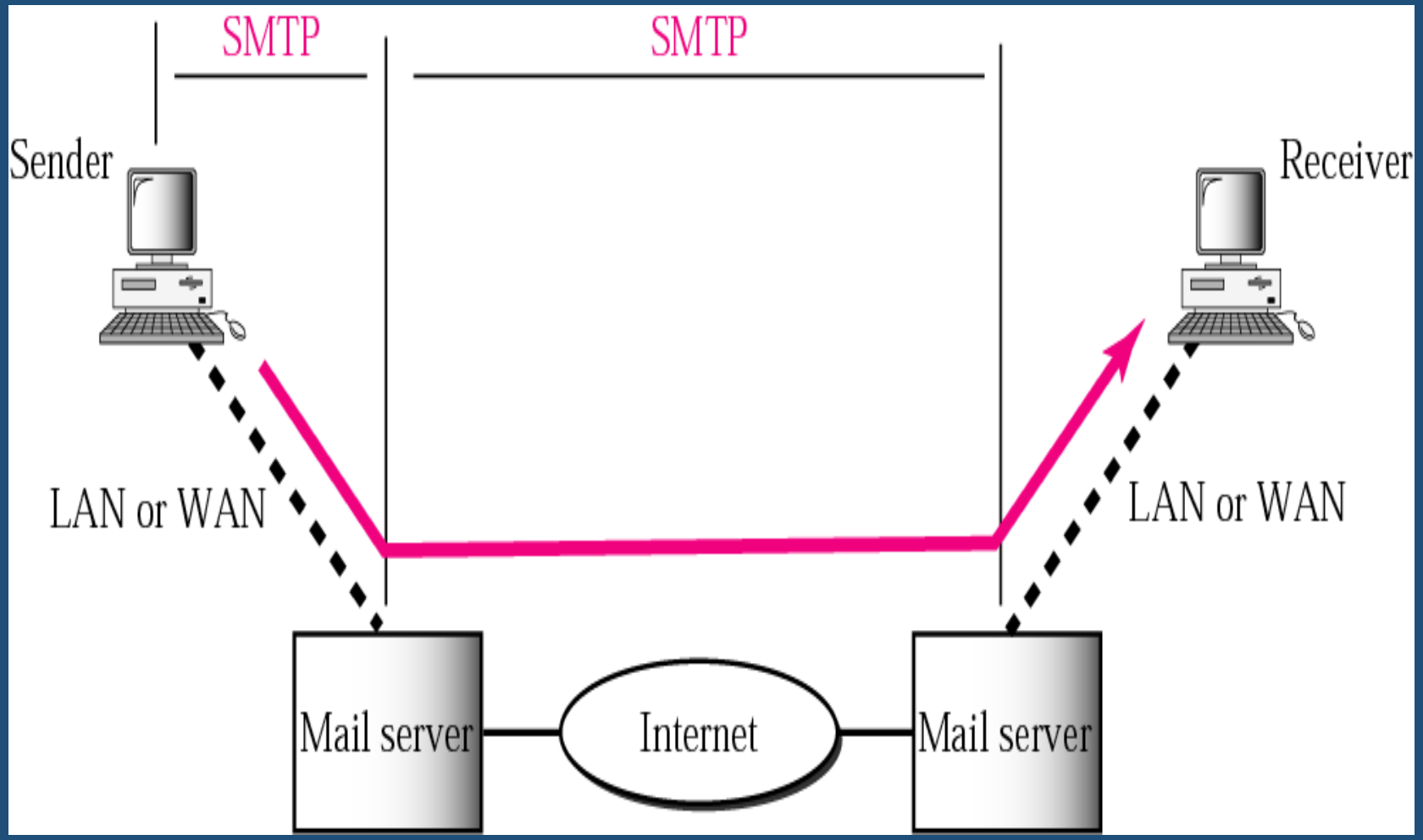
- MTA is a computer process or software agent that transfers email messages from one computer to another.

- An MTA implements in both the client (sending) and server (receiving) .ie, **MTA Client and MTA server.**

- The actual mail transfer is done through message transfer agents. To send mail, a system must have the **client MTA**, and to receive mail, a system must have a **server MTA.**

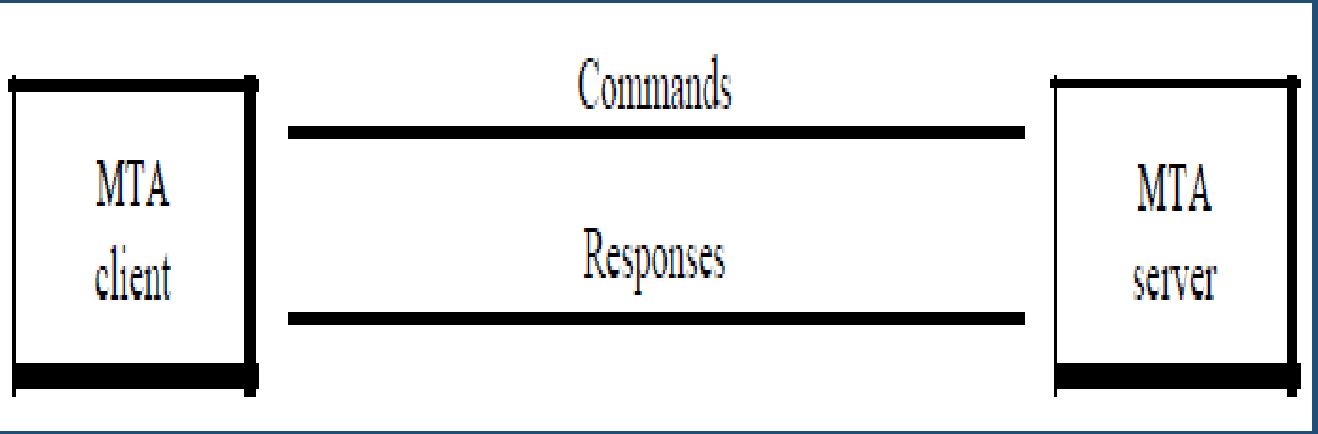
- **The formal protocol that defines the MTA client and MTA server in the Internet is called the Simple Mail Transfer Protocol (SMTP).**

Message Transfer Agent: SMTP



SMTP is used two times, between the sender and the sender's mail server and between the two mail servers. Another protocol called POP and IMAP are needed between the mail server and the receiver.

- SMTP uses commands and responses to transfer messages between an MTA client and an MTA server.



Scenario: Alice sends mail to Bob

- To illustrate the basic operation of SMTP, let's walk through a common scenario. Suppose Alice wants to send Bob a simple ASCII message.

1. Alice invokes her user agent for e-mail, provides Bob's e-mail address (for example, bob@someschool.edu), composes a message, and instructs the user agent to send the message.
2. Alice's user agent sends the message to her mail server, where it is placed in a message queue.
3. The client side of SMTP, running on Alice's mail server, sees the message in the message queue. It opens a TCP connection to an SMTP server, running on Bob's mail server.
4. After some initial SMTP handshaking, the SMTP client sends Alice's message into the TCP connection.
5. At Bob's mail server, the server side of SMTP receives the message. Bob's mail server then places the message in Bob's mailbox.
6. Bob invokes his user agent to read the message at his convenience.

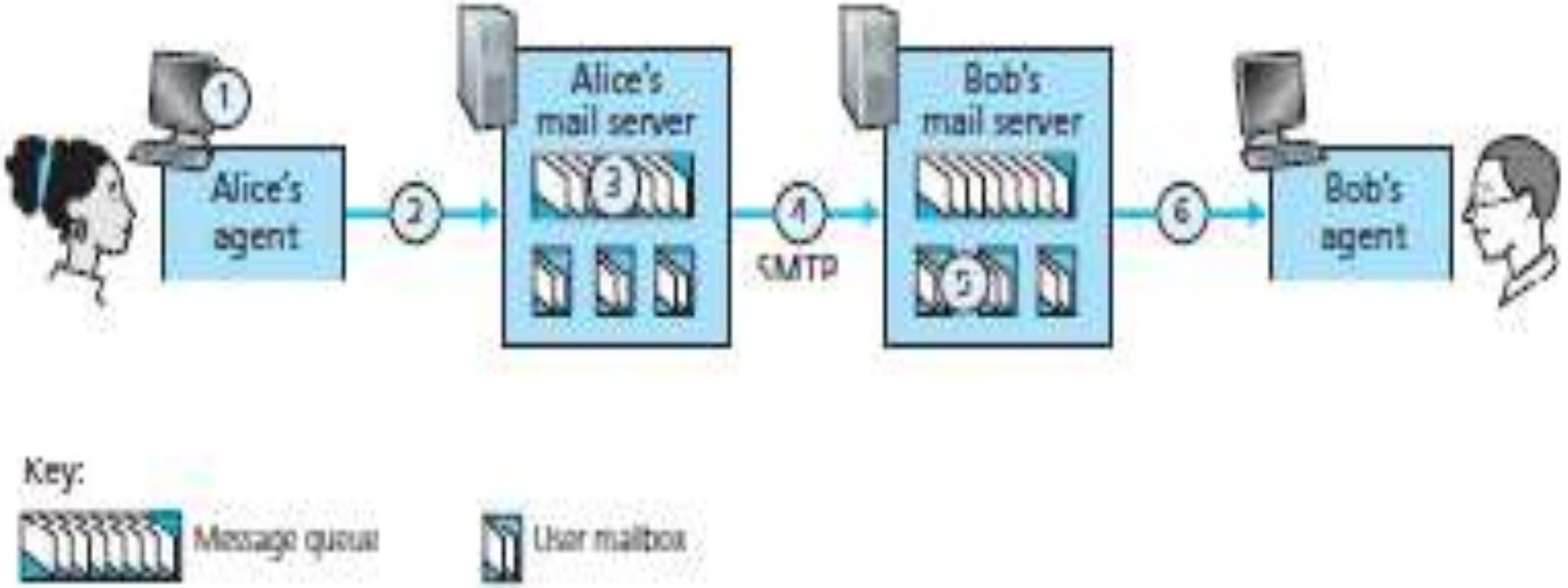


Figure 2.17 ♦ Alice sends a message to Bob

Comparison with HTTP

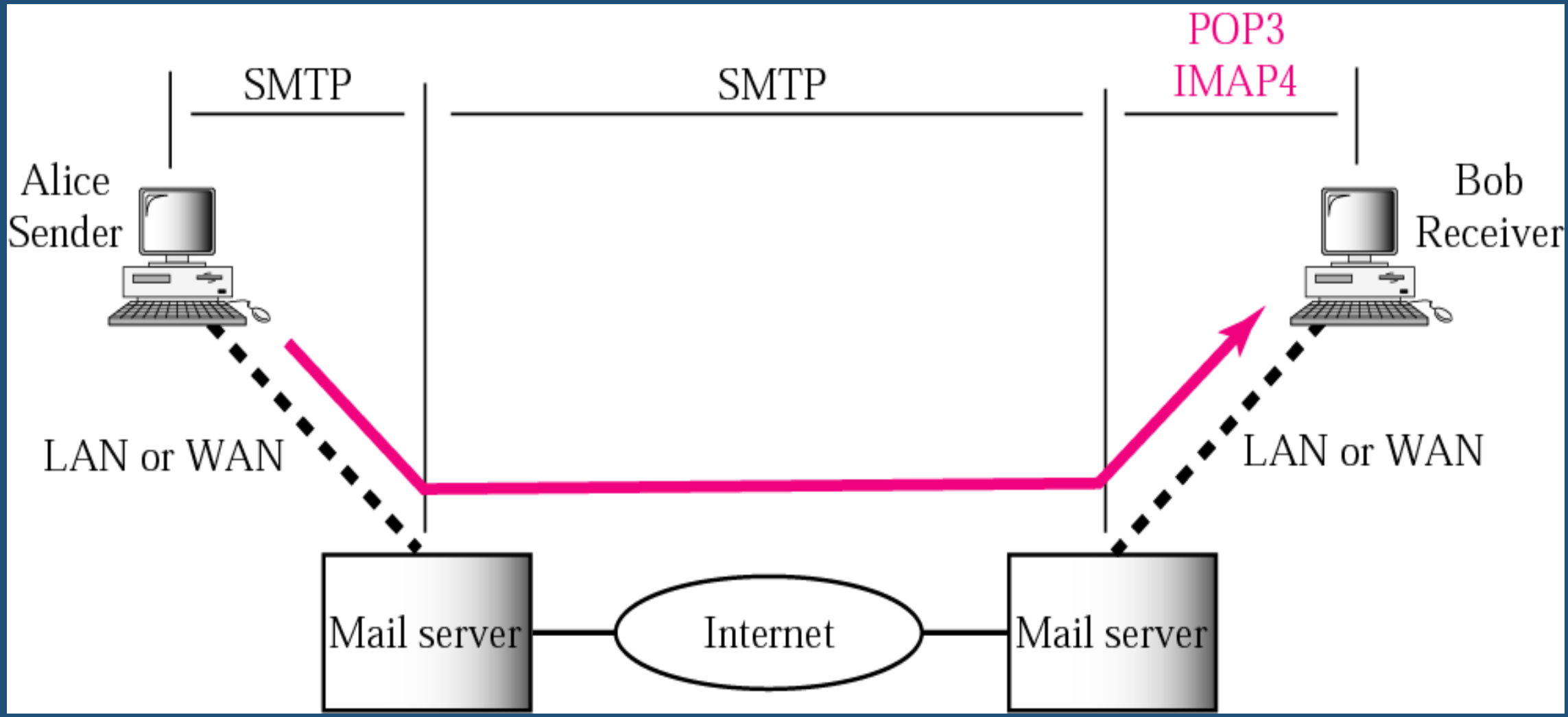
- Both protocols are used to transfer files from one host to another.
- HTTP transfers files (also called objects) from a Web server to a Web client (typically a browser).
- SMTP transfers files (that is, e-mail messages) from one mail server to another mail server.
- When transferring the files, HTTP use (persistent or non persistent) and SMTP use persistent connections.
- **HTTP is a pull protocol** : someone loads information on a Web server and users use HTTP to pull the information from the server at their convenience.
- **SMTP is a push protocol**: Sending mail server pushes the file to the receiving mail server.

Mail Access Agent(MAA): POP3 & IMAP4

- The first and the second stages of mail delivery use SMTP. However, SMTP is not involved in the third stage because **SMTP is a *push* protocol**; it pushes the message from the client to the server.

- In other words, the direction of the bulk: data (messages) is from the client to the server. On the other hand, **the third stage needs a *pull* protocol**; the client must pull messages from the server. The direction of the bulk data is from the server to the client. **The third stage uses a message access agent.**

- Currently **two message access protocols** are available:
  - **Post Office Protocol, version 3 (POP3)**
  - **Internet Mail Access Protocol, version 4 (IMAP4)**


POP3
IMAP4
SMTP
SMTP
Alice
Sender
Bob
Receiver
LAN or WAN
LAN or WAN
Mail server
Internet
Mail server

POP3 (Post Office Protocol, version 3)

- Post Office Protocol, version 3 (POP3) is simple and limited in functionality.
- The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.
- Mail access starts with the client when the user needs to download e-mail from the mailbox on the mail server.
- The client opens a connection to the server. It then sends its user name and password to access the mailbox.
- The user can then list and retrieve the mail messages, one by one.

- POP3 Progress through 3 phases:
  - **User authorization**(the user agent sends a username and a password to authenticate the user.)
  - **Transaction**(the user agent retrieves messages)
  - **Update**(occurs after the client has issued the quit command, ending the POP3 session)
- The **authorization phase** has two principal commands:
  - user <username>
  - pass <password>

POP 3

- POP works by downloading your emails from your provider's mail server, and then marking them for deletion there.
- This means you can only ever read those email messages in that email client, and on that computer.
- You won't be able to access any previously downloaded emails from any other device, with any other email client, or through webmail.
- Because POP doesn't sync folders, sent emails are only available in the email client where the message was sent.
- Emails will not show up on the server or other email clients connected to the mailbox.

- In a POP3 transaction, the user agent issues commands, and the server responds to each command with a reply.

- There are two possible responses**: +OK and -ERR**

- **+OK:** used by the server to indicate that the previous command was fine.

- **-ERR:** used by the server to indicate that something was wrong with the previous command.

- POP3 is **deficient** in several ways.
  - It does not allow the user to organize her mail on the server; the user cannot have different folders on the server. (Of course, the user can create folders on her own computer.)
  - In addition, POP3 does not allow the user to partially check the contents of the mail before downloading.
  - “By using POP, You can download emails from mail server to your PC .After downloading the original mail is removed from the server. So you cant access it from another computer.”

- **IMAP4 provides the following extra functions:**
  - A user can check the e-mail header prior to downloading.
  - A user can search the contents of the e-mail for a specific string of characters prior to downloading.
  - A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.
  - A user can create, delete, or rename mailboxes on the mail server.
  - A user can create a hierarchy of mailboxes in a folder for e-mail storage.

IMAP

- IMAP allows you to access your emails from any client, on any device, and sign in to webmail at any time, until you delete them.

- You'll always see the same emails, no matter how you access your provider's server.

- The primary difference is that POP downloads emails from the server for permanent local storage, while IMAP leaves them on the server while caching (temporarily storing) emails locally. In this way, IMAP is effectively a form of cloud storage.

Application Layer

- The application layer provides services to the user.

- Communication is provided using a **logical connection**, which means that the two application layers assume that there is an imaginary direct connection through which they can send and receive messages.

Network Application Architectures

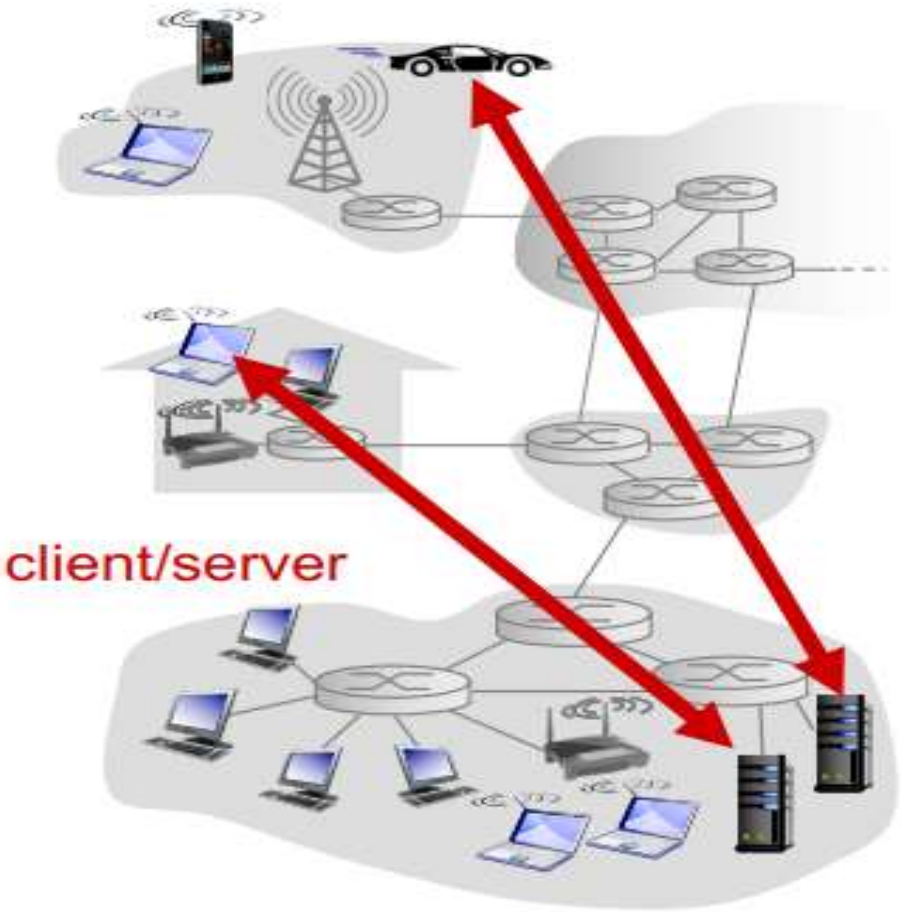
- Application's architecture is distinctly different from the network architecture (e.g., the five-layer Internet architecture discussed).
- From the application developer's perspective, the network architecture is fixed and provides a specific set of services to applications.
- **The application architecture, on the other hand, is designed by the application developer and dictates how the application is structured over the various end systems**.
- The two architectural paradigms used in modern network applications:
    - **The client-server architecture**
    - **The peer-to-peer (P2P) architecture**
        - **Hybrid Architecture (Client-Server and P2P)**

1. The client-server architecture

- **Client**: service requester ,sometimes-on or always-on.
- **Server**: Service provider , always-on.
- When a Web server receives a request for an object from a client host, it responds by sending the requested object to the client host.
- First characteristics of client-server architecture is **clients do not directly communicate with each other.**
- Second characteristics is **server has a fixed, well-known address, called an IP address.**
- Because the server has a fixed, well-known address, the server is always on.
- A client can always contact the server by sending a packet to the server's IP address.
- In the client-server architecture ,server operates as a centralized system.

1. The client-server architecture

Client-server architecture



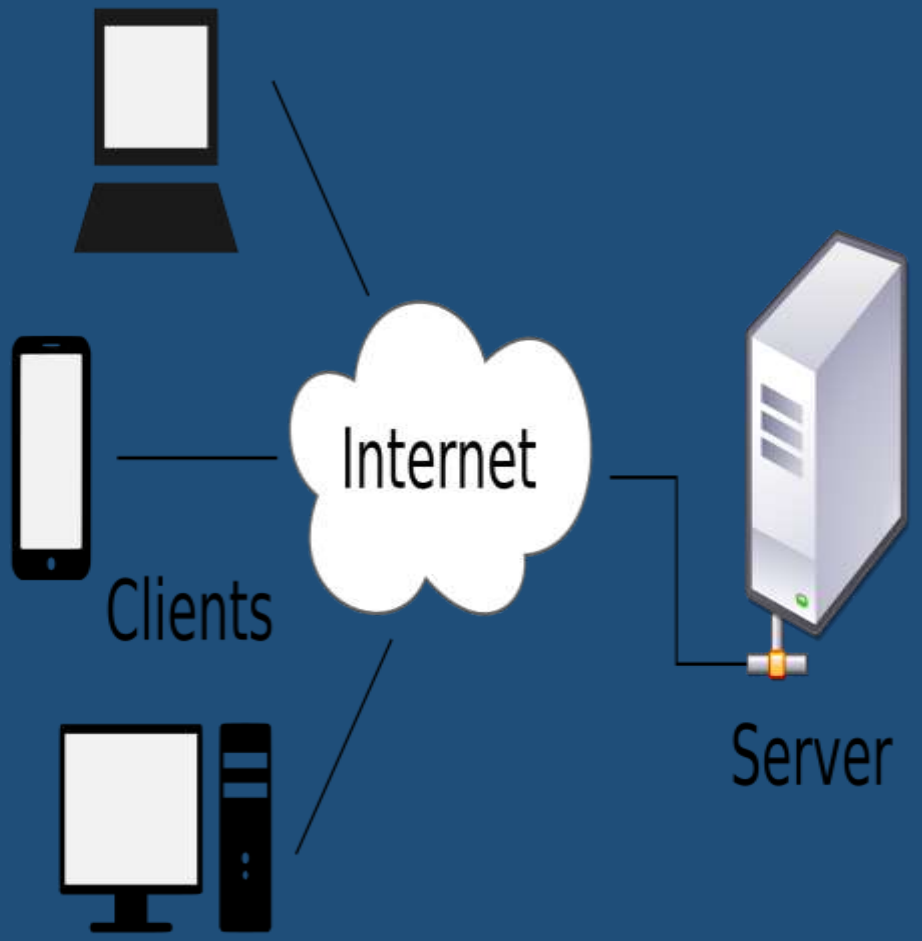
server:

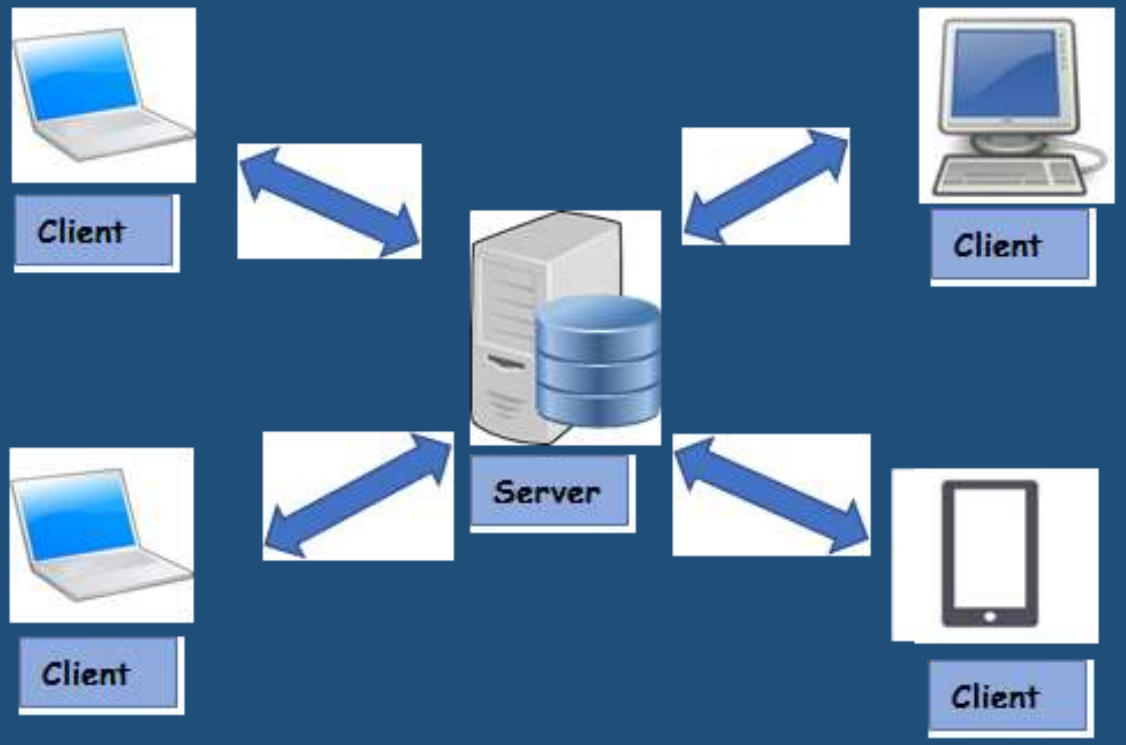
- always-on host
- permanent IP address
- data centers for scaling

clients:

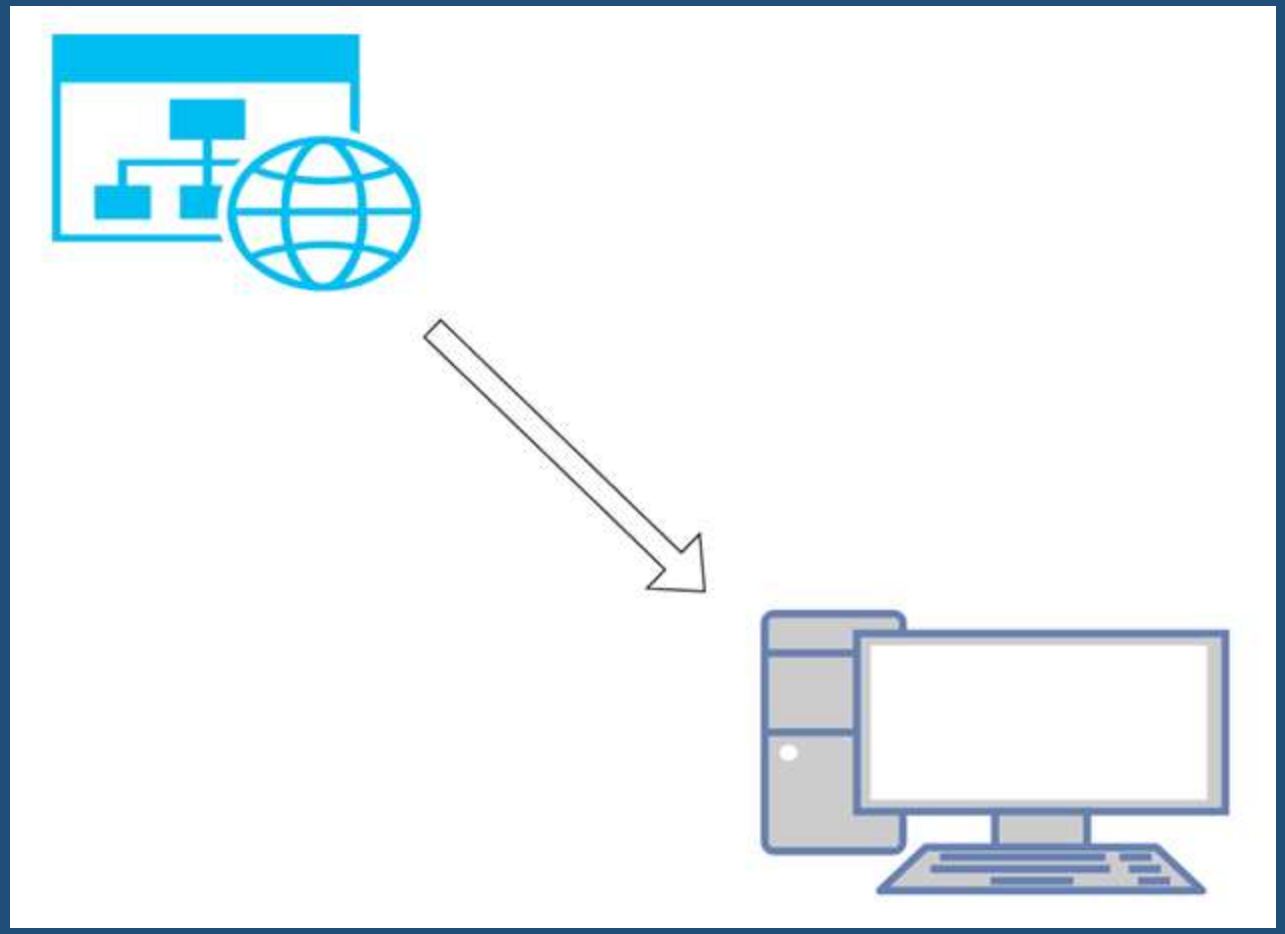
- communicate with server
- may be intermittently connected
- may have dynamic IP addresses
- do not communicate directly with each other

1. The client-server architecture





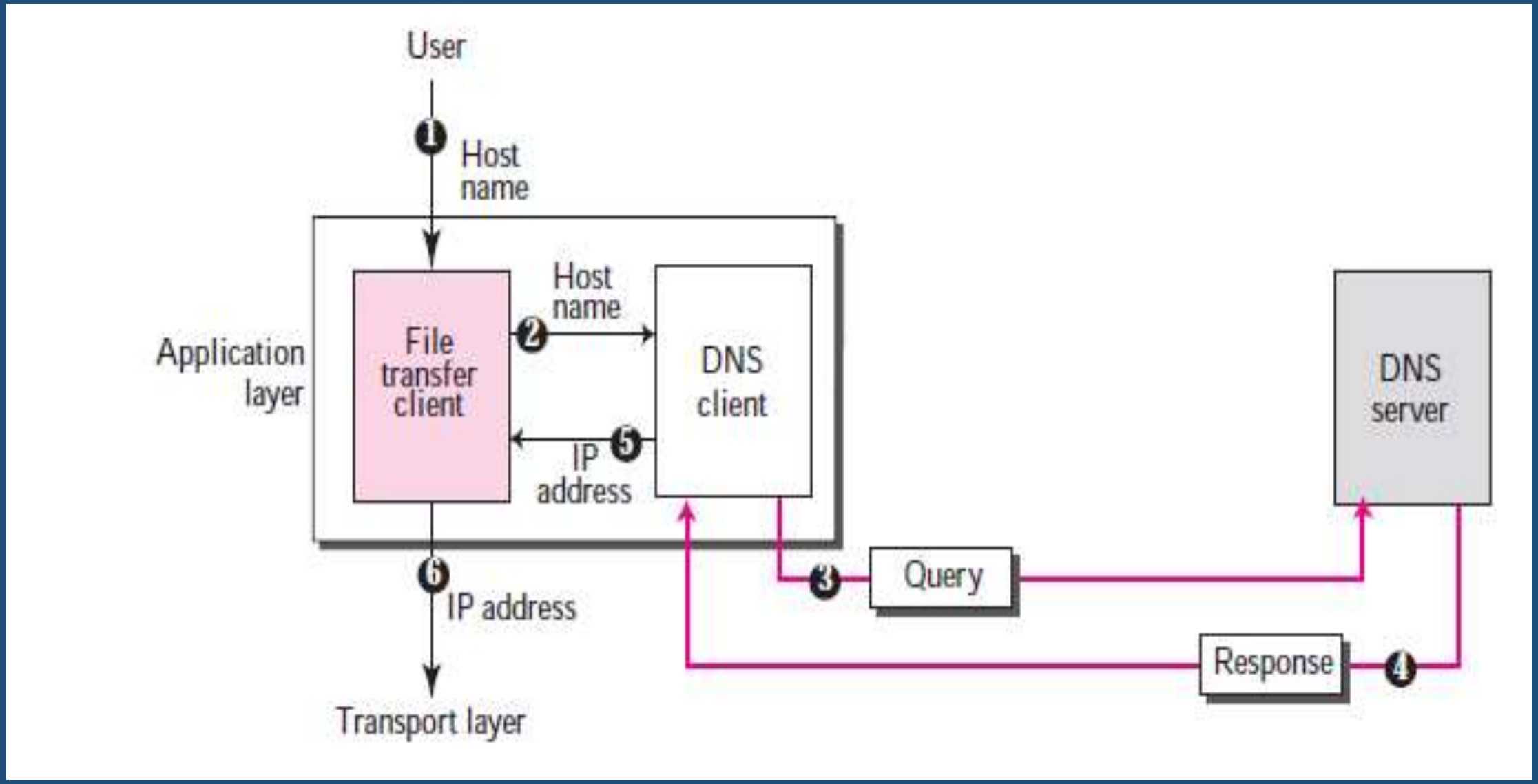
1. The client-server architecture



Domain Name System(DNS)

- There are two ways to identify a host—by a **hostname** and by an **IP address**.
- **People** prefer the **more mnemonic hostname identifier**, while **routers** prefer **fixed-length, hierarchically structured IP addresses.**
- In order to reconcile these preferences, **we need a directory service that translates hostnames to IP addresses.**
- This is the main task of the Internet's **domain name system (DNS)**.

DNS


User
1
Host name
Application layer
File transfer client
2
Host name
DNS client
5
IP address
DNS server
6
IP address
3
Query
Response
4
Transport layer

- The DNS is :

  (1) A distributed database implemented in a hierarchy of **DNS servers**.

  (2) An application-layer protocol that allows hosts to query the distributed database.

- As an example, consider what happens when a browser (that is, an HTTP client), running on some user's host, requests the URL www.someschool.edu/index.html. In order for the user's host to be able to send an HTTP request message to the Web server www.someschool.edu, the user's host must first obtain the IP address of www.someschool.edu. This is done as follows.

1. The same user machine runs the client side of the DNS application.

2. The browser extracts the hostname, www.someschool.edu, from the URL and passes the hostname to the client side of the DNS application.

3. The DNS client sends a query containing the hostname to a DNS server.

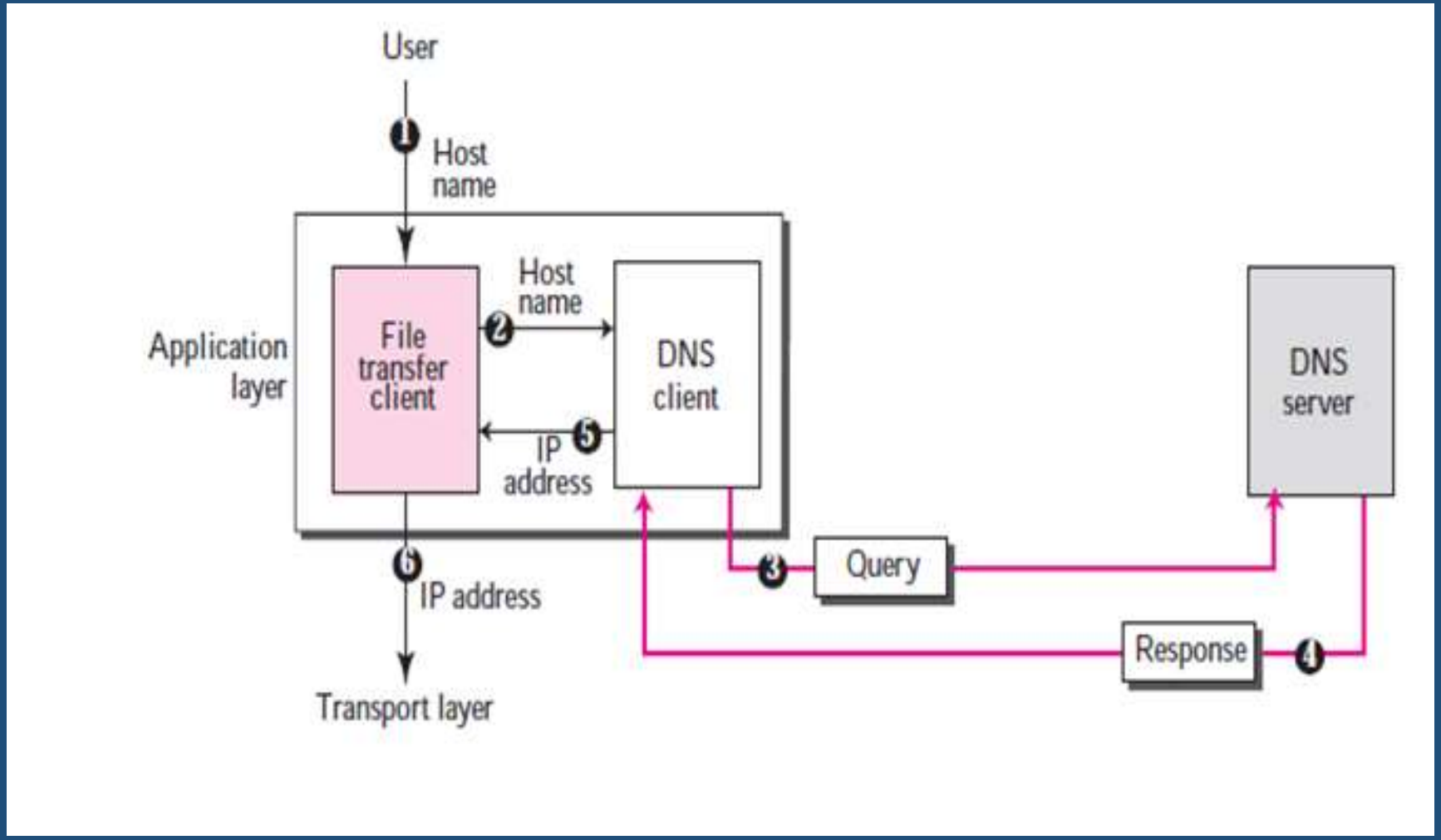
4. The DNS client eventually receives a reply, which includes the IP address for the hostname.

5. Once the browser receives the IP address from DNS, it can initiate a TCP connection to the HTTP server process located at port 80 at that IP address.

Services of DNS

- Translating hostname to IP address
- Host aliasing
- Mail server aliasing
- Load distribution

1. Translating hostname to IP address



2. Host aliasing.

- A host with a complicated hostname can have one or more alias names. For example, a hostname such as relay1.westcoast.enterprise.com could have, say, two aliases such as enterprise.com and www.enterprise.com.

- In this case, the hostname relay1.westcoast.enterprise.com is said to be a **canonical hostname**.

- Alias hostnames, when present, are typically more mnemonic than canonical hostnames.

- DNS can be invoked by an application to obtain the canonical hostname for a supplied alias hostname as well as the IP address of the host.

Mail server aliasing

It is highly desirable that e-mail addresses be mnemonic.

- For example, if Bob has an account with Hotmail, Bob's e-mail address might be as simple as bob@hotmail.com.
- However, the hostname of the Hotmail mail server is more complicated and much less mnemonic than simply hotmail.com (for example, the canonical hostname might be something like relay1.west-coast.hotmail.com).
- DNS can be invoked by a mail application to obtain the canonical hostname for a supplied alias hostname as well as the IP address of the host.

Load distribution

- DNS is used to perform load distribution among replicated servers.
- Eg: busy sites such as cnn.com are replicated over multiple servers.
- For replicated web servers , set of IP address is associated with one canonical name.
- DNS database contains this set of IP address.

Why not centralized DNS ?

- In centralized design, clients simply direct all queries to the single DNS server.
- This design is inappropriate for today's internet, with its growing no.of hosts.
- The problem that arise when we try to centralize DNS are:
  - **Single point of failure** : If the DNS server crashes, so does the entire Internet
  - **Traffic Volume**: A single DNS server would have to handle all DNS queries (for all the HTTP requests and e-mail messages generated from hundreds of millions of hosts).
  - **Distant centralized database**: A single DNS server cannot be “close to” all the querying clients. If we put the single DNS server in New York City, then all queries from Australia must travel to the other side of the globe, perhaps over slow and congested links. This can lead to significant delays.
  - **Maintenance** : The single DNS server would have to keep records for all Internet hosts. Not only would this centralized database be huge, but it would have to bt updated frequently to account for every new host.

A Distributed, Hierarchical Database or Distributed DNS

- The DNS uses a large number of servers, organized in a hierarchical fashion and distributed around the world.

- No single DNS server has all of the mappings for all of the hosts in the Internet. Instead, the **mappings are distributed across the DNS servers.**

- **There are three classes of DNS servers**:
    - Root DNS servers
    - Top-level domain (TLD) DNS servers
    - Authoritative DNS servers

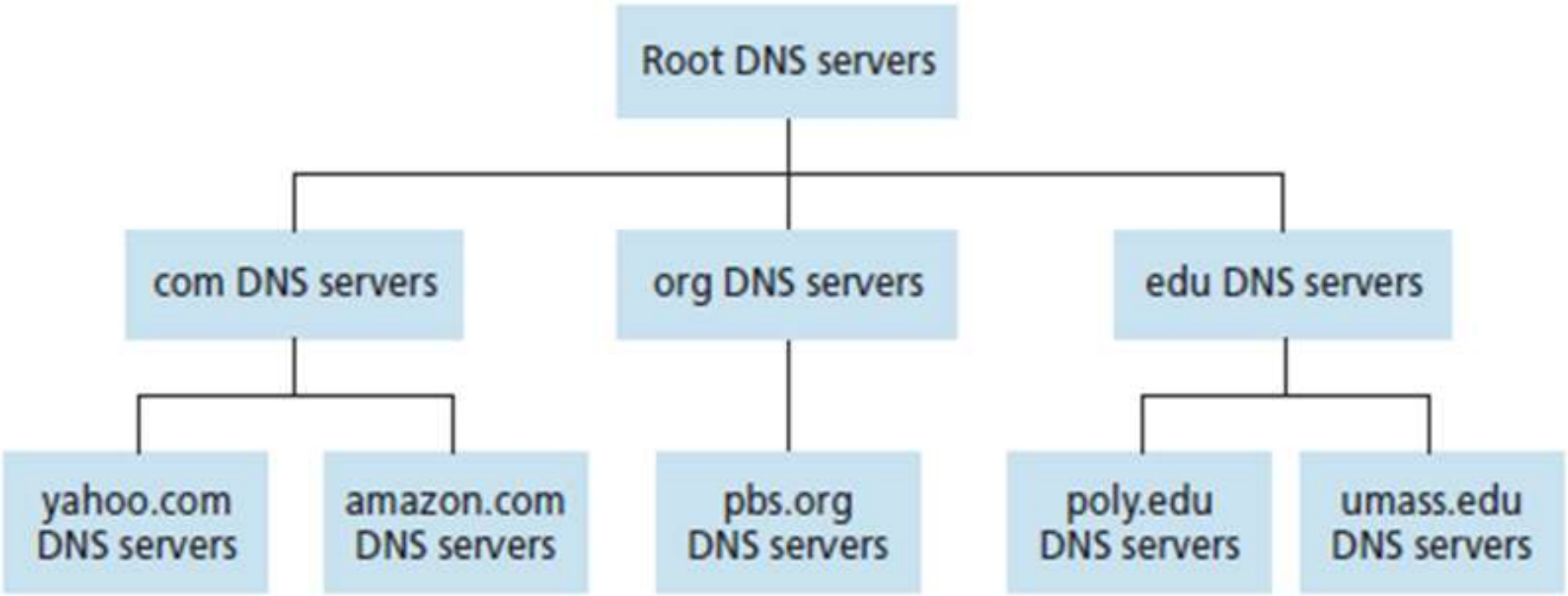


Figure 2.19 ♦ Portion of the hierarchy of DNS servers

- To understand how these three classes of servers interact, suppose a DNS client wants to determine the IP address for the hostname www.amazon.com.

- To a first approximation, the following events will take place.

  - The client first contacts one of the root servers, which returns IP addresses for TLD servers for the top-level domain com.

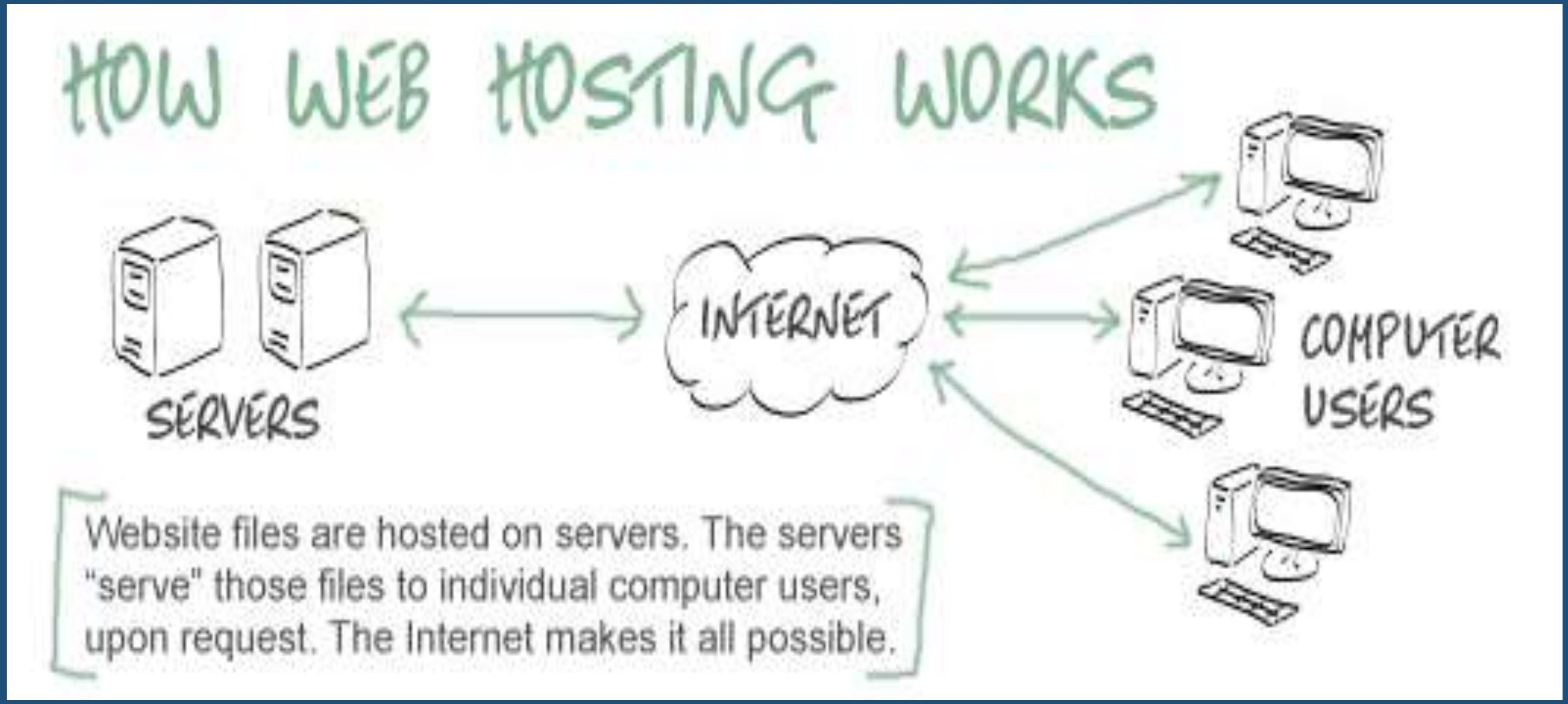
  - The client then contacts one of these TLD servers, which returns the IP address of an authoritative server for amazon.com.

  - Finally, the client contacts one of the authoritative servers for amazon.com, which returns the IP address for the hostname www.amazon.com.

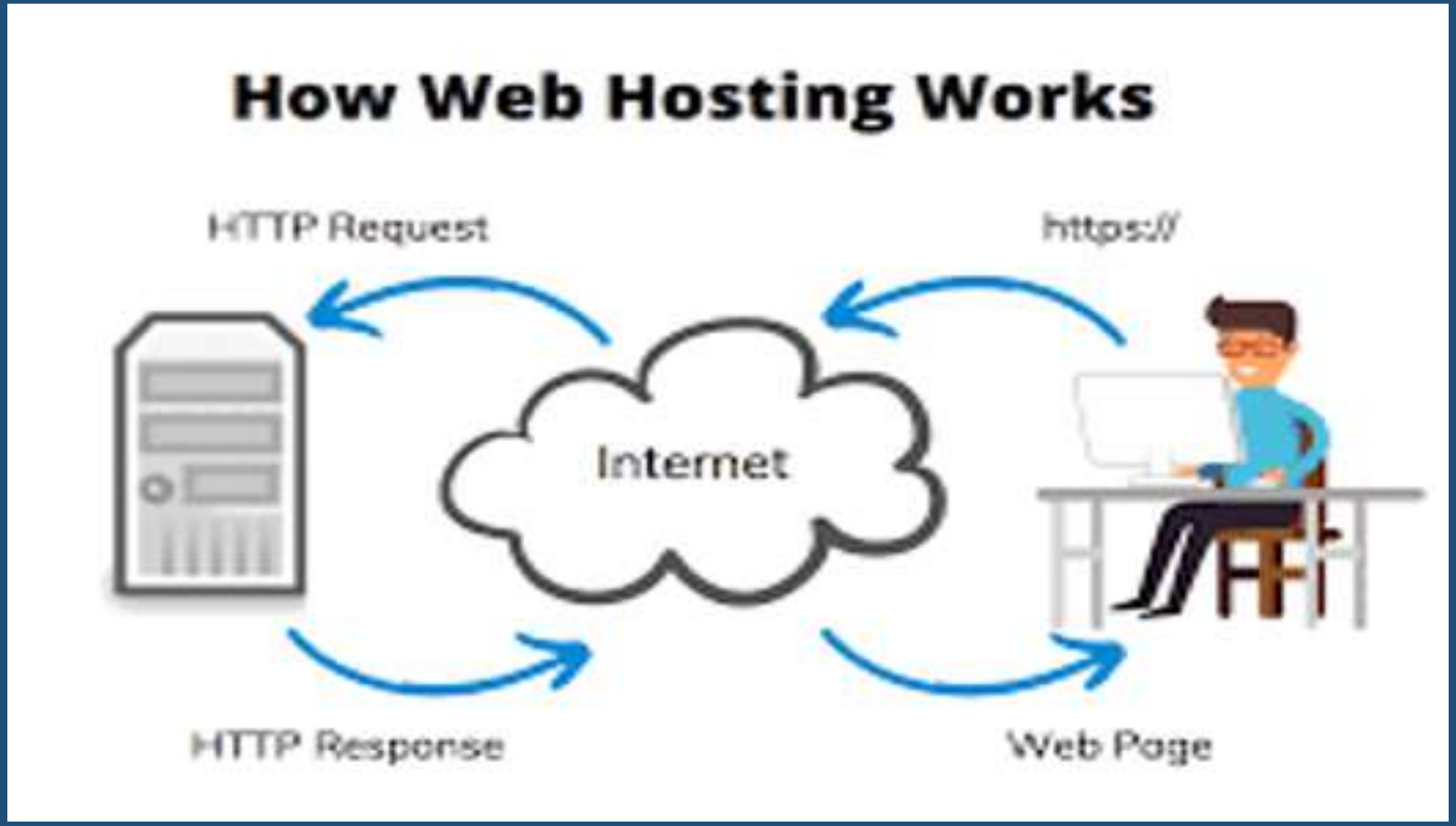
The web and HTTP

- The Web is the common name for the World Wide Web, a subset of the Internet consisting of the pages that can be accessed by a Web browser.
- The World Wide Web—commonly referred to as WWW, W3, or the Web—is **an interconnected system of public webpages accessible through the Internet**.
- The Web is not the same as the Internet: the Web is one of many applications built on top of the Internet.
- Browsers such as Internet Explorer, Google Chrome, Opera or Mozilla Firefox are used to access Web documents, or Web pages, which are connected via links.
- Web pages are formatted in a language called Hypertext Markup Language (HTML). It this language that allows users to click through pages on the Web via links.
- The Web uses **HTTP protocol to transmit data and share information**.

The web and HTTP



The web and HTTP



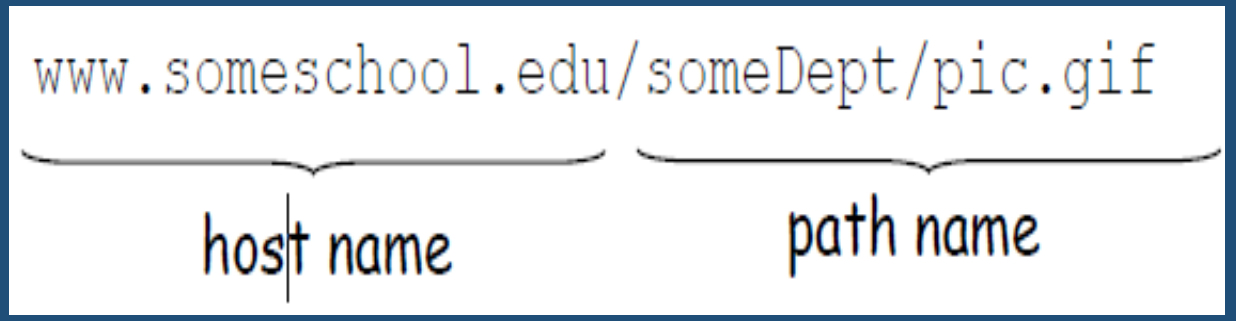
Website - A group of interlinked and well structured Web pages that exist on the same domain is called website.

Webpage- It is a collection of text, image, audio, video etc.

Website can contain single or thousand pages.

The web and HTTP

- Web page consists of objects.
- Object can be HTML file, JPEG image, Java applet, audio file,…
- Web page consists of base HTML-file which includes several referenced objects.
- Each object is addressable by a URL.
- Example of URL:



- URL consist of two components: **hostname and pathname**
- URL is the combination of hostname and pathname

HTTP overview

HTTP: hypertext transfer protocol

- Web's application layer protocol
- client/server model
  - *client:* browser that requests, receives, (using HTTP protocol) and "displays" Web objects
  - *server:* Web server sends (using HTTP protocol) objects in response to requests

HTTP overview (continued)

uses TCP:

- client initiates TCP connection (creates socket) to server, port 80
- server accepts TCP connection from client
- HTTP messages (application-layer protocol messages) exchanged between browser (HTTP client) and Web server (HTTP server)
- TCP connection closed

HTTP is "stateless"

- server maintains no information about past client requests

aside

protocols that maintain "state" are complex!

- past history (state) must be maintained
- if server/client crashes, their views of "state" may be inconsistent, must be reconciled

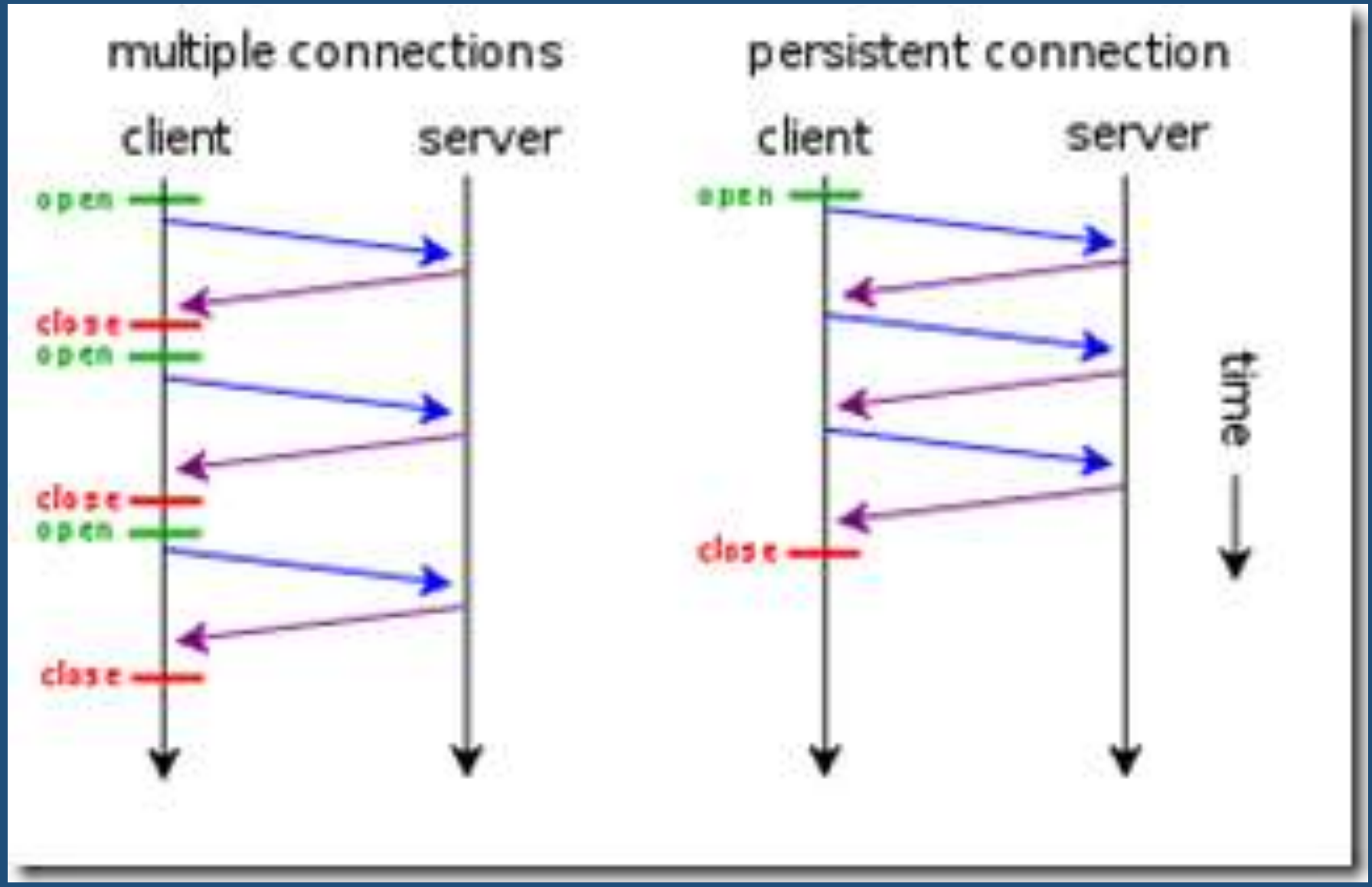
HTTP connections

non-persistent HTTP

- at most one object sent over TCP connection
  - connection then closed
- downloading multiple objects required multiple connections

persistent HTTP

- multiple objects can be sent over single TCP connection between client, server


multiple connections
client
server
open
close
open
close
open
close
persistent connection
client
server
open
close
time

Non-persistent HTTP

suppose user enters URL:
`www.someSchool.edu/someDepartment/home.index`

(contains text, references to 10 jpeg images)

1a. HTTP client initiates TCP connection to HTTP server (process) at www.someSchool.edu on port 80

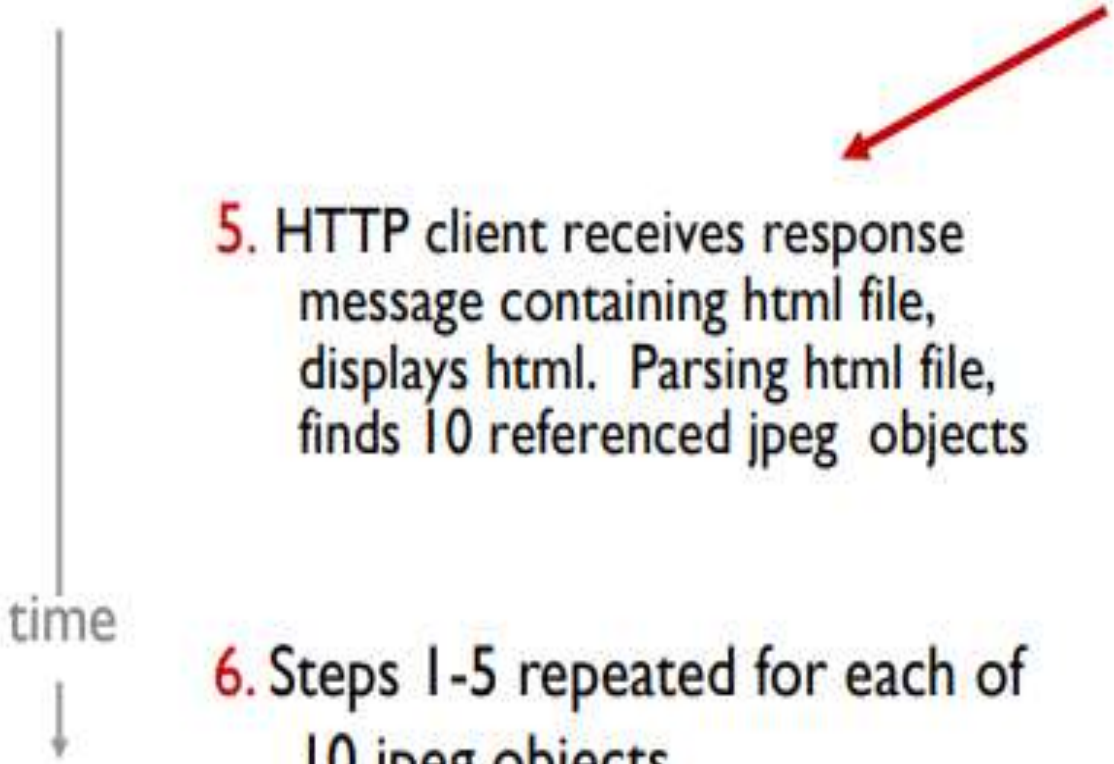
1b. HTTP server at host www.someSchool.edu waiting for TCP connection at port 80. "accepts" connection, notifying client

2. HTTP client sends HTTP *request message* (containing URL) into TCP connection socket. Message indicates that client wants object someDepartment/home.index

3. HTTP server receives request message, forms *response message* containing requested object, and sends message into its socket

time

Non-persistent HTTP (cont.)



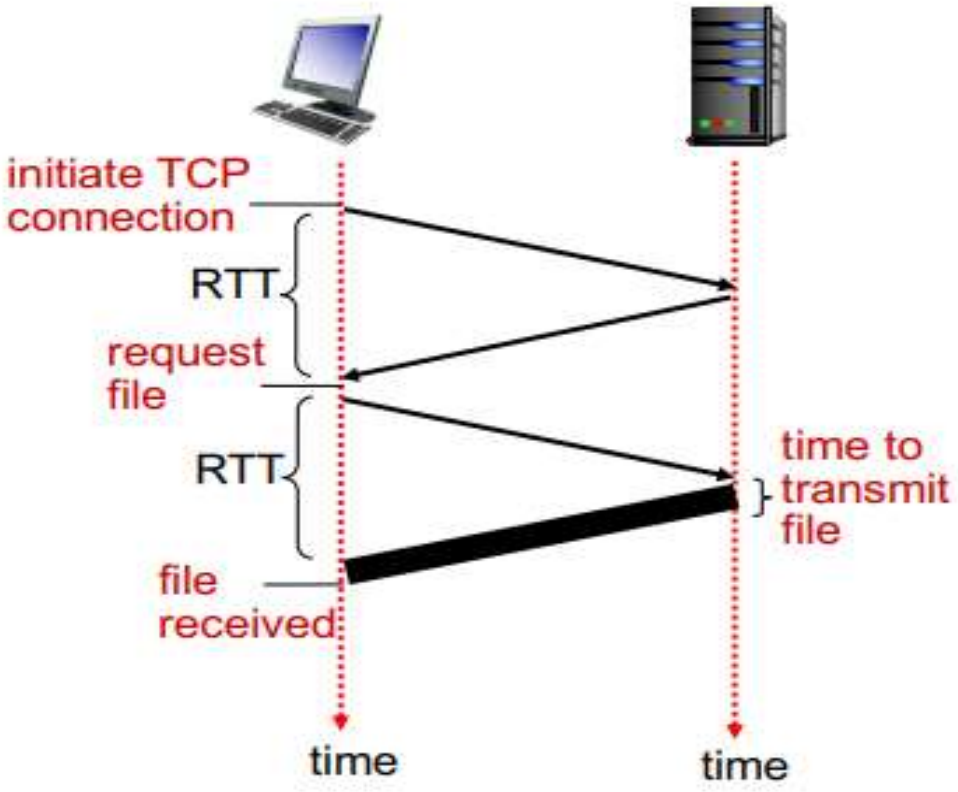
Non-persistent HTTP: response time

RTT (definition): time for a small packet to travel from client to server and back

HTTP response time:

- one RTT to initiate TCP connection
- one RTT for HTTP request and first few bytes of HTTP response to return
- file transmission time
- non-persistent HTTP response time =

  2RTT+ file transmission time



Persistent HTTP

non-persistent HTTP issues:

- requires 2 RTTs per object
- OS overhead for *each* TCP connection
- browsers often open parallel TCP connections to fetch referenced objects

persistent HTTP:

- server leaves connection open after sending response
- subsequent HTTP messages between same client/server sent over open connection
- client sends requests as soon as it encounters a referenced object
- as little as one RTT for all the referenced objects

FTP

- FTP stands for File transfer protocol.
- FTP is a standard internet protocol provided by TCP/IP used for transmitting the files from one host to another.
- It is mainly used for transferring the web page files from their creator to the computer that acts as a server for other computers on the internet.
- It is also used for downloading the files to computer from other servers.
- FTP uses client-server architecture in servers with security features, username and password protection for file transfer.

FTP

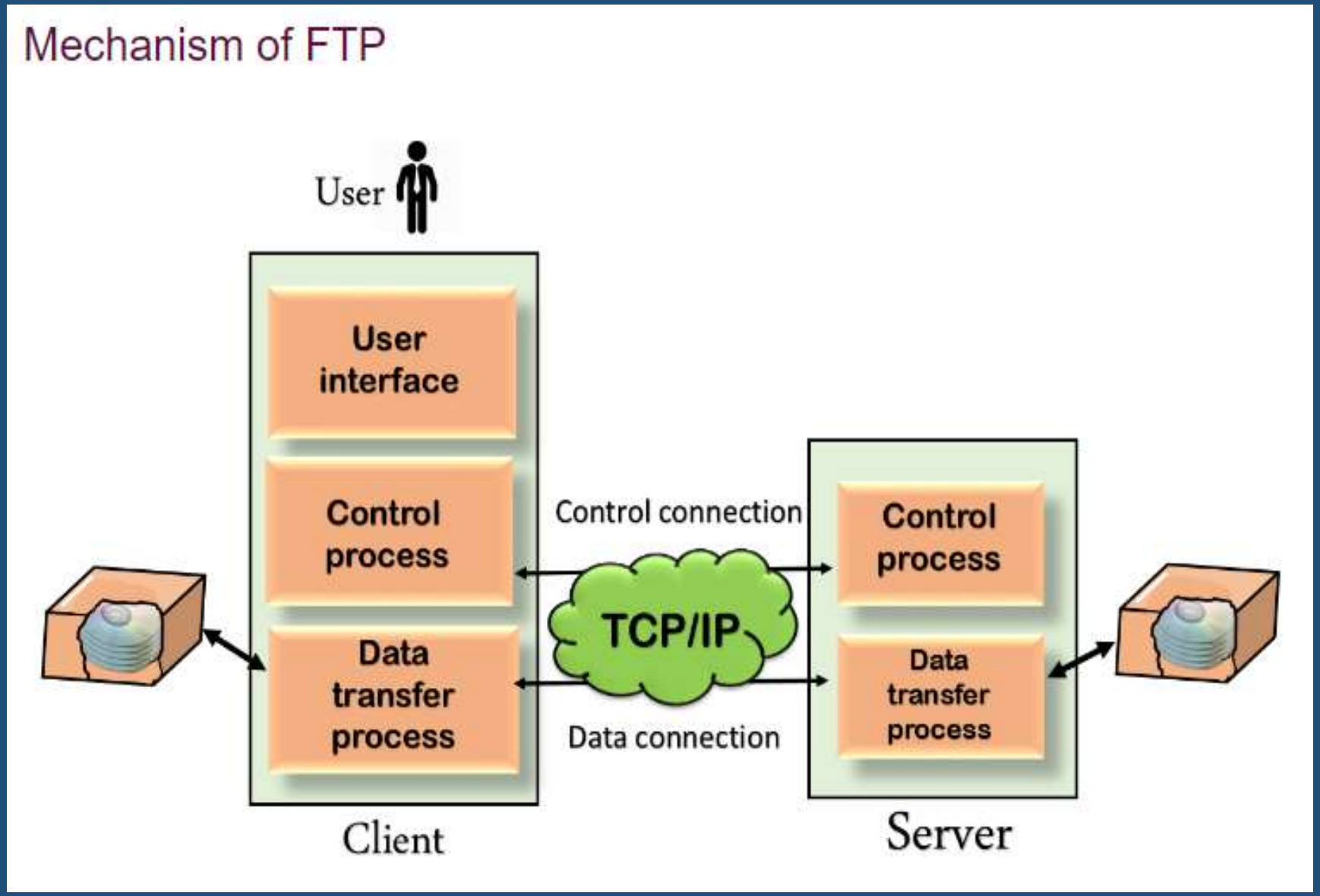
Objectives of FTP

- It provides the sharing of files.
- It is used to encourage the use of remote computers.
- It transfers the data more reliably and efficiently.

Why FTP?

- Although transferring files from one system to another is very simple and straightforward, but sometimes it can cause problems. For example, two systems may have different file conventions. Two systems may have different ways to represent text and data. Two systems may have different directory structures.
- FTP protocol overcomes these problems by establishing two connections between hosts. One connection is used for data transfer, and another connection is used for the control connection.

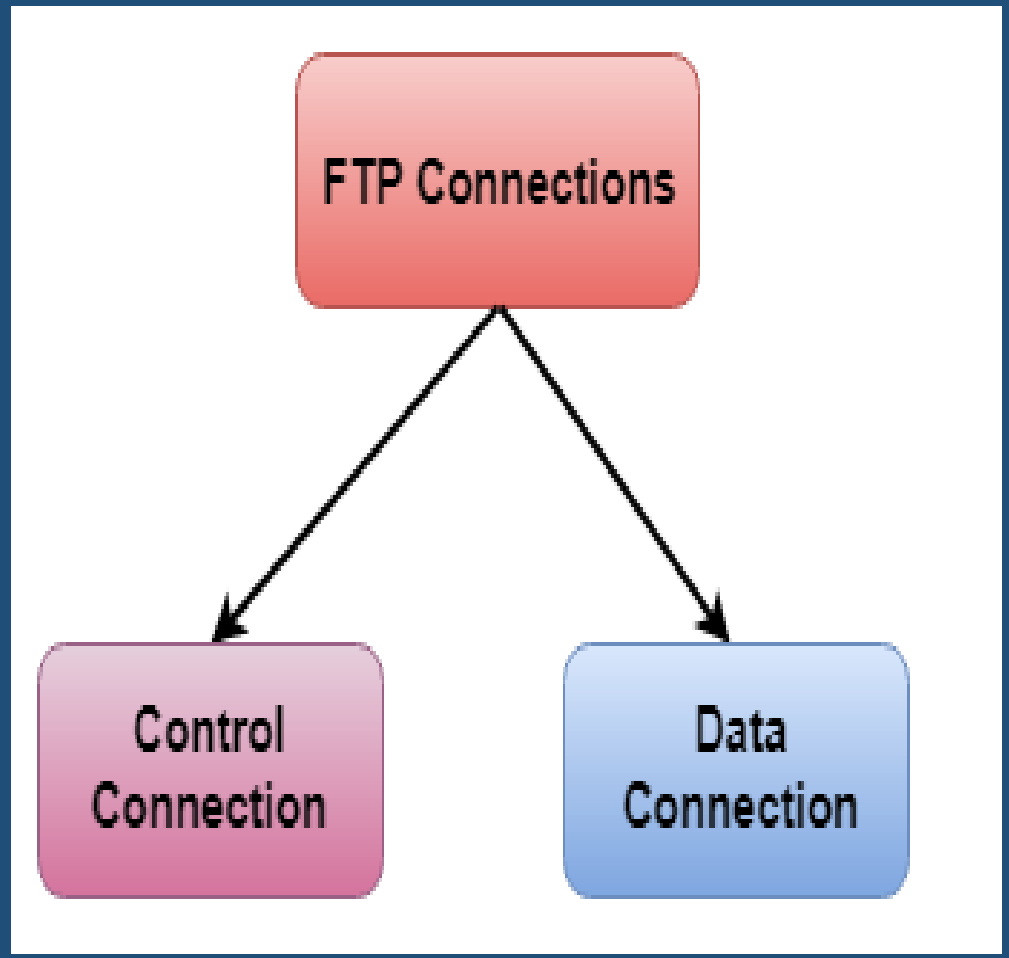
Basic Model of FTP



The above figure shows the basic model of the FTP. The FTP client has three components: the user interface, control process, and data transfer process. The server has two components: the server control process and the server data transfer process.

FTP CONNECTIONS

- **There are two types of connections in FTP:**
  - **Control Connection:** The control connection uses very simple rules for communication. Through control connection, we can transfer a line of command or line of response at a time. The control connection is made between the control processes. The control connection remains connected during the entire interactive FTP session.
  - **Data Connection:** The Data Connection uses very complex rules as data types may vary. The data connection is made between data transfer processes. The data connection opens when a command comes for transferring the files and closes when the file is transferred.



FTP Clients

- An FTP client is **an application on your computer that connects you to remote servers through FTP and other protocols**.
- An FTP client provides an environment in which you can upload files to a server, download files from a server to your device, and view and manage files stored on your web server.
- It allows a user to connect to a remote host and upload or download the files.
- It has a set of commands that we can use to connect to a host, transfer the files between you and your host and close the connection.
- The FTP program is also available as a built-in component in a Web browser. This GUI based FTP client makes the file transfer very easy and also does not require to remember the FTP commands.

Advantages of FTP:

- **Speed:** One of the biggest advantages of FTP is speed. The FTP is one of the fastest way to transfer the files from one computer to another computer.
- **Efficient:** It is more efficient as we do not need to complete all the operations to get the entire file.
- **Security:** To access the FTP server, we need to login with the username and password. Therefore, we can say that FTP is more secure.
- **Back & forth movement:** FTP allows us to transfer the files back and forth. Suppose you are a manager of the company, you send some information to all the employees, and they all send information back on the same server.

Disadvantages of FTP:

- The standard requirement of the industry is that all the FTP transmissions should be encrypted. However, not all the FTP providers are equal and not all the providers offer encryption. So, we will have to look out for the FTP providers that provides encryption.
- FTP serves two operations, i.e., to send and receive large files on a network. However, the size limit of the file is 2GB that can be sent. It also doesn't allow you to run simultaneous transfers to multiple receivers.
- Passwords and file contents are sent in clear text that allows unwanted eavesdropping. So, it is quite possible that attackers can carry out the brute force attack by trying to guess the FTP password.
- It is not compatible with every system.

THE END