Syllabus

Module I (10 Hours) Syllabus

- Overview of Computer Networks and the Internet. History. Protocols, Review of last mile technologies used for internet access. Packet switching. Basic ideas about delay queuing throughput. Concept of Quality of Service, Protocol layering. OSI model and TCP model
- Application layer protocols - Client-server architecture Network layer 7 application architecture, Web, HTTP, FTP, SMTP, POP3, and DNS, Peer-to-peer file sharing networks.

- Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education, 1 st Edition (2011).
- James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Overview of the Internet

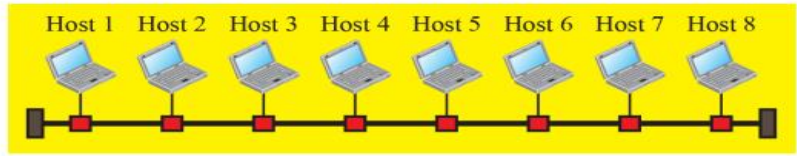
- Networks
  - A network is the interconnection of a set of devices capable of communication.
  - A device can be a host such as a large computer, desktop, laptop, workstation, cellular phone, or security system.
  - A device can also be a connecting device such as a router which connects the network to other networks, a switch which connects devices together, a modem (modulator-demodulator) that changes the form of data, and so on.
- Internetwork
  - A combination of networks.

Overview of the Internet

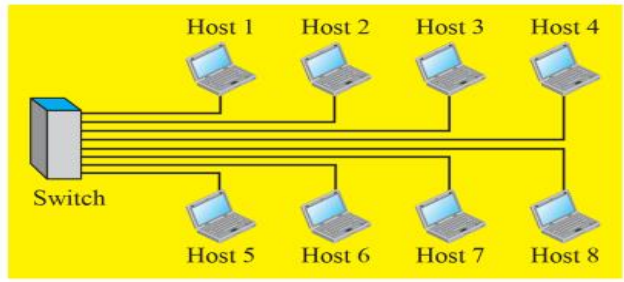
- Local Area Networks
  - Usually privately owned and connects some host in an office, building or campus. Figure
- Wide Area Networks
  - Wider geographical span
  - A LAN interconnects hosts; a WAN interconnects connecting devices such as switches, routers, or modems.
  - A LAN is normally privately owned by the organization that uses it; a WAN is normally created and run by communication companies and leased by an organization that uses it.
  - Point-to-Point WANs: connects two communicating devices through a transmission media. Figure
  - Switched WANs: network with more than two ends. Figure
- Internetwork(internet)
  - Two or more networks are connected Figure

Overview of the Internet

Figure 1.1: An Isolated LAN in the past and today



a. LAN with a common cable (past)



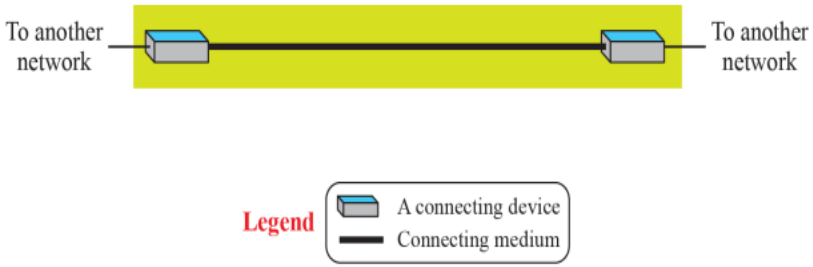
b. LAN with a switch (today)

Back

[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

Overview of the Internet

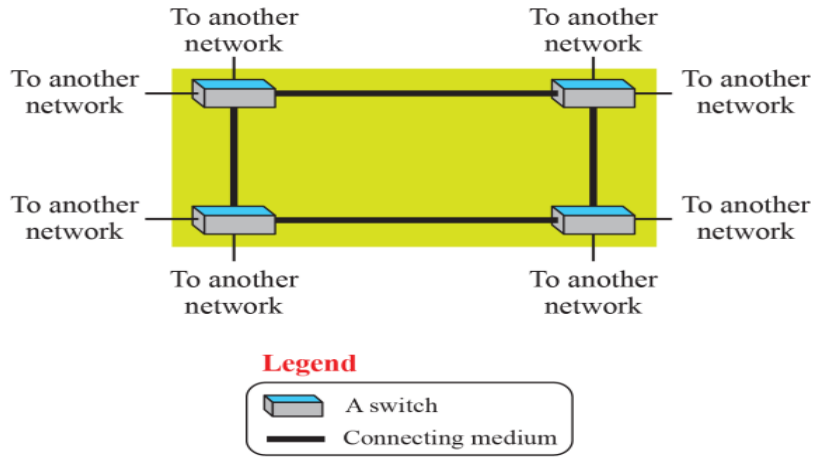
Figure 1.2: A Point-to-Point WAN



Back

Overview of the Internet

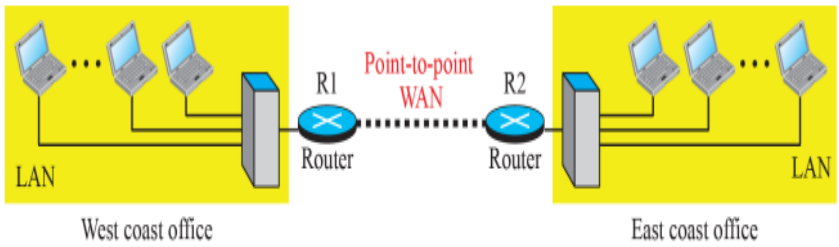
Figure 1.3: A Switched WAN



Back

Overview of the Internet

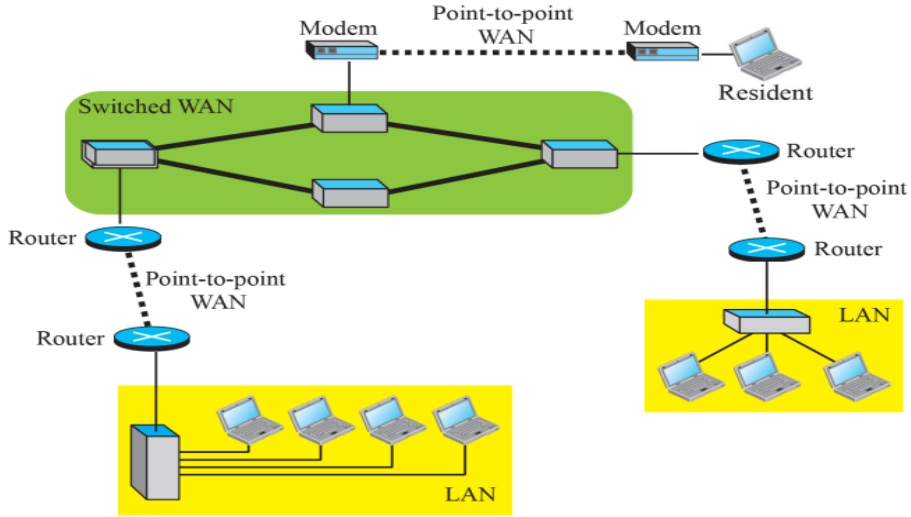
Figure 1.4: An internetwork made of two LANs and one WAN



Back

Overview of the Internet

Figure 1.5: A heterogeneous network made of WANs and LANs



Back

Overview of the Internet

Switching

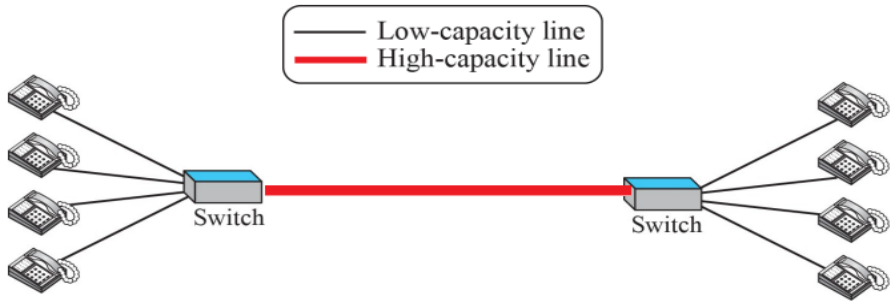
- An internet is a switched network in which a switch connects at least two links together.
- A switch needs to forward data from a link to another link when required.
  - Circuit-Switched Network
  - Packet-Switched Network

Overview of the Internet

Circuit-switched network

- A dedicated connection, called circuit, is always available between the two end systems; the switch can only make it active or inactive. Eg: Telephone networks
- Switches have only forwarding capability.

Figure 1.6: A circuit-switched network



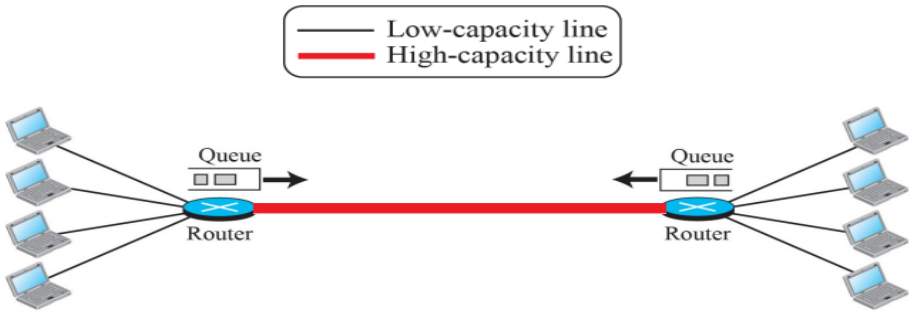
[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

Overview of the Internet

Packet-switched network

- Communication between the two ends is done in blocks of data called packets.
- Switches have forwarding and storing capability.

Figure 1.7: A packet-switched network



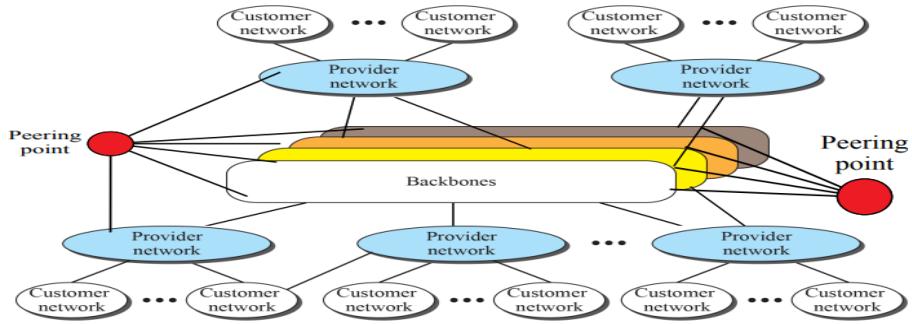
[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

Overview of the Internet

The Internet

- The most notable internet is called the Internet and is composed of thousands of inter-connected networks.
- Figure1.8 shows a conceptual (not geographical) view of the Internet.

Figure 1.8: The Internet today



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

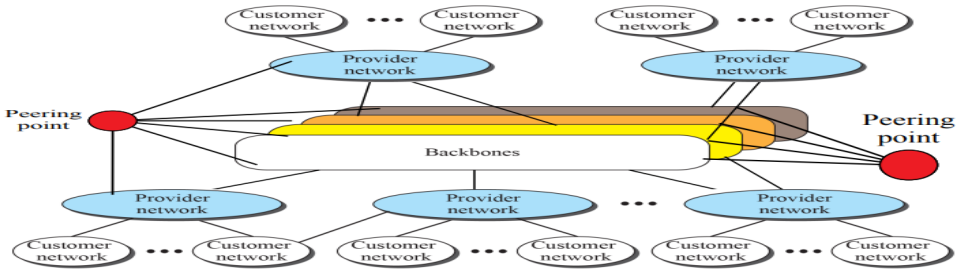
Overview of the Internet

The Internet

- Backbones:- large networks owned by communication companies like Verizon, Sprint, AT&T etc. These are connected through some complex switching systems, called peering points.
- Provider Network:- use the services of the backbones for a fee.
- Customer Network:- networks at the edge of the Internet that actually use the services provided by the Internet.

ISPs

Figure 1.8: The Internet today



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

Overview of the Internet

Accessing the Internet

- The Internet today is an internetwork that allows any user to become part of it.
- The user needs to be physically connected to an ISP.
- To connect to the Internet is to change the voice line between the residence or business and the telephone center to a point-to-point WAN
- Using Telephone Networks
  - Dial-up Service: Add to the telephone line a modem that converts data to voice.
  - DSL: Telephone companies have upgraded their telephone lines to provide higher speed Internet services to residences or small businesses. DSL service also allows the line to be used simultaneously for voice and data communication.
- Using Cable Networks
- Using Wireless Networks
- Direct Connection

Overview of the Internet

Accessing the Internet

- This is the overview of the Internet structure.
- For communication to happen, we need both hardware and software.
- This is similar to a complex computation in which we need both a computer and a program.
- In the next section, we learn how these combinations of hardware and software are coordinated with each other using protocol layering.

PROTOCOL LAYERING

- A protocol defines the rules that both the sender and receiver and all intermediate devices need to follow to be able to communicate effectively.
- When communication is simple, we may need only one simple protocol; when the communication is complex, we need a protocol at each layer, or protocol layering.

PROTOCOL LAYERING

Scenarios

- Let us develop two simple scenarios to better understand the need for protocol layering.
  - First Scenario (Figure 1.9)
  - Second Scenario (Figure 1.10)
- Principle of Protocol Layering
  - Enables us to divide a complex task into several smaller and simpler tasks.
  - Modularity: means independent layers and a layer (module) can be defined as a black box with inputs and outputs, without concern about how inputs are changed to outputs.
  - Advantage: Allows us to separate the services from the implementation, Communication does not always use only two end systems; there are intermediate systems that need only some layers, but not all layers(less expensive).
  - Disadvantage: having a single layer makes the job easier

NEXT

PROTOCOL LAYERING

- First Scenario (Figure 1.9)

Figure 1.9: A single-layer protocol

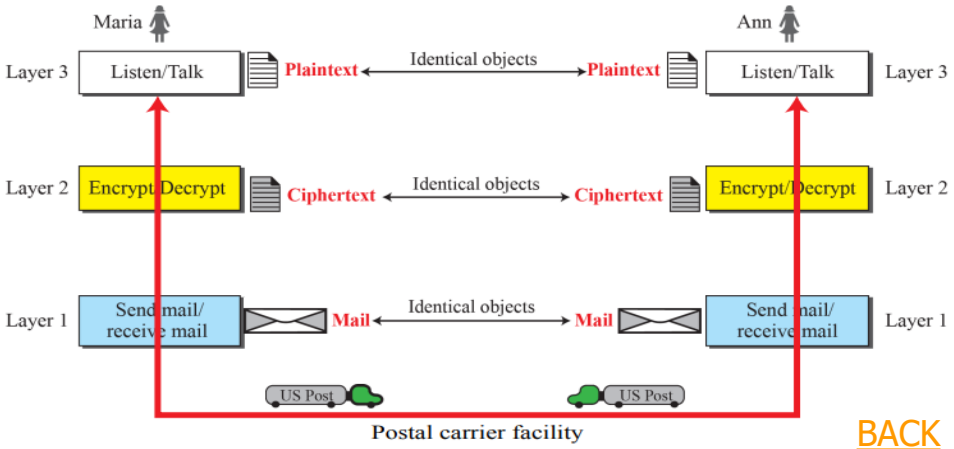
BACK

[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

PROTOCOL LAYERING

- Second Scenario (Figure 1.10)

Figure 1.10: A three-layer protocol



BACK

[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

PROTOCOL LAYERING

Principles of Protocol Layering

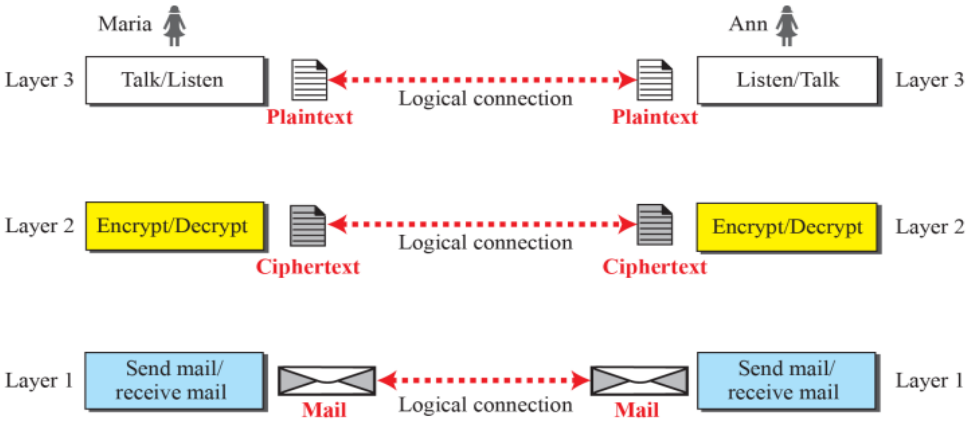
- The first principle dictates that if we want bidirectional communication, we need to make each layer so that it is able to perform two opposite tasks, one in each direction (Eg: encrypt/decrypt).
- The second principle that we need to follow in protocol layering is that the two objects under each layer at both sites should be identical (object under layer 3 at both sites should be a plaintext letter).

PROTOCOL LAYERING

Logical(imaginary) Connections

- This means that we have layer-to-layer communication.

Figure 1.11: Logical connection between peer layers



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

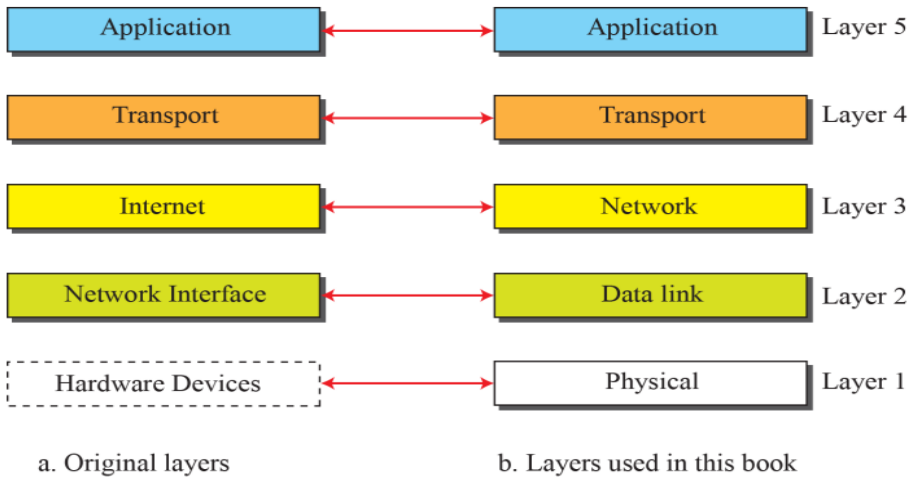
PROTOCOL LAYERING

TCP/IP (Transmission Control Protocol/Internet Protocol) Protocol Suite

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

PROTOCOL LAYERING

Figure 1.12: Layers in the TCP/IP protocol suite



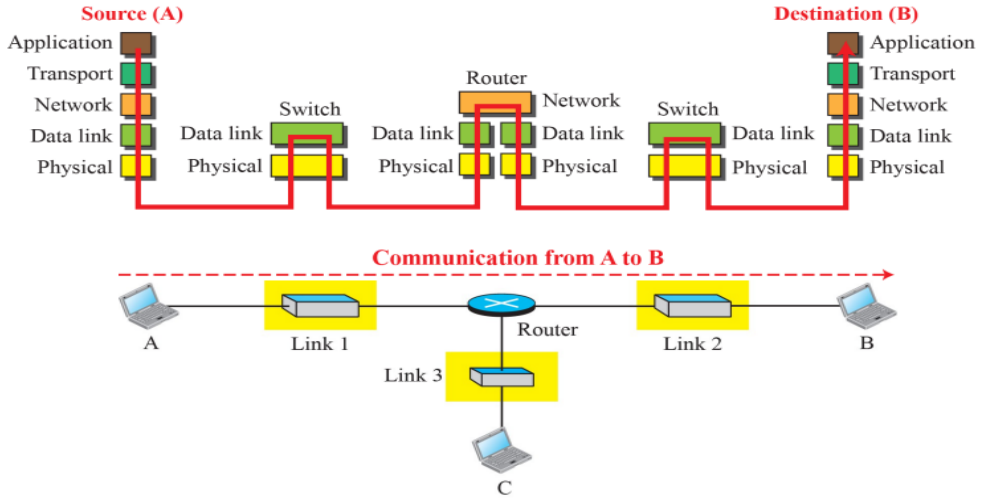
[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

PROTOCOL LAYERING

TCP/IP Protocol Suite

- Layered Architecture
  - To show how the layers in the TCP/IP protocol suite are involved in communication between two hosts, we assume that we want to use the suite in a small internet made up of three LANs (links), each with a link-layer switch.
  - The router is involved only in three layers; there is no transport or application layer in a router as long as the router is used only for routing.
  - A link-layer switch in a link, however, is involved only in two layers, data-link and physical.

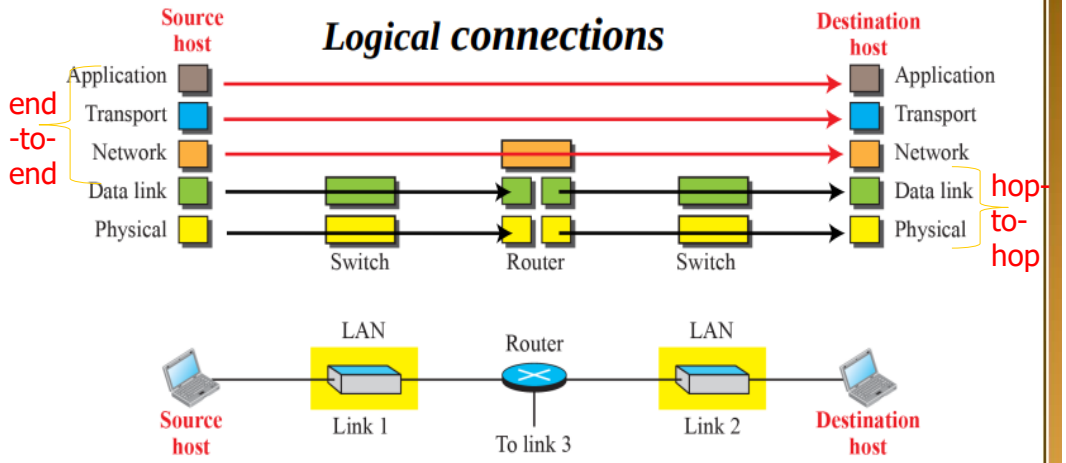
Figure 1.13: Communication through an internet



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

- Layers in the TCP/IP Protocol Suite

Figure 1.14: Logical connections between layers in TCP/IP



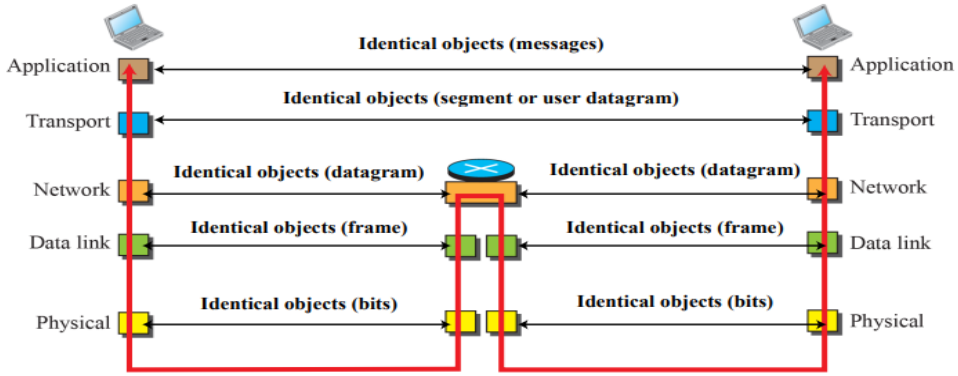
[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

Layers in the TCP/IP Protocol Suite

- The duty of the application, transport and network layers is end-to-end.
- The duty of the data-link and physical layers is hop-to-hop, in which a hop is a host or router.
- The domain of duty of the top three layers is the internet, and the domain of duty of the two lower layers is the link.
- In the top three layers, the data unit (packets) should not be changed by any router or link-layer switch.
- In the bottom two layers, the packet created by the host is changed only by the routers, not by the link-layer switches.

Figure 1.15: Identical objects in the TCP/IP protocol suite

Notes: We have not shown switches because they don't change objects.



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

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

PROTOCOL LAYERING

TCP/IP Protocol Suite

Description of Each Layer in TCP/IP

- Application Layer
- Transport Layer
- Network Layer
- Data-link Layer
- Physical Layer

PROTOCOL LAYERING

TCP/IP Protocol Suite

Description of Each Layer in TCP/IP

Application Layer

- Logical connection between the two application layers is end to-end.
- The two application layers exchange messages between each other as though there were a bridge between the two layers.
- Communication at the application layer is between two processes (two programs running at this layer).

PROTOCOL LAYERING

TCP/IP Protocol Suite

Description of Each Layer in TCP/IP

Application Layer

- To communicate, a process sends a request to the other process and receives a response.
- Process-to-process communication is the duty of the application layer.
- The application layer in the Internet includes many predefined protocols, but a user can also create a pair of processes to be run at the two hosts.

PROTOCOL LAYERING

TCP/IP Protocol Suite

Description of Each Layer in TCP/IP

Application Layer

- The Hypertext Transfer Protocol (HTTP) is a vehicle for accessing the World Wide Web (WWW).
- The Simple Mail Transfer Protocol (SMTP) is the main protocol used in electronic mail (e-mail) service.
- The File Transfer Protocol (FTP) is used for transferring files from one host to another.
- The Terminal Network (TELNET) and Secure Shell (SSH) are used for accessing a site remotely.
- The Simple Network Management Protocol (SNMP) is used by an administrator to manage the Internet at global and local levels.
- The Domain Name System (DNS) is used by other protocols to find the network-layer address of a computer.
- The Internet Group Management Protocol (IGMP) is used to collect membership in a group.

PROTOCOL LAYERING

TCP/IP Protocol Suite

Description of Each Layer in TCP/IP

Transport Layer

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

PROTOCOL LAYERING

TCP/IP Protocol Suite

Description of Each Layer in TCP/IP

Transport Layer

Why we need an end-to-end transport layer when we already have an end-to-end application layer?

- The reason is the separation of tasks and duties.
- The transport layer should be independent of the application layer.
- In addition, we have more than one protocol in the transport layer, which means that each application program can use the protocol that best matches its requirement.

PROTOCOL LAYERING

TCP/IP Protocol Suite

Description of Each Layer in TCP/IP

Transport Layer

- Transmission Control Protocol (TCP) is a connection-oriented protocol that first establishes a logical connection between transport layers at two hosts before transferring data.
- It creates a logical pipe between two TCPs for transferring a stream of bytes.
- TCP provides flow control (matching the sending data rate of the source host with the receiving data rate of the destination host to prevent overwhelming the destination), error control (to guarantee that the segments arrive at the destination without error and resending the corrupted ones), and congestion control to reduce the loss of segments due to congestion in the network.

PROTOCOL LAYERING

TCP/IP Protocol Suite

Transport Layer

- User Datagram Protocol (UDP) is a connectionless protocol that transmits user datagrams without first creating a logical connection.
- In UDP, each user datagram is an independent entity without being related to the previous or the next one (the meaning of the term connectionless).
- UDP is a simple protocol that does not provide flow, error, or congestion control.
- Its simplicity is attractive to an application program that needs to send short messages and cannot afford the retransmission of the packets involved in TCP, when a packet is corrupted or lost.
- A new protocol, Stream Control Transmission Protocol (SCTP) is designed to respond to new applications that are emerging in the multimedia.

PROTOCOL LAYERING

TCP/IP Protocol Suite

Network Layer

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

PROTOCOL LAYERING

TCP/IP Protocol Suite

Network Layer

- Why we need the network layer?

We could have added the routing duty to the transport layer and dropped this layer.

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

PROTOCOL LAYERING

TCP/IP Protocol Suite

Network Layer

- The network layer in the Internet includes the main protocol, Internet Protocol (IP), that defines the format of the packet, called a datagram at the network layer.
- IP also defines the format and the structure of addresses used in this layer.
- IP is also responsible for routing a packet from its source to its destination, which is achieved by each router forwarding the datagram to the next router in its path.

PROTOCOL LAYERING

TCP/IP Protocol Suite

Network Layer

- IP is a connectionless protocol that provides no flow control, no error control, and no congestion control services.
- This means that if any of theses services is required for an application, the application should relay only on the transport-layer protocol.
- The network layer also includes unicast (one-to-one) and multicast (one-to-many) routing protocols.
- The network layer also has some auxiliary protocols that help IP in its delivery and routing tasks.

PROTOCOL LAYERING

TCP/IP Protocol Suite

Network Layer

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

PROTOCOL LAYERING

TCP/IP Protocol Suite

Data-link Layer

- An internet is made up of several links (LANs and WANs) connected by routers.
- The routers are responsible for choosing the best links.
- When the next link to travel is determined by the router, the data-link layer is responsible for taking the datagram and moving it across the link.
- The link can be a wired LAN with a link-layer switch, a wireless LAN, a wired WAN, or a wireless WAN.
- Data link layer is responsible for moving the packet through the link.

PROTOCOL LAYERING

TCP/IP Protocol Suite

Data-link Layer

- TCP/IP does not define any specific protocol for the data-link layer. It supports all the standard and proprietary protocols.
- Any protocol that can take the datagram and carry it through the link suffices for the network layer.
- The data-link layer takes a datagram and encapsulates it in a packet called a frame.
- Each link-layer protocol may provide a different service.
- Some link-layer protocols provide complete error detection and correction, some provide only error correction.

PROTOCOL LAYERING

TCP/IP Protocol Suite

Physical Layer

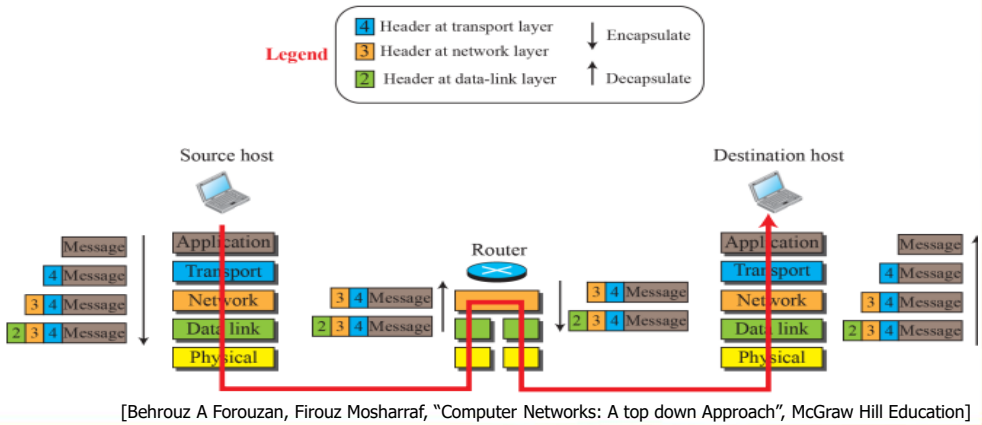
- Physical layer is responsible for carrying individual bits in a frame across the link.
- Two devices are connected by a transmission medium (cable or air).
- We need to know that the transmission medium does not carry bits; it carries electrical or optical signals.
- So the bits received in a frame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

PROTOCOL LAYERING

Encapsulation and Decapsulation

- One of the important concepts in protocol layering in the Internet is encapsulation/ decapsulation

Figure 1.16: Encapsulation / Decapsulation



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

PROTOCOL LAYERING

Encapsulation and Decapsulation

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

  - Encapsulation at the Source Host
  - Decapsulation and Encapsulation at Router
  - Decapsulation at the Destination Host

PROTOCOL LAYERING

Encapsulation at the Source Host

At the source, we have only encapsulation.

1. At the application layer, the data to be exchanged is referred to as a message. The message is passed to the transport layer.
2. The transport layer takes the message as the payload, the load that the transport layer should take care of. It adds the transport layer header to the payload, such as information needed for flow, error control, or congestion control. The result is the transport-layer packet, which is called the segment (in TCP) and the user datagram (in UDP). The transport layer then passes the packet to the network layer

PROTOCOL LAYERING

Encapsulation at the Source Host

At the source, we have only encapsulation.

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

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

PROTOCOL LAYERING

Decapsulation and Encapsulation at Router

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

1. After the set of bits are delivered to the data-link layer, this layer decapsulates the datagram from the frame and passes it to the network layer.
2. The network layer only inspects the source and destination addresses in the datagram header and consults its forwarding table to find the next hop to which the datagram is to be delivered. The datagram is then passed to the data-link layer of the next link.
3. The data-link layer of the next link encapsulates the datagram in a frame and passes it to the physical layer for transmission

PROTOCOL LAYERING

Decapsulation at the Destination Host

- At the destination host, each layer only decapsulates the packet received, removes the payload, and delivers the payload to the next-higher layer protocol until the message reaches the application layer.
- Decapsulation in the host involves error checking.

PROTOCOL LAYERING

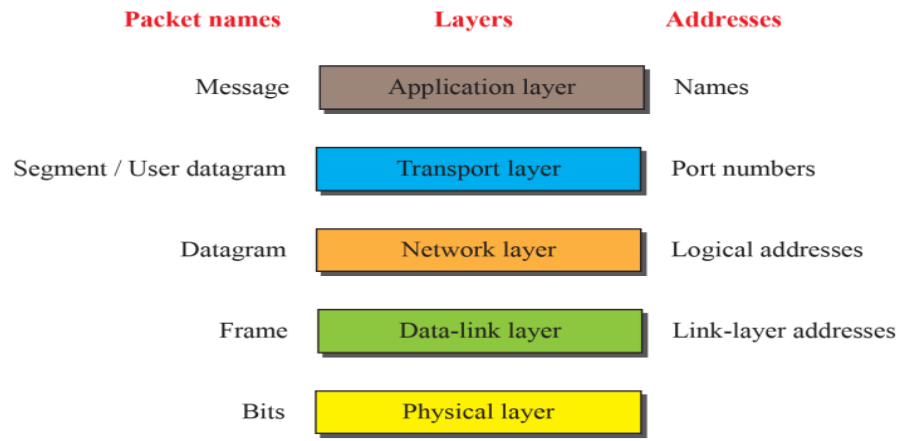
Addressing

- We have logical communication between pairs of layers in this model.
- Any communication that involves two parties needs two addresses: source address and destination address.
- Although it looks as if we need five pairs of addresses, one pair per layer, we normally have only four because the physical layer does not need addresses; the unit of data exchange at the physical layer is a bit, which definitely cannot have an address.

PROTOCOL LAYERING

Addressing

Figure 1.17: Addressing in the TCP/IP protocol suite



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

PROTOCOL LAYERING

Addressing

- At the application layer, we use names to define the site that provides services, such as someorg.com, or the e-mail address, such as somebody@coldmail.com.
- At the transport layer, addresses are called port numbers, and these define the application-layer programs at the source and destination.
- Port numbers are local addresses that distinguish between several programs running at the same time

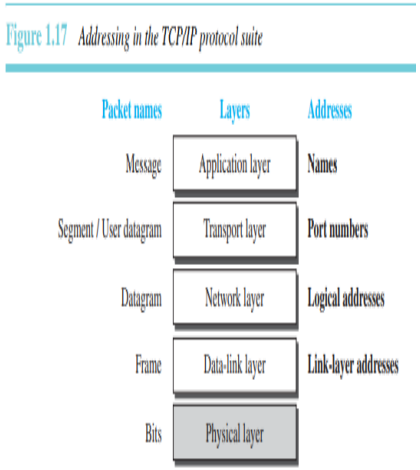


Figure 1.17 Addressing in the TCP/IP protocol suite

[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

PROTOCOL LAYERING

Addressing

- At the network-layer, the addresses are global, with the whole Internet as the scope.
- A network-layer address uniquely defines the connection of a device to the Internet.
- The link-layer addresses, sometimes called MAC addresses, are locally defined addresses, each of which defines a specific host or router in a network (LAN or WAN).

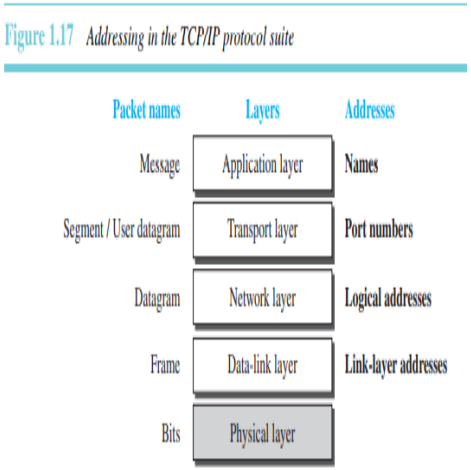


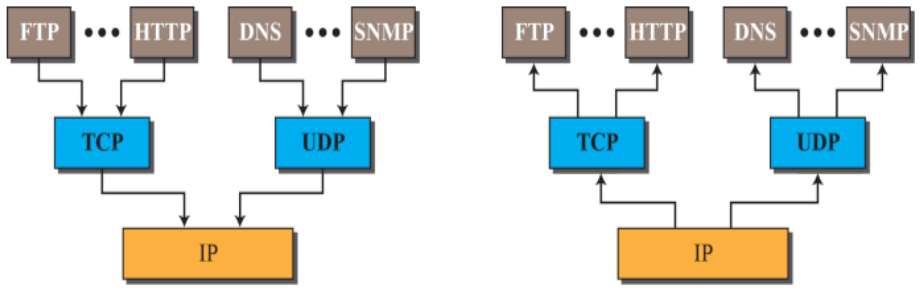
Figure 1.17 Addressing in the TCP/IP protocol suite

[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

PROTOCOL LAYERING

Multiplexing and Demultiplexing

Figure 1.18: Multiplexing and demultiplexing



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

PROTOCOL LAYERING

Multiplexing and Demultiplexing

- We have multiplexing at the source and demultiplexing at the destination.
- Multiplexing means that a protocol at a layer can encapsulate a packet from several next-higher layer protocols (one at a time); demultiplexing means that a protocol can decapsulate and deliver a packet to several next-higher layer protocols (one at a time).

PROTOCOL LAYERING

Multiplexing and Demultiplexing

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

PROTOCOL LAYERING

OSI Model

- An ISO (International Organization for Standardization) standard that covers all aspects of network communications is the Open Systems Interconnection (OSI) model.
- ISO is the organization; OSI is the model.
- An open system is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture.
- The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software.

PROTOCOL LAYERING

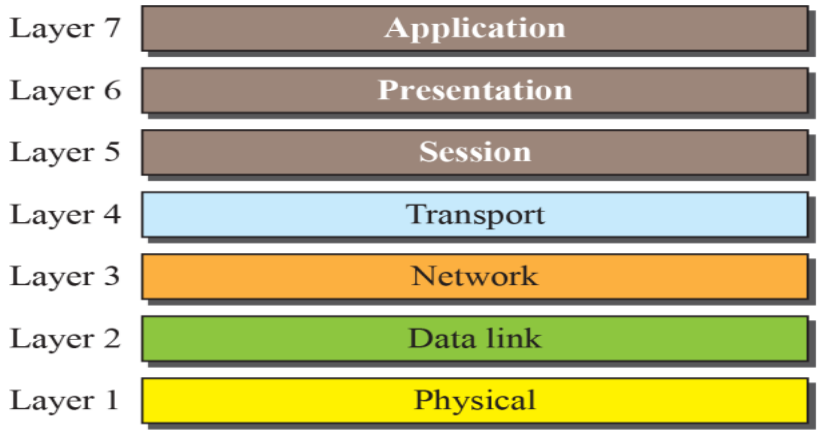
OSI Model

- The OSI model is a model for understanding and designing a network architecture that is flexible, robust, and interoperable.
- The OSI model was intended to be the basis for the creation of the protocols in the OSI stack.
- The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems.
- It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network

PROTOCOL LAYERING

OSI Model

Figure 1.19: The OSI model



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

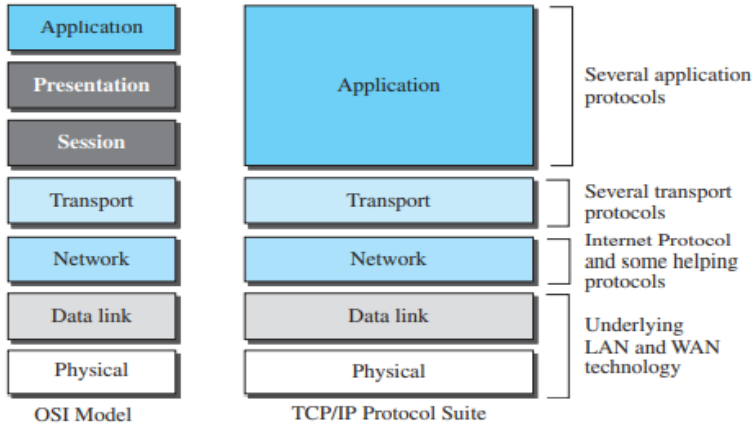
PROTOCOL LAYERING

- Let's consider the two additional layers present in the OSI reference model—the presentation layer and the session layer.
- The role of the presentation layer is to provide services that allow communicating applications to interpret the meaning of data exchanged.
- These services include data compression and data encryption as well as data description (frees the applications from having to worry about the internal format in which data are represented/stored—formats that may differ from one computer to another).
- The session layer provides for delimiting and synchronization of data exchange, including the means to build a checkpointing and recovery scheme.

PROTOCOL LAYERING

OSI versus TCP/IP

Figure 1.20 *TCP/IP and OSI model*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

PROTOCOL LAYERING

OSI versus TCP/IP

- The application layer in the suite is usually considered to be the combination of three layers in the OSI model.

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

INTERNET HISTORY

Early History
- Birth of Packet-Switched Networks
- Advanced Research Projects Agency Network (ARPANET)

Birth of the Internet
- TCP/IP
- Military Network (MILNET)
- Computer Science Network (CSNET)
- National Science Foundation Network (NSFNET)
- Advanced Network Services Network (ANSNET)

Internet Today
- World Wide Web
- Multimedia
- Peer-to-Peer Applications

INTERNET HISTORY

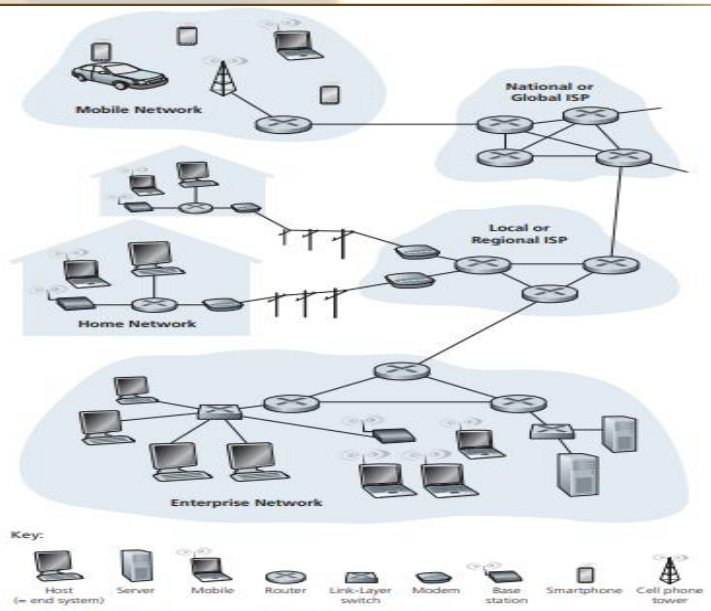


Figure 1.21 Some pieces of the Internet

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

INTERNET HISTORY

- Internet standards are developed by the Internet Engineering Task Force (IETF).
- The IETF standards documents are called requests for comments (RFCs).
- RFCs started out as general requests for comments (hence the name) to resolve network and protocol design problems that faced the precursor to the Internet.
- RFCs define protocols such as TCP, IP, HTTP (for the Web), and SMTP (for e-mail).
- There are currently more than 6,000 RFCs.
- Other bodies also specify standards for network components, like IEEE 802 LAN/MAN Standards Committee, for example, specifies the Ethernet and wireless WiFi standards.

INTERNET HISTORY

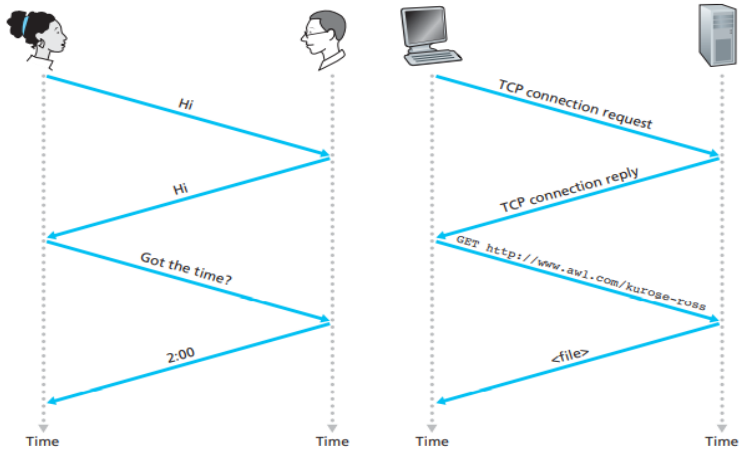


Figure 1.2 3 A human protocol and a computer network protocol

- 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.

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

The Network Core

- In a network application, end systems exchange messages with each other.
- To send a message from a source end system to a destination end system, the source breaks long messages into smaller chunks of data known as packets.
- Between source and destination, each packet travels through communication links and packet switches (for which there are two predominant types, routers and link layer switches).
- Packets are transmitted over each communication link at a rate equal to the full transmission rate of the link.
- If a source end system or a packet switch is sending a packet of L bits over a link with transmission rate R bits/sec, then the time to transmit the packet is L/R seconds.

The Network Core

Store-and-Forward Transmission

- Most packet switches use store-and-forward transmission at the inputs to the links.
- Store-and-forward transmission means that the packet switch must receive the entire packet before it can begin to transmit the first bit of the packet onto the outbound link.

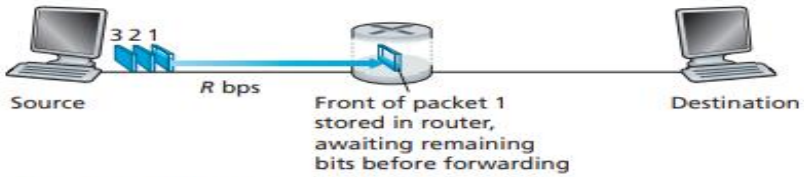


Figure 1.23 Store-and-forward packet switching

The Network Core

Store-and-Forward Transmission

- Calculate the amount of time that elapses from when the source begins to send the packet until the destination has received the entire packet. (Here we will ignore propagation delay—the time it takes for the bits to travel across the wire at near the speed of light)

- The source begins to transmit at time 0; at time L/R seconds, the source has transmitted the entire packet, and the entire packet has been received and stored at the router (since there is no propagation delay).

The Network Core

Store-and-Forward Transmission

- At time L/R seconds, since the router has just received the entire packet, it can begin to transmit the packet onto the outbound link towards the destination; at time 2L/R, the router has transmitted the entire packet, and the entire packet has been received by the destination.
- Thus, the total delay is 2L/R.
- If the switch instead forwarded bits as soon as they arrive (without first receiving the entire packet), then the total delay would be L/R since bits are not held up at the router.

The Network Core

Store-and-Forward Transmission

- Calculate the amount of time that elapses from when the source begins to send the first packet until the destination has received all three packets.
- As before, at time L/R, the router begins to forward the first packet.
- But also at time L/R the source will begin to send the second packet, since it has just finished sending the entire first packet.
- Thus, at time 2L/R, the destination has received the first packet and the router has received the second packet.
- Similarly, at time 3L/R, the destination has received the first two packets and the router has received the third packet.
- Finally, at time 4L/R the destination has received all three packets!

The Network Core

Store-and-Forward Transmission

- Consider the general case of sending one packet from source to destination over a path consisting of N links each of rate R (thus, there are N-1 routers between source and destination).
- Applying the same logic as above, we see that the end-to-end delay is:

$$d_{\text{end-to-end}} = N\frac{L}{R}$$

The Network Core

Queuing Delays and Packet Loss

- For each attached link, the packet switch has an output buffer (also called an output queue), which stores packets that the router is about to send into that link.
- In addition to the store-and-forward delays, packets suffer output buffer queuing delays.
- These delays are variable and depend on the level of congestion in the network.
- Since the amount of buffer space is finite, an arriving packet may find that the buffer is completely full with other packets waiting for transmission.
- In this case, packet loss will occur—either the arriving packet or one of the already-queued packets will be dropped

The Network Core

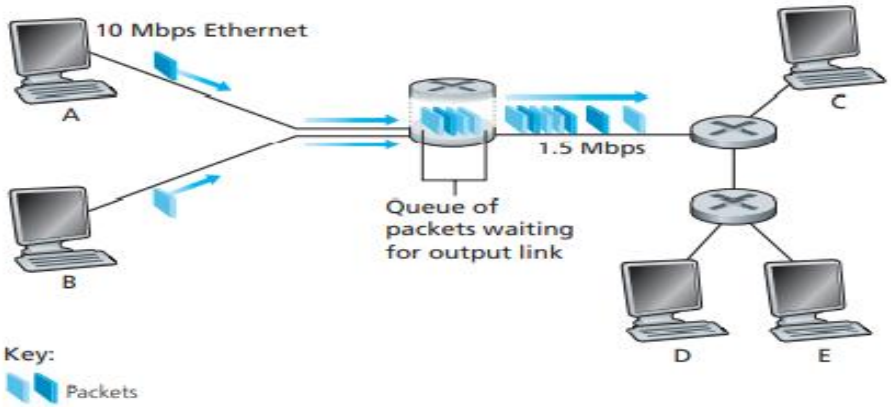


Figure 1. 24 Packet switching

- Figure 1.24 illustrates a simple packet-switched network.

The Network Core

Circuit Switching and Packet Switching

- There are two fundamental approaches to moving data through a network of links and switches: circuit switching and packet switching.
- In circuit-switched networks, the resources needed along a path (buffers, link transmission rate) to provide for communication between the end systems are reserved for the duration of the communication session between the end systems.
- In packet-switched networks, these resources are not reserved; a session's messages use the resources on demand, and as a consequence, may have to wait (that is, queue) for access to a communication link.

The Network Core

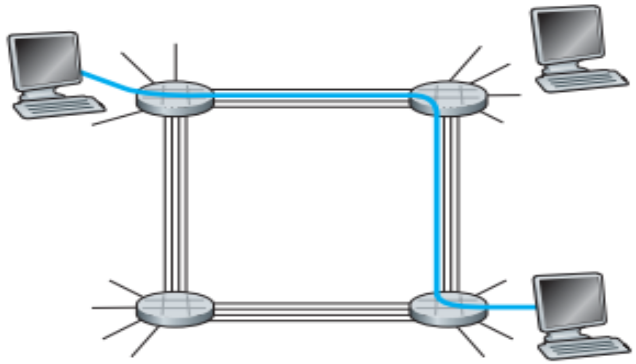


Figure 1. 25 A simple circuit-switched network consisting of four switches and four links

The Network Core

Multiplexing in Circuit-Switched Networks

- A circuit in a link is implemented with either frequency-division multiplexing (FDM) or time-division multiplexing (TDM).
- With FDM, the frequency spectrum of a link is divided up among the connections established across the link.

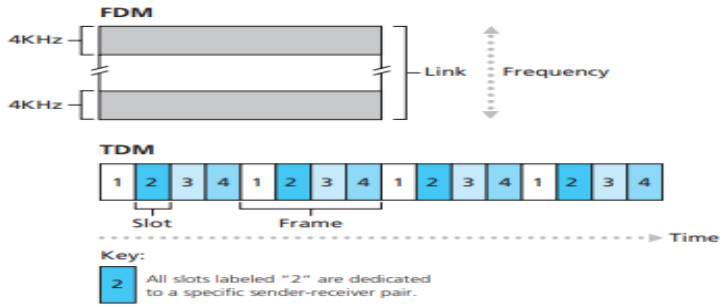


Figure 1. 26 With FDM, each circuit continuously gets a fraction of the bandwidth. With TDM, each circuit gets all of the bandwidth periodically during brief intervals of time (that is, during slots)

Delay, Loss, and Throughput in Packet-Switched Networks

Overview of Delay in Packet-Switched Networks

- Packet suffers from several types of delays at each node along the path.
- The most important of these delays are the nodal processing delay, queuing delay, transmission delay, and propagation delay; together, these delays accumulate to give a total nodal delay.

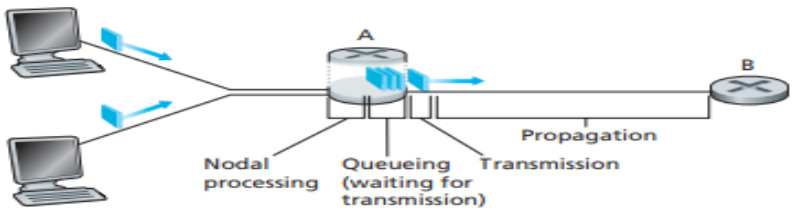


Figure 1. 27 The nodal delay at router A

Types of Delay

Processing Delay

- The time required to examine the packet's header and determine where to direct the packet is part of the processing delay.
- The processing delay can also include other factors, such as the time needed to check for bit-level errors in the packet that occurred in transmitting the packet's bits from the upstream node to router A.
- Processing delays in high-speed routers are typically on the order of microseconds or less.
- After this nodal processing, the router directs the packet to the queue that precedes the link to router B.

Types of Delay

Queuing Delay

- At the queue, the packet experiences a queuing delay as it waits to be transmitted onto the link.
- The length of the queuing delay of a specific packet will depend on the number of earlier-arriving packets that are queued and waiting for transmission onto the link.
- If the queue is empty and no other packet is currently being transmitted, then our packet's queuing delay will be zero.
- On the other hand, if the traffic is heavy and many other packets are also waiting to be transmitted, the queuing delay will be long.
- The number of packets that an arriving packet might expect to find is a function of the intensity and nature of the traffic arriving at the queue.
- Queuing delays can be on the order of microseconds to milliseconds in practice.

Types of Delay

Transmission Delay

- Assuming that packets are transmitted in a first-come-first-served manner, the packet can be transmitted only after all the packets that have arrived before it have been transmitted.
- Denote the length of the packet by L bits, and denote the transmission rate of the link from router A to router B by R bits/sec.
- For example, for a 10 Mbps Ethernet link, the rate is R = 10 Mbps; for a 100 Mbps Ethernet link, the rate is R = 100 Mbps.
- The transmission delay is L/R. This is the amount of time required to push (that is, transmit) all of the packet's bits into the link.
- Transmission delays are typically on the order of microseconds to milliseconds in practice.

Types of Delay

Propagation Delay

- Once a bit is pushed into the link, it needs to propagate to router B. The time required to propagate from the beginning of the link to router B is the propagation delay.
- The bit propagates at the propagation speed of the link.
- The propagation speed depends on the physical medium of the link (that is, fiber optics, twisted-pair copper wire, coaxial cable and so on) and is in the range of 2.10^8 meters/sec to 3.10^8 meters/sec which is equal to, or a little less than, the speed of light.

Types of Delay

Propagation Delay

- The propagation delay is the distance between two routers divided by the propagation speed.
- That is, the propagation delay is d/s, where d is the distance between router A and router B and s is the propagation speed of the link.
- Once the last bit of the packet propagates to node B, it and all the preceding bits of the packet are stored in router B.
- The whole process then continues with router B now performing the forwarding.
- In wide-area networks, propagation delays are on the order of milliseconds.

Types of Delay

Comparing Transmission and Propagation 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, but has nothing to do with the distance between the two routers.
- The propagation delay 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, but has nothing to do with the packet's length or the transmission rate of the link.

Types of Delay

- If d_{proc}, d_{queue}, d_{trans}, and d_{prop} denote the processing, queuing, transmission, and propagation delays, then the total nodal delay is given by $d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$
- The contribution of these delay components can vary significantly.
- For example, d_{prop} can be negligible (for example, a couple of microseconds) for a link connecting two routers on the same university campus; however, d_{prop} is hundreds of milliseconds for two routers interconnected by a geostationary satellite link, and can be the dominant term in d_{nodal}.

Types of Delay

- Similarly, d_{trans} can range from negligible to significant. Its contribution is typically negligible for transmission rates of 10 Mbps and higher (for example, for LANs); however, it can be hundreds of milliseconds for large Internet packets sent over low-speed dial-up modem links.
- The processing delay, d_{proc}, is often negligible; however, it strongly influences a router's maximum throughput, which is the maximum rate at which a router can forward packets.

Queuing Delay and Packet Loss

- The most complicated and interesting component of nodal delay is the queuing delay, d_{queue}.
- Unlike the other three delays (namely, d_{proc}, d_{trans}, and d_{prop}), the queuing delay can vary from packet to packet.
- For example, if 10 packets arrive at an empty queue at the same time, the first packet transmitted will suffer no queuing delay, while the last packet transmitted will suffer a relatively large queuing delay (while it waits for the other nine packets to be transmitted).
- Therefore, when characterizing queuing delay, one typically uses statistical measures, such as average queuing delay, variance of queuing delay, and the probability that the queuing delay exceeds some specified value

Queuing Delay and Packet Loss

When is the queuing delay large and when is it insignificant?

- The answer to this question depends on the rate at which traffic arrives at the queue, the transmission rate of the link, and the nature of the arriving traffic, that is, whether the traffic arrives periodically or arrives in bursts.
- Let a denote the average rate at which packets arrive at the queue (a is in units of packets/sec).
- R is the transmission rate; that is, it is the rate (in bits/sec) at which bits are pushed out of the queue.
- Suppose all packets consist of L bits. Then the average rate at which bits arrive at the queue is La bits/sec.

Queuing Delay and Packet Loss

When is the queuing delay large and when is it insignificant?

- Assume that the queue is very big, so that queuing delay will approach infinity and can hold essentially an infinite number of bits.
- The ratio La/R, called the traffic intensity, often plays an important role in estimating the extent of the queuing delay.
- If La/R > 1, then the average rate at which bits arrive at the queue exceeds the rate at which the bits can be transmitted from the queue.
- In this situation, the queue will tend to increase without bound and the queuing delay will approach infinity!
- Therefore, one of the golden rules in traffic engineering is: Design your system so that the traffic intensity is no greater than 1.

Queuing Delay and Packet Loss

When is the queuing delay large and when is it insignificant?

- Now consider the case La/R ≤ 1.
- The nature of the arriving traffic impacts the queuing delay.
- For example, if packets arrive periodically—that is, one packet arrives every L/R seconds—then every packet will arrive at an empty queue and there will be no queuing delay.
- If packets arrive in bursts but periodically, there can be a significant average queuing delay.
- For example, suppose N packets arrive simultaneously every (L/R)N seconds. Then the first packet transmitted has no queuing delay; the second packet transmitted has a queuing delay of L/R seconds; and more generally, the nth packet transmitted has a queuing delay of (n-1)L/R seconds.

Queuing Delay and Packet Loss

- As the traffic intensity approaches 1, the average queuing delay increases rapidly.
- A small percentage increase in the intensity will result in a much larger percentage-wise increase in delay.
- If some event causes an even slightly larger-than-usual amount of traffic, the delays you experience can be huge.

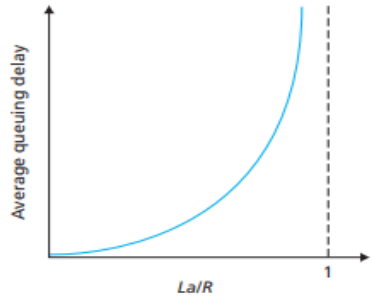


Figure 1.28 Dependence of average queuing delay on traffic intensity

Packet Loss

- We have assumed that the queue is capable of holding an infinite number of packets.
- In reality a queue preceding a link has finite capacity, although the queuing capacity greatly depends on the router design and cost.
- Because the queue capacity is finite, packet delays do not really approach infinity as the traffic intensity approaches 1.
- Instead, a packet can arrive to find a full queue.
- With no place to store such a packet, a router will drop that packet; that is, the packet will be lost.

Packet Loss

- From an end-system viewpoint, a packet loss will look like a packet having been transmitted into the network core but never emerging from the network at the destination.
- The fraction of lost packets increases as the traffic intensity increases.
- Therefore, performance at a node is often measured not only in terms of delay, but also in terms of the probability of packet loss.

End-to-End Delay

- Consider the total delay from source to destination.
- Suppose there are N-1 routers between the source host and the destination host.
- Let's also suppose that the network is uncongested (so that queuing delays are negligible), the processing delay at each router and at the source host is d_{proc}, the transmission rate out of each router and out of the source host is R bits/sec, and the propagation on each link is d_{prop}.
- The nodal delays accumulate and give an end-to-end delay, $d_{end\text{-}end} = N (d_{proc} + d_{trans} + d_{prop})$
  where, once again, dtrans = L/R, where L is the packet size.

Throughput in Computer Networks

- Another critical performance measure in computer networks is end-to-end throughput.
- To define throughput, consider transferring a large file from Host A to Host B across a computer network.
- This transfer might be, a large video clip from one peer to another in a P2P file sharing system.
- The instantaneous throughput at any instant of time is the rate (in bits/sec) at which Host B is receiving the file.
- If the file consists of F bits and the transfer takes T seconds for Host B to receive all F bits, then the average throughput of the file transfer is F/T bits/sec.

QUALITY OF SERVICE

- The Internet was originally designed for best-effort service with of guarantee of predictable performance.
- Best-effort service is often sufficient for a traffic that is not sensitive to delay, such as file transfers and e-mail.
- Such a traffic is called elastic because it can stretch to work under delay conditions; it is also called available bit rate because applications can speed up or slow down according to the available bit rate.
- The real-time traffic generated by some multimedia applications.
- The real-time traffic is delay sensitive and therefore requires guaranteed and predictive performance.
- 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.

QUALITY OF SERVICE

Data-Flow Characteristics

- If we want to provide quality of service for an Internet application, we first need to define what we need for each application.
- Traditionally, four types of characteristics are attributed to a flow: reliability, delay, jitter, and bandwidth.
- Reliability: Reliability is a characteristic that a flow needs in order to deliver the packets safe and sound to the destination.
- Delay: Source-to-destination delay is another flow characteristic.
- Jitter: Jitter is the variation in delay for packets belonging to the same flow.
- Bandwidth: Different applications need different bandwidths. In video conferencing we need to send millions of bits per second to refresh a color screen while the total number of bits in an e-mail may not reach even a million.

QUALITY OF SERVICE

Sensitivity of Applications

Table 1.1 *Sensitivity of applications to flow characteristics*

Application	*Reliability*	*Delay*	*Jitter*	*Bandwidth*
FTP	High	Low	Low	Medium
HTTP	High	Medium	Low	Medium
Audio-on-demand	Low	Low	High	Medium
Video-on-demand	Low	Low	High	High
Voice over IP	Low	High	High	Low
Video over IP	Low	High	High	High

[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

QUALITY OF SERVICE

- For those applications with a high level of sensitivity to reliability, we need to do error checking and discard the packet if corrupted.
- For those applications with a high level of sensitivity to delay, we need to be sure that they are given priority in transmission.
- For those applications with a high level of sensitivity to jitter, we need to be sure that the packets belonging to the same application pass the network with the same delay.
- For those applications that require high bandwidth, we need to allocate enough bandwidth to be sure that the packets are not lost.
- Several scheduling techniques are designed to improve the quality of service.
- Various service models in Quality of Service (QOS) are also designed. [Integrated Services (IntServ), Differentiated Services (DiffServ)]

QUALITY OF SERVICE

Integrated Services (IntServ)

- To provide different QoS for different applications, IETF developed the integrated services (IntServ) model.
- In this model, which is a flow-based architecture, resources such as bandwidth are explicitly reserved for a given data flow.
- In other words, the model is considered a specific requirement of an application in one particular case regardless of the application type (data transfer, or voice over IP, or video-on-demand).
- What is important are the resources the application needs, not what the application is doing.

QUALITY OF SERVICE

Integrated Services (IntServ)

- The model is based on three schemes:
  1. The packets are first classified according to the service they require.
  2. The model uses scheduling to forward the packets according to their flow characteristics.
  3. Devices like routers use admission control to determine if the device has the capability (available resources to handle the flow) before making a commitment. For example, if an application requires a very high data rate, but a router in the path cannot provide such a data rate, it denies the admission.
- Integrated Services is a flow-based QoS model designed for IP. In this model packets are marked by routers according to flow characteristics.

QUALITY OF SERVICE

Differentiated Services (DiffServ)

- Packets are marked by applications into classes according to their priorities.
- Routers and switches, using various queuing strategies, route the packets.
- This model was introduced by the IETF (Internet Engineering Task Force) to handle the shortcomings of Integrated Services.
- Two fundamental changes were made:
  1. The main processing was moved from the core of the network to the edge of the network. This solves the scalability problem. The routers do not have to store information about flows. The applications, or hosts, define the type of service they need each time they send a packet.

QUALITY OF SERVICE

Differentiated Services (DiffServ)

2. The per-flow service is changed to per-class service. The router routes the packet based on the class of service defined in the packet, not the flow. This solves the service-type limitation problem.

- Differentiated Services is a class-based QoS model designed for IP. In this Model packets are marked by applications according to their priority.

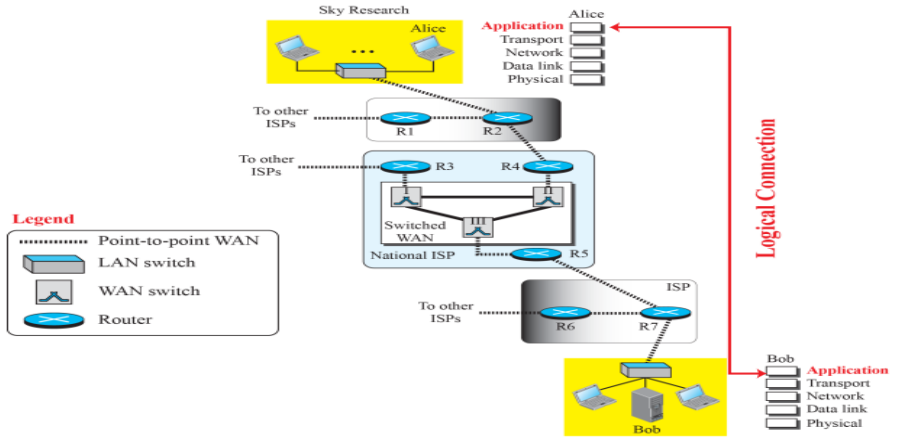
Application Layer

- In this section, we introduce the nature of services provided by the Internet and introduce two categories of applications: the traditional one, based on the client server paradigm, and the new one, based on the peer-to-peer paradigm.
- We also discuss some predefined or standard applications based on the client-server paradigm such as surfing the Web, file transfer, e-mail, and so on.
- We also discuss the concept and protocols in the peer-to-peer paradigm.

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.

Figure 1.29 Logical connection at the application layer



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

Application Layer

- The application layer is different from other layers in that it is the highest layer in the suite.
- The protocols in this layer do not provide services to any other protocol in the suite; they only receive services from the protocols in the transport layer.
- This means that protocols can be removed from this layer easily.
- New protocols can be also added to this layer as long as the new protocol can use the service provided by one of the transport-layer protocols.

Application Layer

Standard and Nonstandard Protocols

- Standard Application-Layer Protocols are several application-layer protocols that have been standardized and documented by the Internet authority, and we are using them in our daily interaction with the Internet.
- Each standard protocol is a pair of computer programs that interact with the user and the transport layer to provide a specific service to the user.
- Nonstandard Application-Layer Protocols: A programmer can create a nonstandard application-layer program if she can write programs that provide service to the user by interacting with the transport layer.

Application-Layer Paradigms

- To use the Internet we need two application programs to interact with each other: one running on a computer somewhere in the world, the other running on another computer somewhere else in the world.
- The two programs need to send messages to each other through the Internet infrastructure.
- Should both application programs be able to request services and provide services, or should the application programs just do one or the other?
- Two paradigms have been developed during the lifetime of the Internet to answer this question: the client-server paradigm and the peer-to-peer paradigm.

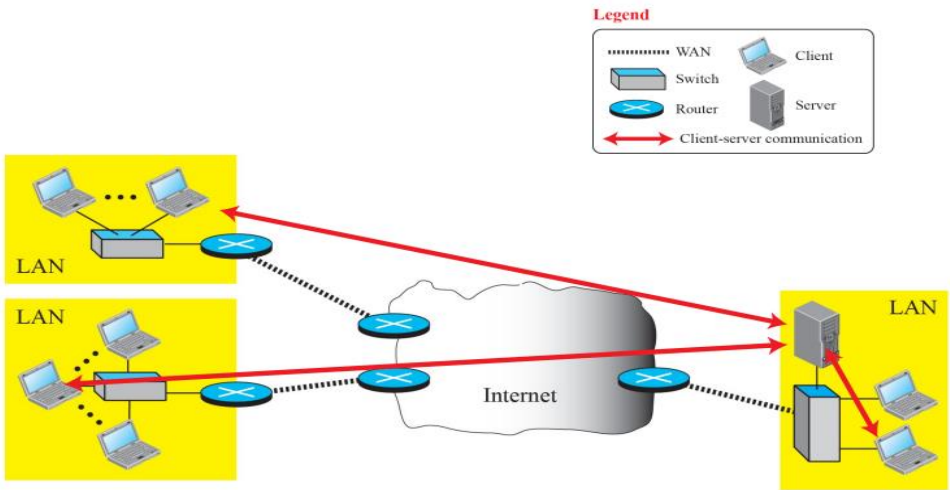
Application-Layer Paradigms

Traditional Paradigm: Client-Server

- The traditional paradigm is called the client-server paradigm.
- In this paradigm, the service provider is an application program, called the server process; it runs continuously, waiting for another application program, called the client process, to make a connection through the Internet and ask for service.
- The server process must be running all the time; the client process is started when the client needs to receive service.

Application-Layer Paradigms

Figure 1.30 Example of a client-server paradigm



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

Application-Layer Paradigms

Traditional Paradigm: Client-Server

- One problem with this paradigm is that the concentration of the communication load is on the shoulder of the server, which means the server should be a powerful computer.
- Even a powerful computer may become overwhelmed if a large number of clients try to connect to the server at the same time.
- Another problem is that there should be a service provider willing to accept the cost and create a powerful server for a specific service, which means the service must always return some type of income for the server in order to encourage such an arrangement.
- Several traditional services are still using this paradigm, including the World Wide Web (WWW) and its vehicle Hyper Text Transfer Protocol (HTTP), File Transfer Protocol (FTP), secure shell (SSH), e-mail, and so on.

Application-Layer Paradigms

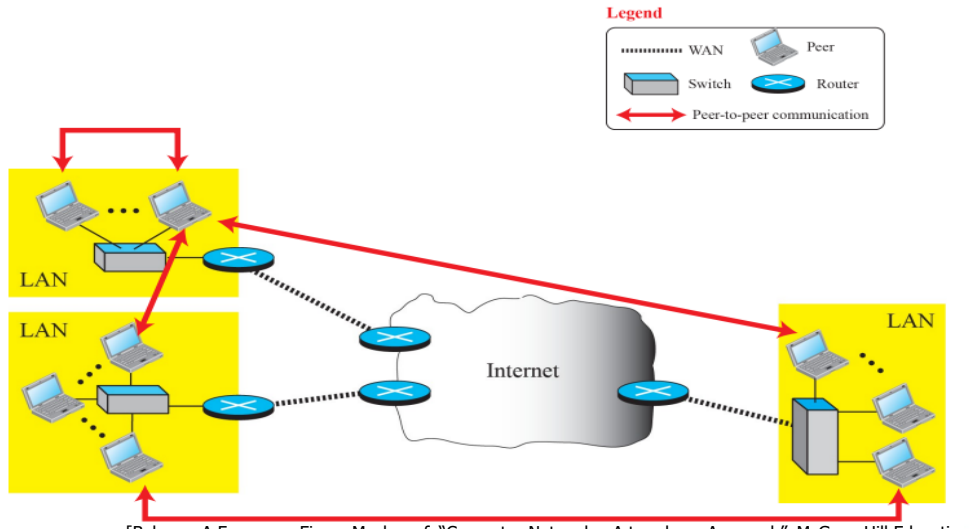
New Paradigm: Peer-to-Peer

- A new paradigm, called the peer-to-peer paradigm (often abbreviated P2P paradigm) has emerged to respond to the needs of some new applications.
- In this paradigm, there is no need for a server process to be running all the time and waiting for the client processes to connect.
- The responsibility is shared between peers.
- A computer connected to the Internet can provide service at one time and receive service at another time.
- A computer can even provide and receive services at the same time.
- One of the areas that really fits in this paradigm is the Internet telephony.

Application-Layer Paradigms

New Paradigm: Peer-to-Peer

Figure 1.31 Example of a peer-to-peer paradigm



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

Application-Layer Paradigms

New Paradigm: Peer-to-Peer

- Peer-to-peer paradigm has been proved to be easily scalable and cost-effective in eliminating the need for expensive servers to be running and maintained all the time, there are also some challenges.
- The main challenge has been security; it is more difficult to create secure communication between distributed services than between those controlled by some dedicated servers.
- The other challenge is applicability; it appears that not all applications can use this new paradigm.
- For example, not many Internet users are ready to become involved, if one day the Web can be implemented as a peer-to-peer service.
- There are some new applications, such as BitTorrent [communication protocol for peer to peer file sharing], Skype, IPTV, and Internet telephony, that use this paradigm.

Application-Layer Paradigms

Mixed Paradigm

- An application may choose to use a mixture of the two paradigms by combining the advantages of both.
- For example, a light-load client-server communication can be used to find the address of the peer that can offer a service. When the address of the peer is found, the actual service can be received from the peer by using the peer-to-peer paradigm.

CLIENT-SERVER PARADIGM

- In a client-server paradigm, communication at the application layer is between two running application programs called processes: a client and a server.
- A client is a running program that initializes the communication by sending a request; a server is another application program that waits for a request from a client.
- The server handles the request received from a client, prepares a result, and sends the result back to the client.
- This definition of a server implies that a server must be running when a request from a client arrives, but the client needs to be run only when it is needed.

CLIENT-SERVER PARADIGM

- If we have two computers connected to each other somewhere, we can run a client process on one of them and the server on the other.
- We need to be careful that the server program is started before we start running the client program.
- In other words, the lifetime of a server is infinite: it should be started and run forever, waiting for the clients.
- The lifetime of a client is finite: it normally sends a finite number of requests to the corresponding server, receives the responses, and stops

CLIENT-SERVER PARADIGM

Application Programming Interface

How can a client process communicate with a server process?

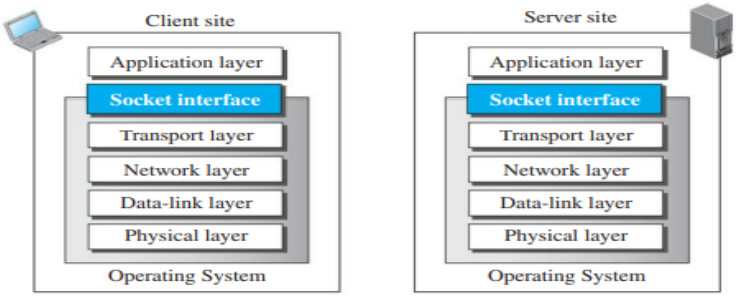
- If we need a process to be able to communicate with another process, we need a new set of instructions to tell the lowest four layers of the TCP/IP suite to open the connection, send and receive data from the other end, and close the connection.
- A set of instructions of this kind is normally referred to as Application Programming Interface (API).
- An interface in programming is a set of instructions between two entities.
- In this case, one of the entities is the process at the application layer and the other is the operating system that encapsulates the first four layers of the TCP/IP protocol suit.
- Several APIs have been designed for communication like socket interface, Transport Layer Interface (TLI), and STREAM.

CLIENT-SERVER PARADIGM

Sockets

- Although a socket is supposed to behave like a terminal or a file, it is not a physical entity like them; it is an abstraction.
- It is a data structure that is created and used by the application program.

Figure 1.32 *Position of the socket interface*



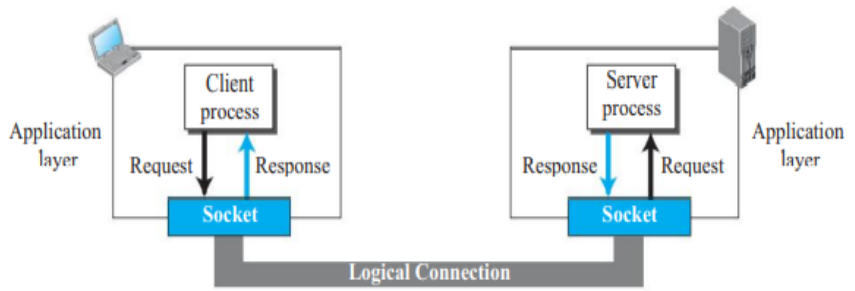
[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

CLIENT-SERVER PARADIGM

Sockets

- Communication between a client process and server process is communication between two sockets, created at two ends as shown in the figure.

Figure 1.33 *Use of sockets in process-to-process communication*



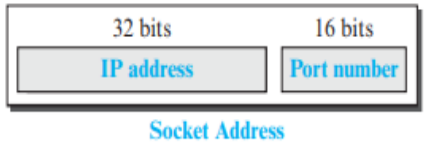
[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

CLIENT-SERVER PARADIGM

Socket Addresses

- Since communication in the client-server paradigm is between two sockets, we need a pair of socket addresses for communication: a local socket address and a remote socket address.

Figure 1.34 *A socket address*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

CLIENT-SERVER PARADIGM

Server Site

- The server needs a local (server) and a remote (client) socket address for communication.

Local Socket Address

- The local (server) socket address is provided by the operating system.
- The operating system knows the IP address of the computer on which the server process is running.
- The port number of a server process needs to be assigned.
- If the server process is a standard one defined by the Internet authority, a port number is already assigned to it.
- If the server process is not standard, the designer of the server process can choose a port number and assign it to the process.
- When a server starts running, it knows the local socket address.

CLIENT-SERVER PARADIGM

Server Site

- The server needs a local (server) and a remote (client) socket address for communication.

Remote Socket Address

- The remote socket address for a server is the socket address of the client that makes the connection.
- Since the server can serve many clients, it does not know beforehand the remote socket address for communication.
- The server can find this socket address when a client tries to connect to the server.
- The client socket address, which is contained in the request packet sent to the server, becomes the remote socket address that is used for responding to the client.
- In other words, although the local socket address for a server is fixed and used during its lifetime, the remote socket address is changed in each interaction with a different client.

CLIENT-SERVER PARADIGM

Client Site

- The client also needs a local (client) and a remote (server) socket address for communication.

Local Socket Address

- The local (client) socket address is also provided by the operating system. The operating system knows the IP address of the computer on which the client is running.
- The port number is a 16-bit temporary integer that is assigned to a client process each time the process needs to start the communication.
- The port number needs to be assigned from a set of integers defined by the Internet authority and called the ephemeral (temporary) port numbers.
- The operating system needs to guarantee that the new port number is not used by any other running client process.

CLIENT-SERVER PARADIGM

Client Site

- The client also needs a local (client) and a remote (server) socket address for communication.

Remote Socket Address

- Finding the remote (server) socket address for a client needs more work. When a client process starts, it should know the socket address of the server it wants to connect to.
- We will have two situations in this case.
  1. The user who starts the client process knows both the server port number and IP address of the computer on which the server is running.
     - This usually occurs in situations when we have written client and server applications and we want to test.

CLIENT-SERVER PARADIGM

Client Site

Remote Socket Address

2. Each standard application has a well-known port number, most of the time, we do not know the IP address. This happens in situations such as when we need to contact a Webpage, send an e-mail to a friend, copy a file from a remote site, and so on.

- In these situations, an identifier that uniquely defines the server process. The client process should now change this identifier.
- Examples of these identifiers are URLs, such as www.xxx.yyy, or e-mail addresses, such as xxxx@yyyy.com.
- The client process should now change this identifier (name) to the corresponding server socket address.
- The client process normally knows the port number because it should be a well-known port number, but the IP address can be obtained using another client server application called the Domain Name System (DNS).

CLIENT-SERVER PARADIGM

Using Services of the Transport Layer

- There are three common transport layer protocols in the TCP/IP suite: UDP, TCP, and SCTP(Stream Control Transmission Protocol).
- Most standard applications have been designed to use the services of one of these protocols.
- When we write a new application, we can decide which protocol we want to use.
- The choice of the transport layer protocol seriously affects the capability of the application processes.

CLIENT-SERVER PARADIGM

Using Services of the Transport Layer

UDP Protocol

- UDP provides connectionless, unreliable, datagram service.
- UDP has an advantage: it is message-oriented. It gives boundaries to the messages exchanged.
- We can compare the connectionless and unreliable service to the regular service provided by the post office.
- An application program may be designed to use UDP if it is sending small messages and the simplicity and speed is more important for the application than reliability.
- For example, some management and multimedia applications fit in this category.

CLIENT-SERVER PARADIGM

Using Services of the Transport Layer

TCP Protocol

- TCP provides connection-oriented, reliable, byte-stream service.
- By numbering the bytes exchanged, the continuity of the bytes can be checked.
- if some bytes are lost or corrupted, the receiver can request the resending of those bytes, which makes TCP a reliable protocol. TCP also can provide flow control and congestion control.
- We can compare the service provided by TCP to the service provided by the telephone company.
- Most of the standard applications that need to send long messages and require reliability may benefit from the service of the TCP.

CLIENT-SERVER PARADIGM

Using Services of the Transport Layer

SCTP Protocol

- SCTP provides a service which is a combination of TCP and UDP protocols.
- Like TCP, SCTP provides a connection-oriented, reliable service, but it is not byte-stream oriented.
- It is a message-oriented protocol like UDP.
- SCTP can provide multistream service by providing multiple network-layer connections.
- SCTP is normally suitable for any application that needs reliability and at the same time needs to remain connected, even if a failure occurs in one network-layer connection.

STANDARD CLIENT-SERVER APPLICATIONS

World Wide Web and HTTP

World Wide Web (abbreviated WWW or Web)

- The idea of the Web was first proposed by Tim Berners-Lee in 1989 at CERN, the European Organization for Nuclear Research, to allow several researchers at different locations throughout Europe to access each others' researches.
- The commercial Web started in the early 1990s.
- The Web today is a repository of information in which the documents, called Web pages, are distributed all over the world and related documents are linked together.
- The popularity and growth of the Web can be related to two terms in the above statement: distributed and linked.
- The linking of web pages was achieved using a concept called hypertext, now it is known as hypermedia.
- The purpose of the Web has gone beyond the simple retrieving of linked documents. [The Web is used to provide electronic shopping and gaming.]

STANDARD CLIENT-SERVER APPLICATIONS
World Wide Web

Architecture

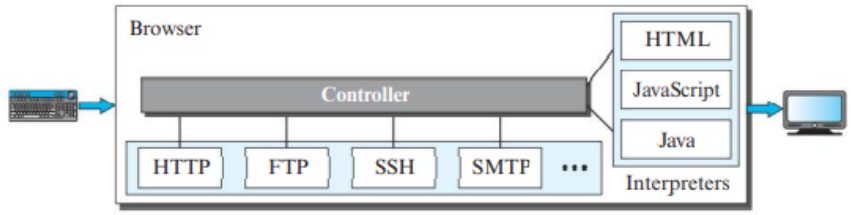
- The WWW today is a distributed client-server service, in which a client using a browser can access a service using a server.
- The service provided is distributed over many locations called sites.
- Each site holds one or more documents, referred to as web pages.
- Each web page, however, can contain some links to other web pages in the same or other sites.
- In other words, a web page can be simple or composite. A simple web page has no links to other web pages; a composite web page has one or more links to other web pages.
- Each web page is a file with a name and address.

STANDARD CLIENT-SERVER APPLICATIONS
World Wide Web

Web Client (Browser)

- A variety of vendors offer commercial browsers that interpret and display a web page, and all of them use nearly the same architecture.
- Each browser usually consists of three parts: a controller, client protocols, and interpreters.

Figure 1.35 *Browser*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS
World Wide Web

Web Client (Browser)

- The controller receives input from the keyboard or the mouse and uses the client programs to access the document.
- After the document has been accessed, the controller uses one of the interpreters to display the document on the screen.
- The client protocol can be one of the protocols such as HTTP or FTP.
- The interpreter can be HTML, Java, or JavaScript, depending on the type of document.
- Some commercial browsers include Internet Explorer, Netscape Navigator, and Firefox.

STANDARD CLIENT-SERVER APPLICATIONS
World Wide Web

Web Server

- The web page is stored at the server.
- Each time a request arrives, the corresponding document is sent to the client.
- To improve efficiency, servers normally store requested files in a cache in memory; memory is faster to access than disk.
- A server can also become more efficient through multithreading or multiprocessing. In this case, a server can answer more than one request at a time.
- Some popular web servers include Apache and Microsoft Internet Information Server.

STANDARD CLIENT-SERVER APPLICATIONS
World Wide Web

Uniform Resource Locator (URL)

- A web page, as a file, needs to have a unique identifier to distinguish it from other web pages.
- To define a web page, we need three identifiers: host, port, and path.
- Before defining the web page, we need to tell the browser what client server application we want to use, which is called the protocol.
- This means we need four identifiers to define the web page. The first is the type of vehicle to be used to fetch the web page; the last three make up the combination that defines the destination object (web page).

STANDARD CLIENT-SERVER APPLICATIONS
World Wide Web

- Protocol:
  - The first identifier is the abbreviation for the client-server program that we need in order to access the web page.
  - Although most of the time the protocol is HTTP (HyperText Transfer Protocol), we can also use other protocols such as FTP (File Transfer Protocol).
- Host:
  - The host identifier can be the IP address of the server or the unique name given to the server.
  - IP addresses can be defined in dotted decimal notations (such as 64.23.56.17); the name is normally the domain name that uniquely defines the host, such as forouzan.com.

STANDARD CLIENT-SERVER APPLICATIONS
World Wide Web

- Port:
  - The port, a 16-bit integer, is normally predefined for the client-server application.
  - For example, if the HTTP protocol is used for accessing the web page, the well-known port number is 80.
- Path:
  - The path identifies the location and the name of the file in the underlying operating system.
  - The format of this identifier normally depends on the operating system.
  - In UNIX, a path is a set of directory names followed by the file name, all separated by a slash.
  - For example, /top/next/last/myfile is a path that uniquely defines a file named myfile, stored in the directory last, which itself is part of the directory next, which itself is under the directory top.

STANDARD CLIENT-SERVER APPLICATIONS
World Wide Web

To combine these four pieces together, the uniform resource locator (URL) has been designed; it uses three different separators between the four pieces as shown below:

protocol://host/path Used most of the time

protocol://host:port/path Used when port number is needed

- Example: The URL http://www.mhhe.com/compsci/forouzan/ defines the web page.
- The string www.mhhe.com is the name of the computer in the McGraw-Hill company (the three letters www are part of the host name and are added to the commercial host).
- The path is compsci/forouzan/, which defines Forouzan's web page under the directory compsci (computer science).

STANDARD CLIENT-SERVER APPLICATIONS
World Wide Web

Web Documents

- The documents in the WWW can be grouped into three broad categories: static, dynamic, and active.

Static Documents

- Static documents are fixed-content documents that are created and stored in a server. Static documents are prepared using one of the several languages: Hypertext Markup Language (HTML), Extensible Markup Language (XML), Extensible Style Language (XSL), and Extensible Hypertext Markup Language (XHTML).

STANDARD CLIENT-SERVER APPLICATIONS
World Wide Web

Web Documents

- The documents in the WWW can be grouped into three broad categories: static, dynamic, and active.

Dynamic Documents

- A dynamic document is created by a web server whenever a browser requests the document.
- Although the Common Gateway Interface (CGI) was used to retrieve a dynamic document in the past, today's options include one of the scripting languages such as Java Server Pages (JSP), which uses the Java language for scripting, or Active Server Pages (ASP), a Microsoft product that uses Visual Basic language for scripting, or ColdFusion, which embeds queries in a Structured Query Language (SQL) database in the HTML document.

STANDARD CLIENT-SERVER APPLICATIONS
World Wide Web

Web Documents

- The documents in the WWW can be grouped into three broad categories: static, dynamic, and active.

Active Documents

- For many applications, we need a program or a script to be run at the client site. These are called active documents.
- For example, suppose we want to run a program that creates animated graphics on the screen or a program that interacts with the user. The program definitely needs to be run at the client site where the animation or interaction takes place.
- When a browser requests an active document, the server sends a copy of the document or a script. The document is then run at the client (browser) site.
- One way to create an active document is to use Java applets, a program written in Java on the server. Another way is to use JavaScripts.

STANDARD CLIENT-SERVER APPLICATIONS

HyperText Transfer Protocol (HTTP)

- The HyperText Transfer Protocol (HTTP) is a protocol that is used to define how the client-server programs can be written to retrieve web pages from the Web.
- An HTTP client sends a request; an HTTP server returns a response.
- The server uses the port number 80; the client uses a temporary port number.
- HTTP uses the services of TCP, which is a connection-oriented and reliable protocol.
- The client and server do not need to worry about errors in messages exchanged or loss of any message, because the TCP is reliable and will take care of this matter.

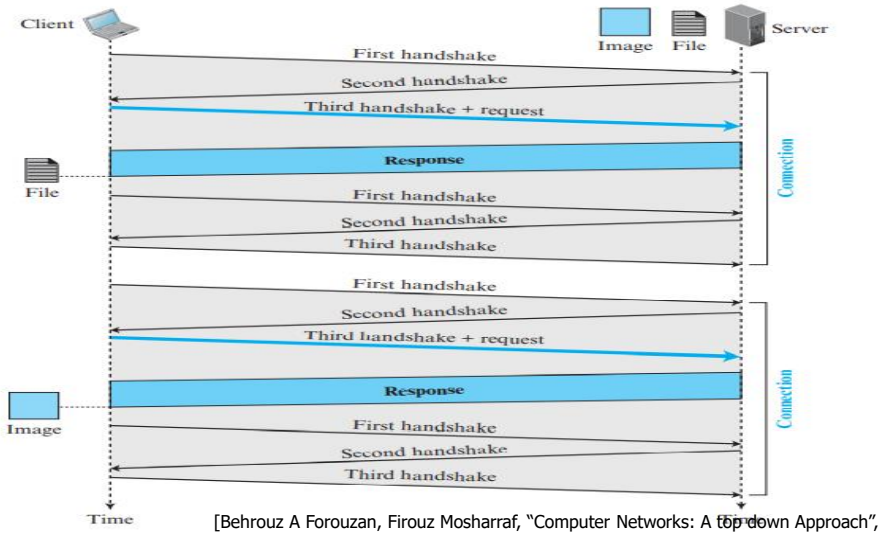
Nonpersistent versus Persistent Connections

- The hypertext concept embedded in web page documents may require several requests and responses.
- If the web pages, objects to be retrieved, are located on different servers, we do not have any other choice than to create a new TCP connection for retrieving each object.
- If some of the objects are located on the same server, we have two choices: to retrieve each object using a new TCP connection or to make a TCP connection and retrieve them all.
- The first method is referred to as nonpersistent connections, the second as persistent connections.
- HTTP specified nonpersistent connections, while persistent connections is the default in version 1.1, but it can be changed by the user.

Nonpersistent Connections

- In a nonpersistent connection, one TCP connection is made for each request/response.
- The following lists the steps in this strategy:
  1. The client opens a TCP connection and sends a request.
  2. The server sends the response and closes the connection.
  3. The client reads the data until it encounters an end-of-file marker; it then closes the connection.
- In this strategy, if a file contains links to N different pictures in different files (all located on the same server), the connection must be opened and closed N + 1 times.
- The nonpersistent strategy imposes high overhead on the server because the server needs N + 1 different buffers each time a connection is opened.

Figure 1.36 *Example* Nonpersistent Connections



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

- The client needs to access a file that contains one link to an image.

- Figure 1.36 shows an example of a nonpersistent connection.
- The client needs to access a file that contains one link to an image. The text file and image are located on the same server.
- Here we need two connections.
- For each connection, TCP requires at least three handshake messages to establish the connection.
- After the connection is established, the object can be transferred.
- After receiving an object, another three handshake messages are needed to terminate the connection.
- This means that the client and server are involved in two connection establishments and two connection terminations.
- If the transaction involves retrieving 10 or 20 objects, the round trip times spent for these handshakes add up to a big overhead

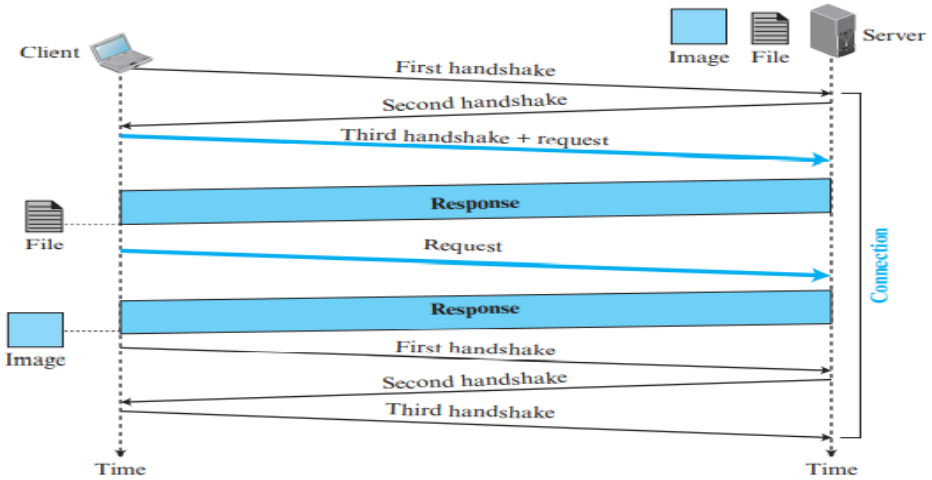
Persistent Connections

- HTTP version 1.1 specifies a persistent connection by default.
- In a persistent connection, the server leaves the connection open for more requests after sending a response.
- The server can close the connection at the request of a client or if a time-out has been reached.
- The sender usually sends the length of the data with each response.

Persistent Connections

- There are some occasions when the sender does not know the length of the data. This is the case when a document is created dynamically or actively.
- In these cases, the server informs the client that the length is not known and closes the connection after sending the data so the client knows that the end of the data has been reached.
- Time and resources are saved using persistent connections.
- Only one set of buffers and variables needs to be set for the connection at each site.
- The round trip time for connection establishment and connection termination is saved.

Figure 1.37 *Example* Persistent Connections



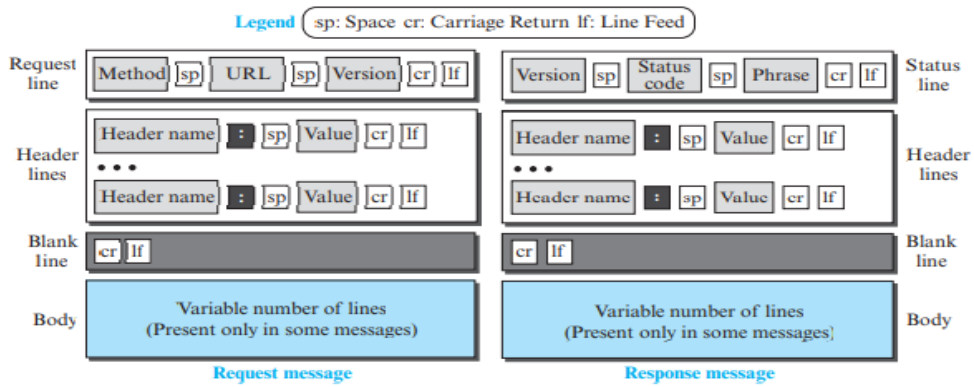
[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

- Only one connection establishment and connection termination is used, but the request for the image is sent separately.

Message Formats

- The HTTP protocol defines the format of the request and response messages.

Figure 1.38 *Formats of the request and response messages*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

Message Formats

- The first section in the request message is called the request line; the first section in the response message is called the status line.
- The other three sections have the same names in the request and response messages.
- However, the similarities between these sections are only in the names; they may have different contents.

Request Message

- The first line in a request message is called a request line.
- There are three fields in this line separated by one space and terminated by two characters (carriage return and line feed).
- The fields are called method, URL, and version.
- The method field defines the request types.
- In version 1.1 of HTTP, several methods are defined, as shown in Table 1.2.

Table 1.2 *Methods*

Method	*Action*
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
PUT	Sends a document from the client to the server
POST	Sends some information from the client to the server
TRACE	Echoes the incoming request
DELETE	Removes the web page
CONNECT	Reserved
OPTIONS	Inquires about available options

[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

Request Message

- The second field, URL defines the address and name of the corresponding web page.
- The third field, version, gives the version of the protocol; the most current version of HTTP is 1.1.

Request Message

- After the request line, we can have zero or more request header lines.
- Each header line sends additional information from the client to the server.
- Each header line has a header name, a colon, a space and a header value.

Request Message

- Table 1.3 shows some header names commonly used in a request.
- The value field defines the values associated with each header name.
- The list of values can be found in the corresponding RFCs.
- The body can be present in a request message. Usually, it contains the comment to be sent or the file to be published on the website when the method is PUT or POST.

Table 1.3 *Request Header Names*

Header	*Description*
User-agent	Identifies the client program
Accept	Shows the media format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows what permissions the client has
Host	Shows the host and port number of the client
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Cookie	Returns the cookie to the server (explained later)
If-Modified-Since	If the file is modified since a specific date

[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS
HyperText Transfer Protocol (HTTP)

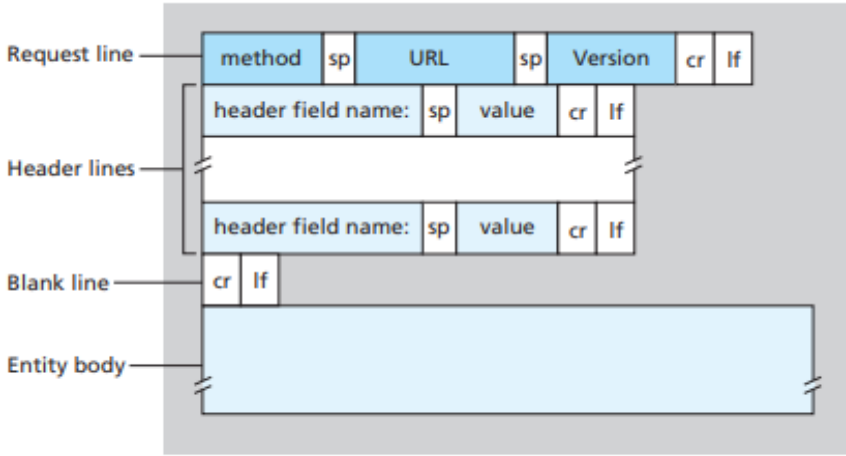


Figure 1.39 General format of an HTTP request message

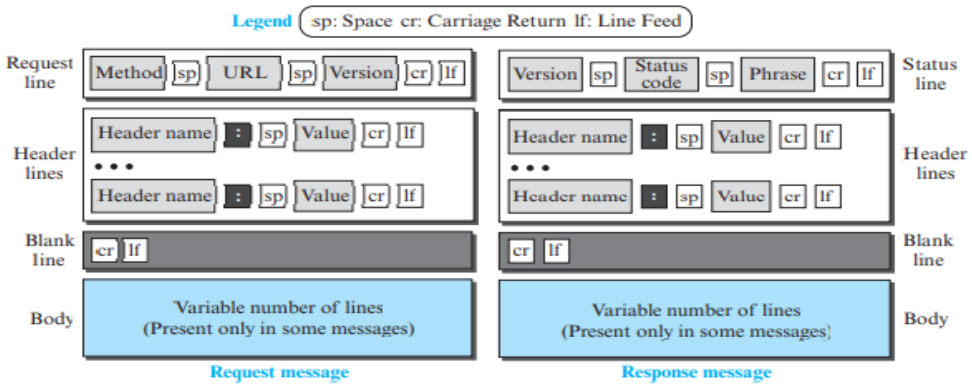
James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

STANDARD CLIENT-SERVER APPLICATIONS
HyperText Transfer Protocol (HTTP)

Response Message

The format of the response message is also shown in Figure 1.38.

Figure 1.38 *Formats of the request and response messages*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS
HyperText Transfer Protocol (HTTP)

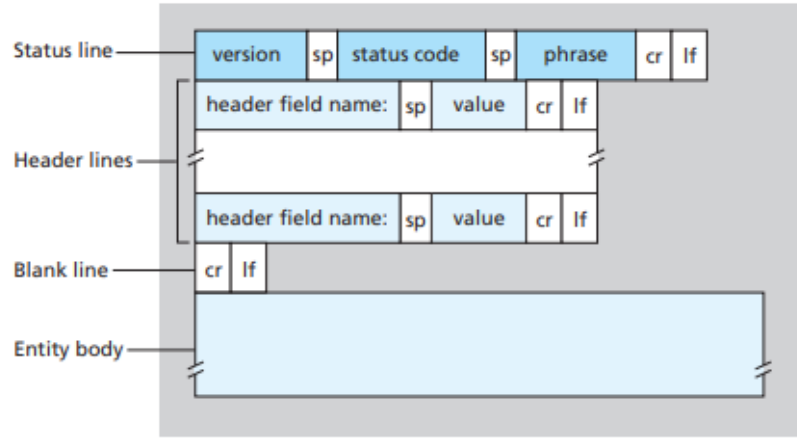


Figure 1.40 General format of an HTTP response message

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Response Message

- A response message consists of a status line, header lines, a blank line, and sometimes a body.
- The first line in a response message is called the status line.
- There are three fields in this line separated by spaces and terminated by a carriage return and line feed.
- The first field defines the version of HTTP protocol, currently 1.1.

Response Message

- The status code field defines the status of the request. It consists of three digits.
- Whereas the codes in the 100 range are only informational, the codes in the 200 range indicate a successful request. The codes in the 300 range redirect the client to another URL, and the codes in the 400 range indicate an error at the client site. Finally, the codes in the 500 range indicate an error at the server site.
- The status phrase explains the status code in text form.

Response Message

- After the status line, we can have zero or more response header lines.
- Each header line sends additional information from the server to the client.
- For example, the sender can send extra information about the document.
- Each header line has a header name, a colon, a space, and a header value.

Response Message

- Table 1.4 shows some header names commonly used in a response message.
- The body contains the document to be sent from the server to the client.
- The body is present unless the response is an error message.

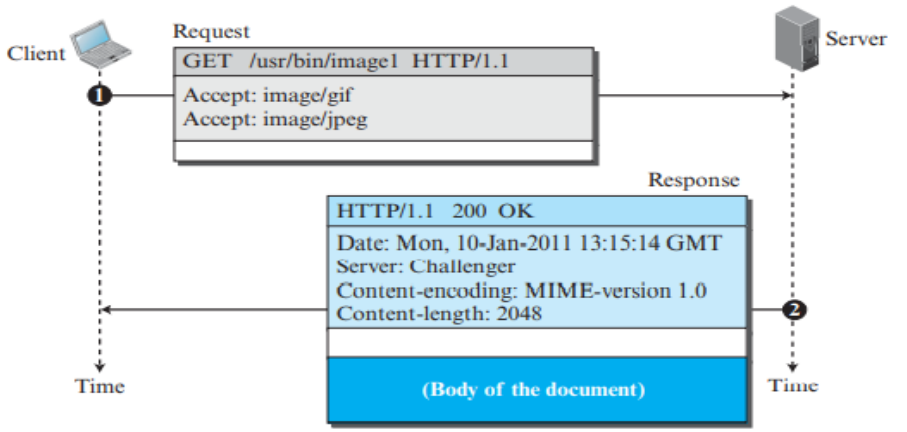
Table 1.4 *Response Header Names*

Header	*Description*
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Server	Gives information about the server
Set-Cookie	The server asks the client to save a cookie
Content-Encoding	Specifies the encoding scheme
Content-Language	Specifies the language
Content-Length	Shows the length of the document
Content-Type	Specifies the media type
Location	To ask the client to send the request to another site
Accept-Ranges	The server will accept the requested byte-ranges
Last-modified	Gives the date and time of the last change

STANDARD CLIENT-SERVER APPLICATIONS
HyperText Transfer Protocol (HTTP)

- An example retrieves a document (see Figure 1.41)

Figure 1.41 *Example*

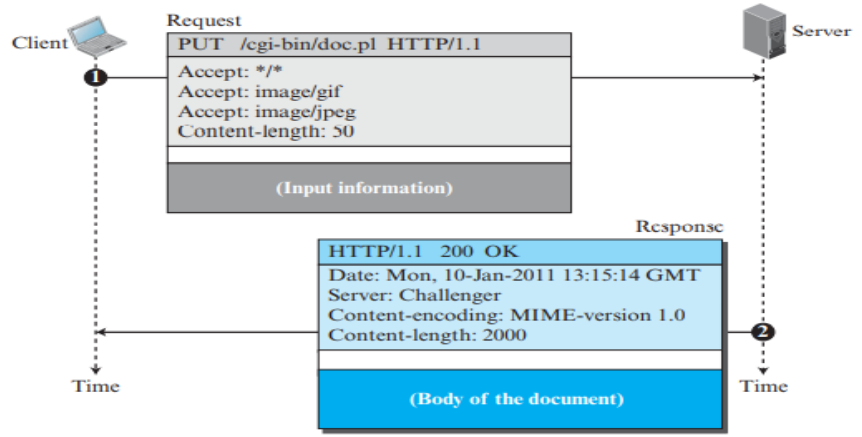


[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS
HyperText Transfer Protocol (HTTP)

- Another example, the client wants to send a web page to be posted on the server(see Figure 1.42)

Figure 1.42 *Example*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS
HyperText Transfer Protocol (HTTP)

Cookies

- The World Wide Web was originally designed as a stateless entity. A client sends a request; a server responds. Their relationship is over.
- The original purpose of the Web, retrieving publicly available documents, exactly fits this design.
- Today the Web has other functions that need to remember some information about the clients like:
  - Websites are being used as electronic stores that allow users to browse through the store, select wanted items, put them in an electronic cart, and pay at the end with a credit card.
  - Some websites need to allow access to registered clients only.
  - Some websites are used as portals: the user selects the web pages he wants to see.
  - Some websites are just advertising agency. For these purposes, the cookie mechanism was devised.

STANDARD CLIENT-SERVER APPLICATIONS
HyperText Transfer Protocol (HTTP)

Cookies

Creating and Storing Cookies

- The creation and storing of cookies depend on the implementation; however, the principle is the same.
  1. When a server receives a request from a client, it stores information about the client in a file or a string. The information may include the domain name of the client, the contents of the cookie (information the server has gathered about the client such as name, registration number, and so on), a timestamp, and other information depending on the implementation.
  2. The server includes the cookie in the response that it sends to the client.
  3. When the client receives the response, the browser stores the cookie in the cookie directory, which is sorted by the server domain name.

STANDARD CLIENT-SERVER APPLICATIONS
HyperText Transfer Protocol (HTTP)

Cookies

Using Cookies

- When a client sends a request to a server, the browser looks in the cookie directory to see if it can find a cookie sent by that server.
- If found, the cookie is included in the request.
- When the server receives the request, it knows that this is an old client, not a new one.
- Note that the contents of the cookie are never read by the browser or disclosed to the user.
- It is a cookie made by the server and eaten by the server.

STANDARD CLIENT-SERVER APPLICATIONS
HyperText Transfer Protocol (HTTP)

Web Caching: Proxy Server

- HTTP supports proxy servers.
- A proxy server is a computer that keeps copies of responses to recent requests.
- The HTTP client sends a request to the proxy server.
- The proxy server checks its cache.
- If the response is not stored in the cache, the proxy server sends the request to the corresponding server.
- Incoming responses are sent to the proxy server and stored for future requests from other clients.
- The proxy server reduces the load on the original server, decreases traffic, and improves latency.
- However, to use the proxy server, the client must be configured to access the proxy instead of the target server. Note that the proxy server acts as both server and client

STANDARD CLIENT-SERVER APPLICATIONS
HyperText Transfer Protocol (HTTP)

Web Caching: Proxy Server

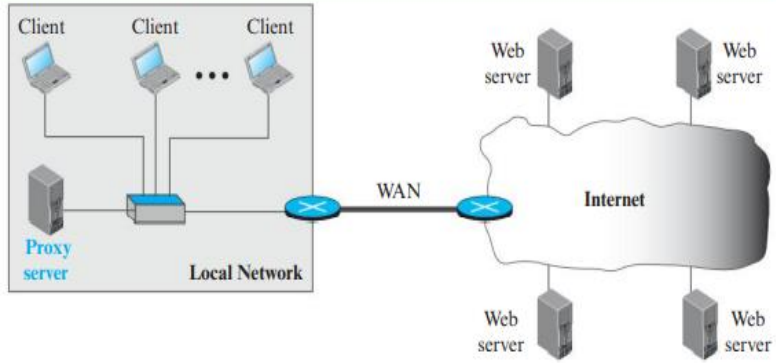
Proxy Server Location

- The proxy servers are normally located at the client site.
- This means that we can have a hierarchy of proxy servers as shown below:
  1. A client computer can also be used as a proxy server, in a small capacity, that stores responses to requests often invoked by the client.
  2. In a company, a proxy server may be installed on the computer LAN to reduce the load going out of and coming into the LAN.
  3. An ISP with many customers can install a proxy server to reduce the load going out of and coming into the ISP network.

STANDARD CLIENT-SERVER APPLICATIONS
HyperText Transfer Protocol (HTTP)

Web Caching: Proxy Server

Figure 1.43 *Example of a proxy server*



STANDARD CLIENT-SERVER APPLICATIONS
HyperText Transfer Protocol (HTTP)

HTTP Security

- HTTP per se does not provide security.
- However, HTTP can be run over the Secure Socket Layer (SSL).
- In this case, HTTP is referred to as HTTPS.

STANDARD CLIENT-SERVER APPLICATIONS

FTP

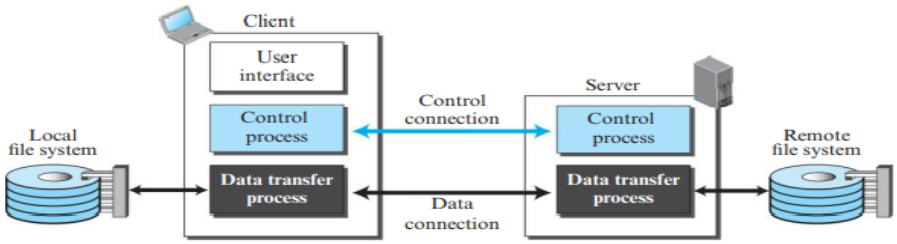
- File Transfer Protocol (FTP) is the standard protocol provided by TCP/IP for copying a file from one host to another.
- Although transferring files from one system to another seems simple and straightforward, some problems must be dealt with first. For example, two systems may use different file name conventions. Two systems may have different ways to represent data. Two systems may have different directory structures.
- All of these problems have been solved by FTP in a very simple and elegant approach.
- Although we can transfer files using HTTP, FTP is a better choice to transfer large files or to transfer files using different formats.

STANDARD CLIENT-SERVER APPLICATIONS

FTP

- Figure 1.44 shows the basic model of FTP.
- The client has three components: user interface, client control process, and the client data transfer process.
- The server has two components: the server control process and the server data transfer process.

Figure 1.44 *FTP*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS

FTP

- The control connection is made between the control processes.
- The data connection is made between the data transfer processes.
- Separation of commands and data transfer makes FTP more efficient.
- The control connection uses very simple rules of communication. We need to transfer only a line of command or a line of response at a time.
- The data connection needs more complex rules due to the variety of data types transferred

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Lifetimes of Two Connections

- The two connections in FTP have different lifetimes.
- The control connection remains connected during the entire interactive FTP session.
- The data connection is opened and then closed for each file transfer activity. It opens each time commands that involve transferring files are used, and it closes when the file is transferred.
- When a user starts an FTP session, the control connection opens. While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.
- FTP uses two well-known TCP ports: port 21 is used for the control connection, and port 20 is used for the data connection.

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Control Connection

- For control communication, FTP uses the NVT ASCII character set as used by TELNET.
- Communication is achieved through commands and responses.
- This simple method is adequate for the control connection because we send one command (or response) at a time. Each line is terminated with a two-character (carriage return and line feed) end-of-line token.
- During the control connection, commands are sent from the client to the server and responses are sent from the server to the client.
- Commands, which are sent from the FTP client control process, are in the form of ASCII uppercase, which may or may not be followed by an argument.

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Control Connection

- Some of the most common commands are shown in Table 1.5.

Table 1.5 *Some FTP commands*

Command	*Argument(s)*	*Description*
ABOR		Abort the previous command
CDUP		Change to parent directory
CWD	Directory name	Change to another directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories or files
MKD	Directory name	Create a new directory
PASS	User password	Password
PASV		Server chooses a port
PORT	port identifier	Client chooses a port
PWD		Display name of current directory
QUIT		Log out of the system
RETR	File name(s)	Retrieve files; files are transferred from server to client
RMD	Directory name	Delete a directory
RNFR	File name (old)	Identify a file to be renamed

[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Control Connection

- Every FTP command generates at least one response.
- A response has two parts: a three-digit number followed by text.
- The numeric part defines the code; the text part defines needed parameters or further explanations.
- The first digit defines the status of the command. The second digit defines the area in which the status applies. The third digit provides additional information.
- Table 1.6 shows some common responses.

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Control Connection

Table 1.6 *Some responses in FTP*

Code	Description	Code	Description
125	Data connection open	250	Request file action OK
150	File status OK	331	User name OK; password is needed
200	Command OK	425	Cannot open data connection
220	Service ready	450	File action not taken; file not available
221	Service closing	452	Action aborted; insufficient storage
225	Data connection open	500	Syntax error; unrecognized command
226	Closing data connection	501	Syntax error in parameters or arguments
230	User login OK	530	User not logged in

[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Data Connection

- The data connection uses the well-known port 20 at the server site.
- The creation of a data connection is different from the control connection.
- The following shows the steps:
  1. The client issues a passive open using an ephemeral port. This must be done by the client because it is the client that issues the commands for transferring files.
  2. The client sends this port number to the server using the PORT command.
  3. The server receives the port number and issues an active open using the well known port 20 and the received ephemeral port number.

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Communication over Data Connection

- The purpose and implementation of the data connection are different from those of the control connection.
- We want to transfer files through the data connection.
- The client must define the type of file to be transferred, the structure of the data, and the transmission mode.
- Before sending the file through the data connection, we prepare for transmission through the control connection.
- The heterogeneity problem is resolved by defining three attributes of communication: file type, data structure, and transmission mode.

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Data Structure

- FTP can transfer a file across the data connection using one of the following interpretations of the structure of the data: file structure, record structure, or page structure.
- The file structure format has no structure. It is a continuous stream of bytes.
- In the record structure, the file is divided into records. This can be used only with text files.
- In the page structure, the file is divided into pages, with each page having a page number and a page header. The pages can be stored and accessed randomly or sequentially.

File Type

- FTP can transfer one of the following file types across the data connection: ASCII file, EBCDIC file, or image file.

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Transmission Mode

- FTP can transfer a file across the data connection using one of the following three transmission modes: stream mode, block mode, or compressed mode.
- The stream mode is the default mode; data are delivered from FTP to TCP as a continuous stream of bytes.
- In the block mode, data can be delivered from FTP to TCP in blocks. Each block is preceded by a 3-byte header.
- The first byte is called the block descriptor; the next two bytes define the size of the block in bytes.

STANDARD CLIENT-SERVER APPLICATIONS
FTP

File Transfer

- File transfer occurs over the data connection under the control of the commands sent over the control connection.

- File transfer in FTP means one of three things: retrieving a file (server to client), storing a file (client to server), and directory listing (server to client).

STANDARD CLIENT-SERVER APPLICATIONS
FTP

Security for FTP

- The FTP protocol was designed when security was not a big issue.
- Although FTP requires a password, the password is sent in plaintext (unencrypted), which means it can be intercepted and used by an attacker.
- The data transfer connection also transfers data in plaintext, which is insecure.
- To be secure, one can add a Secure Socket Layer between the FTP application layer and the TCP layer. In this case FTP is called SSL-FTP.

STANDARD CLIENT-SERVER APPLICATIONS

Electronic Mail

- Electronic mail (or e-mail) allows users to exchange messages.
- First, e-mail is considered a one-way transaction. When Alice sends an e-mail to Bob, she may expect a response, but this is not a mandate. Bob may or may not respond. If he does respond, it is another one-way transaction.
- Second, it is neither feasible nor logical for Bob to run a server program and wait until someone sends an e-mail to him. Bob may turn off his computer when he is not using it.
- This means that the idea of client/ server programming should be implemented in another way: using some intermediate computers (servers). The users run only client programs when they want and the intermediate servers apply the client/server paradigm.

STANDARD CLIENT-SERVER APPLICATIONS
Electronic Mail

Architecture

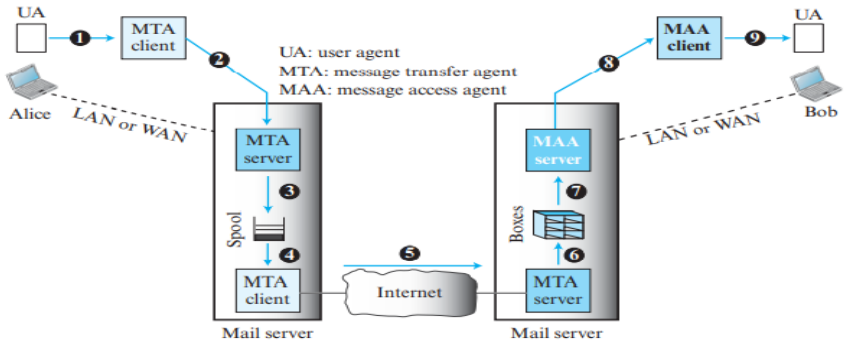
- In the common scenario, the sender and the receiver of the e-mail, Alice and Bob respectively, are connected via a LAN or a WAN to two mail servers.
- The administrator has created one mailbox for each user where the received messages are stored.
- A mailbox is part of a server hard drive, a special file with permission restrictions.
- Only the owner of the mailbox has access to it.
- The administrator has also created a queue (spool) to store messages waiting to be sent.

STANDARD CLIENT-SERVER APPLICATIONS
Electronic Mail

Architecture

- A simple e-mail from Alice to Bob takes nine different steps, as shown in the figure.
- Alice and Bob use three different agents: a User Agent (UA), a Mail Transfer Agent (MTA) and a Message Access Agent (MAA).

Figure 1.45 *Common scenario*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS
Electronic Mail

Architecture

- When Alice needs to send a message to Bob, she runs a UA program to prepare the message and send it to her mail server.
- The mail server at her site uses a queue (spool) to store messages waiting to be sent.
- The message, however, needs to be sent through the Internet from Alice's site to Bob's site using an MTA.
- Here two message transfer agents are needed: one client and one server.
- Like most client-server programs on the Internet, the server needs to run all the time because it does not know when a client will ask for a connection.

STANDARD CLIENT-SERVER APPLICATIONS
Electronic Mail

Architecture

- The client, on the other hand, can be triggered by the system when there is a message in the queue to be sent.
- The user agent at the Bob site allows Bob to read the received message.
- Bob later uses an MAA client to retrieve the message from an MAA server running on the second server.
- There are two important points we need to emphasize. First, Bob cannot bypass the mail server and use the MTA server directly
- Second, Bob needs another pair of client-server programs: message access programs.
- This is because an MTA client-server program is a push program: the client pushes the message to the server. Bob needs a pull program. The client needs to pull the message from the server.

STANDARD CLIENT-SERVER APPLICATIONS
Electronic Mail

- The electronic mail system needs two UAs, two pairs of MTAs (client and server), and a pair of MAAs (client and server).

STANDARD CLIENT-SERVER APPLICATIONS
Electronic Mail

User Agent

- The first component of an electronic mail system is the user agent (UA).
- It provides service to the user to make the process of sending and receiving a message easier.
- A user agent is a software package (program) that composes, reads, replies to, and forwards messages.
- It also handles local mailboxes on the user computers.
- There are two types of user agents: command-driven and GUI-based.

STANDARD CLIENT-SERVER APPLICATIONS
Electronic Mail

User Agent

- Command driven user agents belong to the early days of electronic mail. A command-driven user agent normally accepts a one character command from the keyboard to perform its task. Some examples of command driven user agents are mail, pine, and elm.

- Modern user agents are GUI-based. They contain graphical user interface (GUI) components that allow the user to interact with the software by using both the keyboard and the mouse. They have graphical components such as icons, menu bars, and windows that make the services easy to access. Some examples of GUI-based user agents are Eudora and Outlook

STANDARD CLIENT-SERVER APPLICATIONS
Electronic Mail

Sending Mail

- To send mail, the user, through the UA, creates mail that looks very similar to postal mail.
- It has an envelope and a message.
- The envelope usually contains the sender address, the receiver address, and other information.
- The message contains the header and the body.
- The header of the message defines the sender, the receiver, the subject of the message, and some other information.
- The body of the message contains the actual information to be read by the recipient.

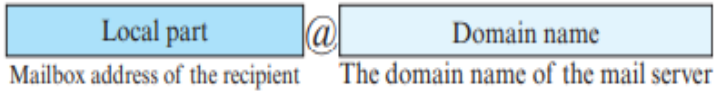
Receiving Mail

- The user agent is triggered by the user (or a timer).
- If a user has mail, the UA informs the user with a notice.
- If the user is ready to read the mail, a list is displayed in which each line contains a summary of the information about a particular message in the mailbox.
- The summary usually includes the sender mail address, the subject, and the time the mail was sent or received.
- The user can select any of the messages and display its contents on the screen.

Addresses

- To deliver mail, a mail handling system must use an addressing system with unique addresses.
- In the Internet, the address consists of two parts: a local part and a domain name, separated by an @ sign (see Figure 1.46).
- Electronic mail allows one name, an alias, to represent several different e-mail addresses; this is called a mailing list.

Figure 1.46 *E-mail address*

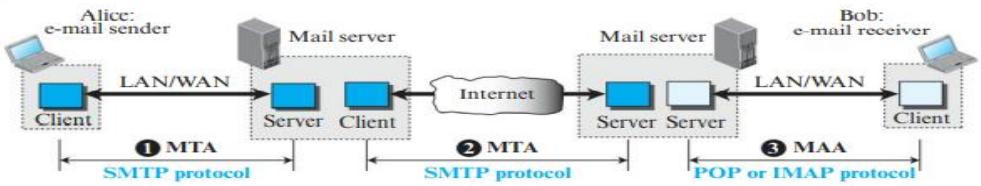


[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

Message Transfer Agent: SMTP

- The formal protocol that defines the MTA client and server in the Internet is called Simple Mail Transfer Protocol (SMTP).
- SMTP is used two times, between the sender and the sender's mail server and between the two mail servers.
- Another protocol is needed between the mail server and the receiver.
- SMTP simply defines how commands and responses must be sent back and forth.

Figure 1.47 *Protocols used in electronic mail* BACK



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

Message Transfer Agent: SMTP

Commands and Responses

- SMTP uses commands and responses to transfer messages between an MTA client and an MTA server.
- The command is from an MTA client to an MTA server; the response is from an MTA server to the MTA client.
- Each command or reply is terminated by a two character (carriage return and line feed) end-of-line token.
- Commands are sent from the client to the server.
- The format of a command is shown below:
  - Keyword: argument(s)

  It consists of a keyword followed by zero or more arguments.
- SMTP defines 14 commands, listed in Table 1.7.

Table 1.7 *SMTP Commands*

Keyword	*Argument(s)*	*Description*
HELO	Sender's host name	Identifies itself
MAIL FROM	Sender of the message	Identifies the sender of the message
RCPT TO	Intended recipient	Identifies the recipient of the message
DATA	Body of the mail	Sends the actual message
QUIT		Terminates the message

Responses

- Responses are sent from the server to the client.
- A response is a three digit code that may be followed by additional textual information.
- Table 1.8 shows the most common response types.

Table 1.8 *Responses*

Code	*Description*
	Positive Completion Reply
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
	Positive Intermediate Reply
354	Start mail input
	Transient Negative Completion Reply
421	Service not available
450	Mailbox not available

Mail Transfer Phases

- The process of transferring a mail message occurs in three phases: connection establishment, mail transfer, and connection termination.

Connection Establishment

- After a client has made a TCP connection to the well known port 25, the SMTP server starts the connection phase.
- This phase involves three steps:
  1. The server sends code 220 (service ready) to tell the client that it is ready to receive mail. If the server is not ready, it sends code 421 (service not available).
  2. The client sends the HELO message to identify itself, using its domain name address. This step is necessary to inform the server of the domain name of the client.
  3. The server responds with code 250 (request command completed) or some other code depending on the situation.

Mail Transfer Phases

Message Transfer

- After connection has been established between the SMTP client and server, a single message between a sender and one or more recipients can be exchanged.
- This phase involves eight steps. Steps 3 and 4 are repeated if there is more than one recipient.
  1. The client sends the MAIL FROM message to introduce the sender of the message. It includes the mail address of the sender (mailbox and the domain name). This step is needed to give the server the return mail address for returning errors and reporting messages.
  2. The server responds with code 250 or some other appropriate code.
  3. The client sends the RCPT TO (recipient) message, which includes the mail address of the recipient.

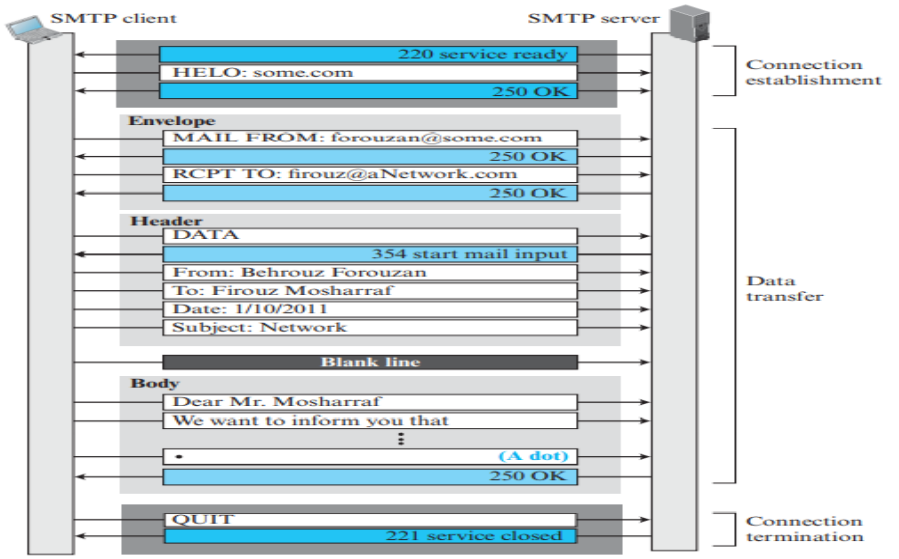
Mail Transfer Phases
Message Transfer

4. The server responds with code 250 or some other appropriate code.
5. The client sends the DATA message to initialize the message transfer.
6. The server responds with code 354 (start mail input) or some other appropriate message.
7. The client sends the contents of the message in consecutive lines. Each line is terminated by a two-character end-of-line token (carriage return and line feed). The message is terminated by a line containing just one period.
8. The server responds with code 250 (OK) or some other appropriate code.

Mail Transfer Phases
Connection Termination

- After the message is transferred successfully, the client terminates the connection.
- This phase involves two steps.
  1. The client sends the QUIT command.
  2. The server responds with code 221 or some other appropriate code

Figure 1.48 *Example*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

Message Access Agent: POP and IMAP

- The first and second stages of mail delivery use SMTP. However, SMTP is not involved in the third stage because SMTP is a push protocol; it pushes the message from the client to the server.
- The third stage needs a pull protocol; the client must pull messages from the server. The direction of the bulk data is from the server to the client.
- The third stage uses a message access agent.
- Currently two message access protocols are available: Post Office Protocol, version 3 (POP3) and Internet Mail Access Protocol, version 4 (IMAP4).
- Figure 1.47 shows the position of these two protocols

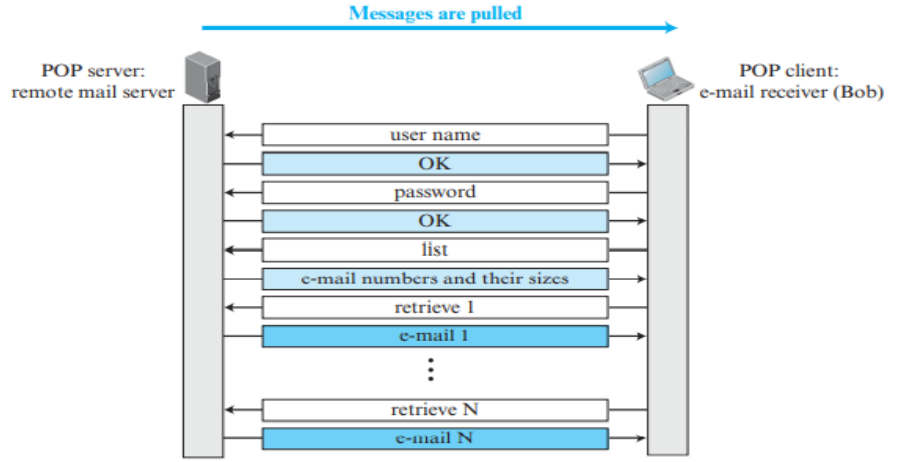
STANDARD CLIENT-SERVER APPLICATIONS
Electronic Mail

POP3

- Post Office Protocol, version 3 (POP3) is simple but limited in functionality.
- The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.
- Mail access starts with the client when the user needs to download its e-mail from the mailbox on the mail server.
- The client opens a connection to the server on TCP port 110.
- It then sends its user name and password to access the mailbox.
- The user can then list and retrieve the mail messages, one by one.

STANDARD CLIENT-SERVER APPLICATIONS
Electronic Mail — POP3

Figure 1.49 *POP3*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS
Electronic Mail — POP3

- POP3 has two modes: the delete mode and the keep mode.
- In the delete mode, the mail is deleted from the mailbox after each retrieval.
- In the keep mode, the mail remains in the mailbox after retrieval.
- The delete mode is normally used when the user is working at her permanent computer and can save and organize the received mail after reading or replying.
- The keep mode is normally used when the user accesses her mail away from her primary computer (for example, from a laptop). The mail is read but kept in the system for later retrieval and organizing.
- Internet Mail Access Protocol, version 4 (IMAP4)
- Multipurpose Internet Mail Extensions (MIME)

STANDARD CLIENT-SERVER APPLICATIONS

Domain Name System (DNS)

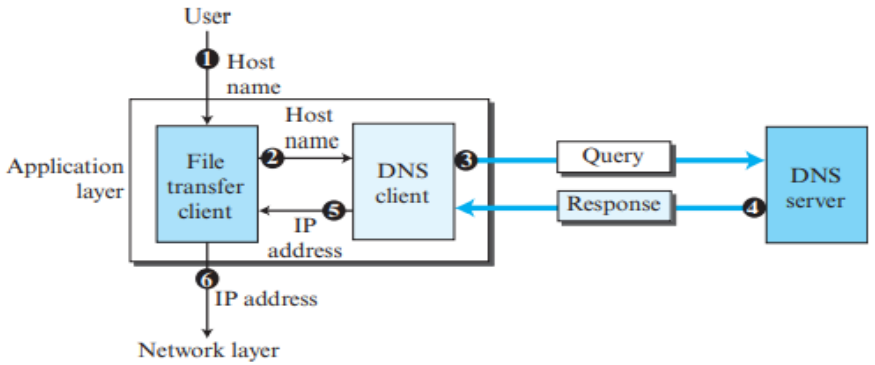
- Since the Internet is so huge today, a central directory system cannot hold all the mapping.
- In addition, if the central computer fails, the whole communication network will collapse.
- A better solution is to distribute the information among many computers in the world.
- In this method, the host that needs mapping can contact the closest computer holding the needed information.
- This method is used by the Domain Name System (DNS).

STANDARD CLIENT-SERVER APPLICATIONS

Domain Name System (DNS)

- Figure 1.50 shows how TCP/IP uses a DNS client and a DNS server to map a name to an address.

Figure 1.50 *Purpose of DNS*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS

Domain Name System (DNS)

The following six steps map the host name to an IP address:

1. The user passes the host name to the file transfer client.
2. The file transfer client passes the host name to the DNS client.
3. Each computer, after being booted, knows the address of one DNS server. The DNS client sends a message to a DNS server with a query that gives the file transfer server name using the known IP address of the DNS server.
4. The DNS server responds with the IP address of the desired file transfer server.
5. The DNS client passes the IP address to the file transfer server.
6. The file transfer client now uses the received IP address to access the file transfer server.

STANDARD CLIENT-SERVER APPLICATIONS
Domain Name System (DNS)

Name Space

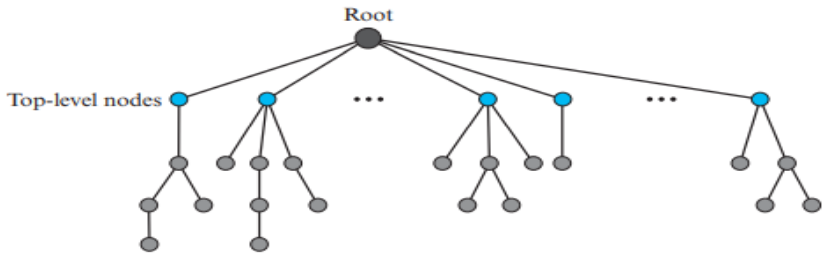
- A name space that maps each address to a unique name can be organized in two ways: flat or hierarchical.
- In a flat name space, a name is assigned to an address. A name in this space is a sequence of characters without structure.
- In a hierarchical name space, each name is made of several parts. The first part can define the nature of the organization, the second part can define the name of an organization, the third part can define departments in the organization, and so on.
- In this case, the authority to assign and control the name spaces can be decentralized.

STANDARD CLIENT-SERVER APPLICATIONS
Domain Name System (DNS)

Domain Name Space

- To have a hierarchical name space, a domain name space was designed.
- In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127 (see Figure 1.51).

Figure 1.51 *Domain name space*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

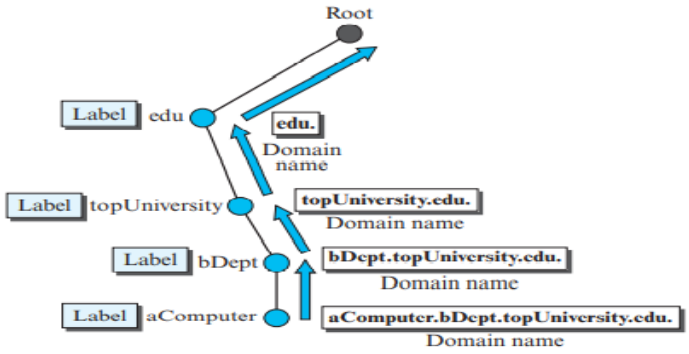
Label

- Each node in the tree has a label, which is a string with a maximum of 63 characters.
- The root label is a null string (empty string).
- DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

Domain Name

- Each node in the tree has a domain name.
- A full domain name is a sequence of labels separated by dots (.).
- The domain names are always read from the node up to the root.
- The last label is the label of the root (null).
- This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.
- Figure 1.52 shows some domain names.

Figure1.52 *Domain names and labels*



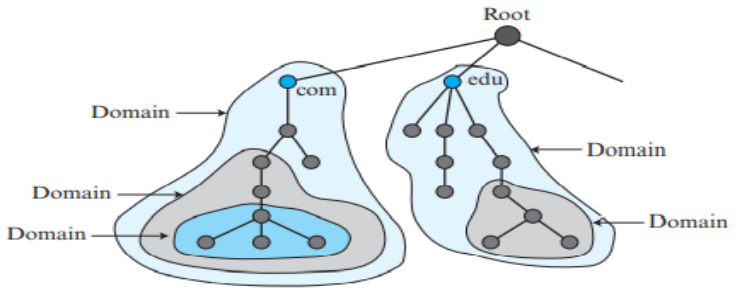
[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

- If a label is terminated by a null string, it is called a fully qualified domain name (FQDN).
- If a label is not terminated by a null string, it is called a partially qualified domain name (PQDN).

Domain

- A domain is a subtree of the domain name space.
- The name of the domain is the name of the node at the top of the subtree.
- Note that a domain may itself be divided into domains.

Figure 1.53 *Domains*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS
Domain Name System (DNS)

Distribution of Name Space

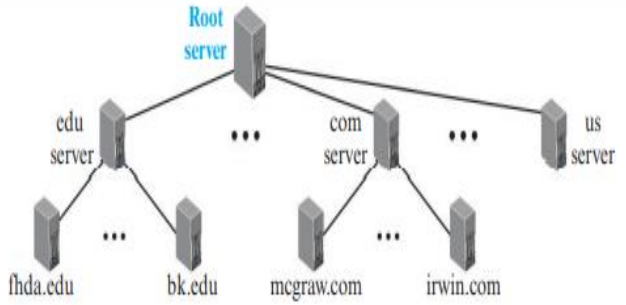
- The information contained in the domain name space must be stored.
- It is very inefficient and also not reliable to have just one computer store such a huge amount of information.

Hierarchy of Name Servers

- The solution to these problems is to distribute the information among many computers called DNS servers.
- One way to do this is to divide the whole space into many domains.
- DNS allows domains to be divided further into smaller domains (subdomains).
- In other words, we have a hierarchy of servers in the same way that we have a hierarchy of names (see Figure 1.54).

STANDARD CLIENT-SERVER APPLICATIONS
Domain Name System (DNS)

Figure 1.54 *Hierarchy of name servers*

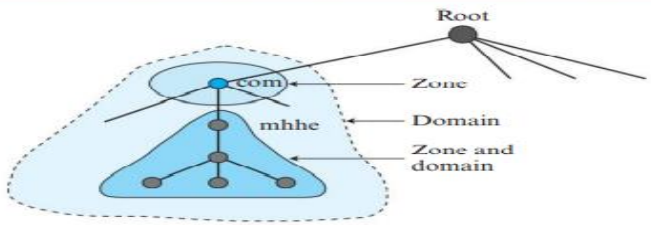


STANDARD CLIENT-SERVER APPLICATIONS
Domain Name System (DNS)

Zone

- Since the complete domain name hierarchy cannot be stored on a single server, it is divided among many servers.
- What a server is responsible for or has authority over is called a zone.
- We can define a zone as a contiguous part of the entire tree.
- The server makes a database called a zone file and keeps all the information for every node under that domain.

Figure 1.55 *Zone*



STANDARD CLIENT-SERVER APPLICATIONS
Domain Name System (DNS)

Root Server

- A root server is a server whose zone consists of the whole tree.
- A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers.

Primary and Secondary Servers

- DNS defines two types of servers: primary and secondary.
- A primary server is a server that stores a file about the zone for which it is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk.
- A secondary server is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk.

Root Server

Primary and Secondary Servers

- A primary server loads all information from the disk file; the secondary server loads all information from the primary server.

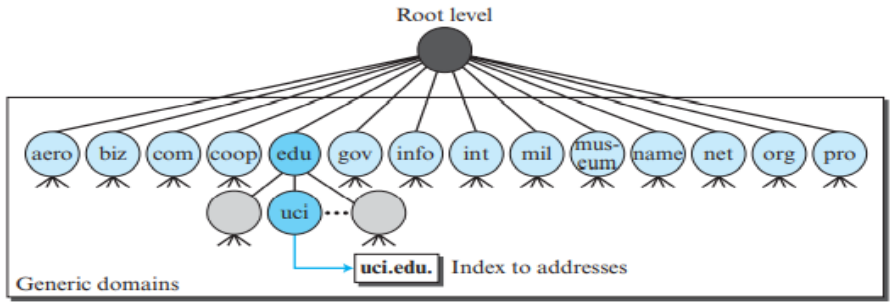
DNS in the Internet

- DNS is a protocol that can be used in different platforms.
- In the Internet, the domain name space (tree) was originally divided into three different sections: generic domains, country domains, and the inverse domain.

Generic Domains

- The generic domains define registered hosts according to their generic behavior.
- Each node in the tree defines a domain, which is an index to the domain name space database (see Figure 1.56).

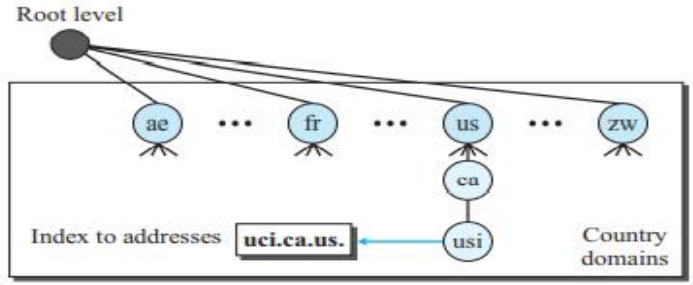
Figure 1.56 *Generic domains*



Country Domains

- The country domains section uses two-character country abbreviations (e.g., us for United States, in for India).
- Second labels can be organizational, or they can be more specific, national designations.

Figure 1.57 *Country domains*



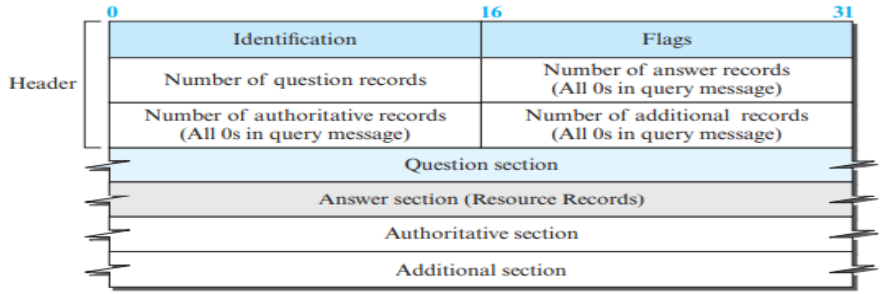
Resolution

- Mapping a name to an address is called name-address resolution.
- DNS is designed as a client-server application.
- A host that needs to map an address to a name or a name to an address calls a DNS client called a resolver.
- Recursive Resolution
- Iterative Resolution

DNS Messages

- To retrieve information about hosts, DNS uses two types of messages: query and response. Both types have the same format as shown in Figure 1.58.

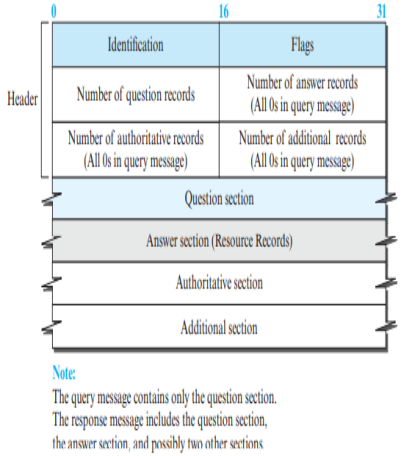
Figure 1.58 *DNS message*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

- The identification field is used by the client to match the response with the query.
- The flag field defines whether the message is a query or response. It also includes status of error.
- The next four fields in the header define the number of each record type in the message.
- The question section, which is included in the query and repeated in the response message, consists of one or more question records.

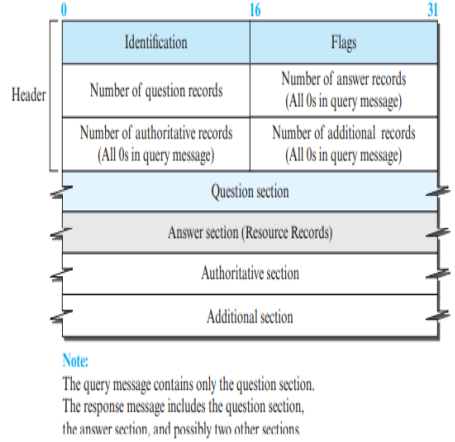
Figure 1.58 *DNS message*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

- The answer section consist of one or more resource records. It is present only in response messages.
- The authoritative section gives information (domain name) about one or more authoritative servers for the query.
- The additional information section provides additional information that may help the resolver.

Figure 1.58 *DNS message*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

STANDARD CLIENT-SERVER APPLICATIONS
Domain Name System (DNS)

Encapsulation

- DNS can use either UDP or TCP.
- In both cases the well-known port used by the server is port 53.
- UDP is used when the size of the response message is less than 512 bytes because most UDP packages have a 512-byte packet size limit.
- If the size of the response message is more than 512 bytes, a TCP connection is used. In that case, one of two scenarios can occur:

STANDARD CLIENT-SERVER APPLICATIONS
Domain Name System (DNS)

Registrars

How are new domains added to DNS?

- This is done through a registrar, a commercial entity accredited by ICANN.
- A registrar first verifies that the requested domain name is unique and then enters it into the DNS database. A fee is charged.

STANDARD CLIENT-SERVER APPLICATIONS
Domain Name System (DNS)

DDNS

- The DNS master file must be updated dynamically.
- The Dynamic Domain Name System (DDNS) was devised to respond to this need.
- In DDNS, when a binding between a name and an address is determined, the information is sent, usually by DHCP to a primary DNS server.
- The primary server updates the zone. The secondary servers are notified either actively or passively.
- In active notification, the primary server sends a message to the secondary servers about the change in the zone, whereas in passive notification, the secondary servers periodically check for any changes.
- To provide security and prevent unauthorized changes in the DNS records, DDNS can use an authentication mechanism.

STANDARD CLIENT-SERVER APPLICATIONS
Domain Name System (DNS)

- To protect DNS, IETF has devised a technology named DNS Security (DNSSEC) that provides message origin authentication and message integrity using a security service called digital signature

PEER-TO-PEER PARADIGM

P2P Networks

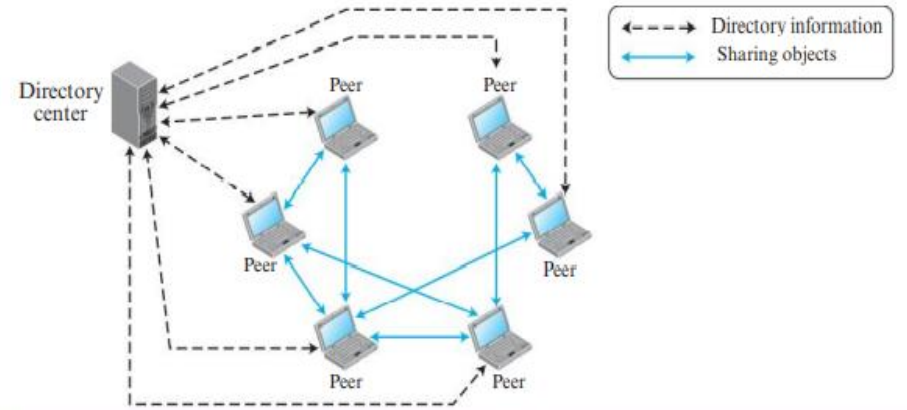
- We first need to divide the P2P networks into two categories: centralized and decentralized.

Centralized Networks

• In a centralized P2P network, the directory system – listing of the peers and what they offer – uses the client-server paradigm, but the storing and downloading of the files are done using the peer-to-peer paradigm (a hybrid P2P network).

PEER-TO-PEER PARADIGM

Figure 1.59 *Centralized network*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

PEER-TO-PEER PARADIGM

Centralized Networks

- A peer, looking for a particular file, sends a query to a central server.
- The server searches its directory and responds with the IP addresses of nodes that have a copy of the file.
- The peer contacts one of the nodes and downloads the file.
- The directory is constantly updated as nodes join or leave the peer.
- Centralized networks make the maintenance of the directory simple but have several drawbacks.
- Accessing the directory can generate huge traffic and slow down the system.
- The central servers are vulnerable to attack, and if all of them fail, the whole system goes down.

PEER-TO-PEER PARADIGM

P2P Networks

- We first need to divide the P2P networks into two categories: centralized and decentralized.

Decentralized Network

•A decentralized P2P network does not depend on a centralized directory system.

•In this model, peers arrange themselves into an overlay network, which is a logical network made on top of the physical network.

•Depending on how the nodes in the overlay network are linked, a decentralized P2P network is classified as either unstructured or structured.

PEER-TO-PEER PARADIGM

Unstructured Networks

- In an unstructured P2P network, the nodes are linked randomly.
- The Gnutella network is an example of a peer-to-peer network that is decentralized but unstructured.

Structured Networks

- A structured network uses a predefined set of rules to link nodes so that a query can be effectively and efficiently resolved.
- The most common technique used for this purpose is the Distributed Hash Table (DHT).
- DHT is used in many applications including Distributed Data Structure (DDS), Content Distributed Systems (CDS), Domain Name System (DNS), and P2P file sharing.
- One popular P2P file sharing protocol that uses the DHT is BitTorrent.

PEER-TO-PEER PARADIGM

Distributed Hash Table (DHT)

- A Distributed Hash Table (DHT) distributes data (or references to data) among a set of nodes according to some predefined rules.
- Each peer in a DHT-based network becomes responsible for a range of data items.
- To avoid the flooding overhead, DHT-based networks allow each peer to have a partial knowledge about the whole network.
- This knowledge can be used to route the queries about the data items to the responsible nodes using effective and scalable procedures.
- There are several protocols that implement DHT systems: Chord, Pastry, and Kademlia.

PEER-TO-PEER PARADIGM

Distributed Hash Table (DHT)

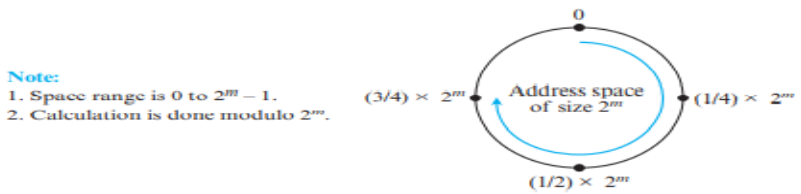
Address Space

•In a DHT-based network, each data item and the peer is mapped to a point in a large address of size 2^m.

•The address space is designed using modular arithmetic, which means that the points in the address space as distributed evenly on a circle with 2^m points (0 to $2^m - 1$) using clockwise direction as shown in Figure 1.60.

•Most of the DHT implementations use m = 160.

Figure 1.60 *Address space*



[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

PEER-TO-PEER PARADIGM

Distributed Hash Table (DHT)

Hashing Peer Identifier

- The first step in creating the DHT system is to place all peers on the address space ring. This is normally done by using a hash function that hashes the peer identifier, normally its IP address, to an m-bit integer, called a node ID.

  node ID = hash (Peer IP address)

Hashing Object Identifier

- The name of the object (for example, a file) to be shared is also hashed to an m-bit integer in the same address space. The result in DHT is called a key.

  key = hash (Object name)

PEER-TO-PEER PARADIGM

Chord

- Chord was published by Stoica et al in 2001.

Identifier Space

- Data items and nodes in Chord are m-bit identifiers that create an identifier space of size 2^m points distributed in a circle in the clockwise direction.
- We refer to the identifier of a data item as k (for key) and the identifier of a peer as N (for node).

Finger Table

- A node in the Chord algorithm should be able to resolve a query: given a key, the node should be able to find the node identifier responsible for that key or forward the query to another node.
- Forwarding means that each node needs to have a routing table. Chord requires that each node knows about m successor nodes and one predecessor node. Each node creates a routing table, called a finger table by Chord.

PEER-TO-PEER PARADIGM

Chord

Table 1.9 *Finger table*

i	*Target Key*	*Successor of Target Key*	*Information about Successor*
1	N + 1	Successor of N + 1	IP address and port of successor
2	N + 2	Successor of N + 2	IP address and port of successor
⋮	⋮	⋮	⋮
m	$N + 2^{m-1}$	Successor of $N + 2^{m-1}$	IP address and port of successor

[Behrouz A Forouzan, Firouz Mosharraf, "Computer Networks: A top down Approach", McGraw Hill Education]

PEER-TO-PEER PARADIGM

Chord

Interface

- Chord needs a set of operations referred to as the Chord interface.

Lookup

- Probably the mostly used operation in Chord is the lookup.
- Chord is designed to let peers share available services between themselves.
- To find the object to be shared, a peer needs to know the node that is responsible for that object: the peer that stores a reference to that object.
- In Chord, a peer that is the successor of a set of keys in the ring is the responsible peer for those keys.
- Finding the responsible node is actually finding the successor of a key.

PEER-TO-PEER PARADIGM

A Popular P2P Network: BitTorrent

- BitTorrent is a P2P protocol, designed by Bram Cohen, for sharing a large file among a set of peers.
- However, the term sharing in this context is different from other filesharing protocols.
- Instead of one peer allowing another peer to download the whole file, a group of peers takes part in the process to give all peers in the group a copy of the file.
- File sharing is done in a collaborating process called a torrent.

PEER-TO-PEER PARADIGM

A Popular P2P Network: BitTorrent

- Each peer participating in a torrent downloads chunks of the large file from another peer that has it and uploads chunks of that file to other peers that do not have it, a kind of tit-for-tat, a trading game played by kids.
- The set of all peers that takes part in a torrent is referred to as a swarm.
- A peer in a swarm that has the complete content file is called a seed; a peer that has only part of the file and wants to download the rest is called a leech.
- A swarm is a combination of seeds and leeches.
- BitTorrent has gone through several versions and implementations, the original one which uses a central node called a tracker.