## CS3591 COMPUTER NETWORKS

**UNIT I INTRODUCTION AND APPLICATION LAYER 10**

Data Communication - Networks – Network Types – Protocol Layering – TCP/IP Protocol suite – OSI Model – Introduction to Sockets - Application Layer protocols: HTTP – FTP – Email protocols (SMTP - POP3 - IMAP - MIME) – DNS – SNMP

**DATA COMMUNICATION**

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

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

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

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

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

# Networks

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

## Components of data communication (Networking)

A data communications system has five components.

1. Message. The message is the information (data) to be communicated
2. Sender. The sender is the device that sends the data message. It can be a computer, telephone handset, camera, and so on.
3. Receiver. The receiver is the device that receives the message. It can be a computer, telephone handset, television, and so on.
4. Transmission medium. The transmission medium is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves
5. Protocol. A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating, just as a person speaking French cannot be understood by a person who speaks only Japanese.

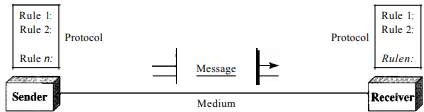


Figure 1.1 *Five components of data communication*

## Application of Computer Networks

* Internet is one of the main applications of the computer networks. Most widely used Internet applications are electronic mail, streaming audio and video, Word Wide Web, MP3 etc.
* Major applications areas of computer networks are:
  + - 1. Business applications 2. Home applications 3. Mobile

## 1. Business applications

* Now a days computers are being used in almost all business processes. For example, use of computers to monitor production, inventories, to make payments. Resource sharing is the important purpose of using computer networks. Resources like programs, equipments and data are required to share amongst various users.

## Database resource

* The database is required to access for decision making by various departments. The database is maintained by dedicated server and users (clients) can access the data. One server can provide services to many clients. The client and server in a network is shown in Fig. 1.2. This arrangement is called as client – server model.

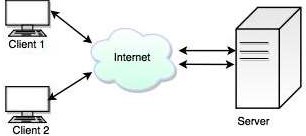


Figure 1.2 *Client-server model*

* Client requests for a service and server acknowledges the request. The server performs the requested work and sends back the result. The process of request and reply for a client-server model is shown in Fig. 1.3.

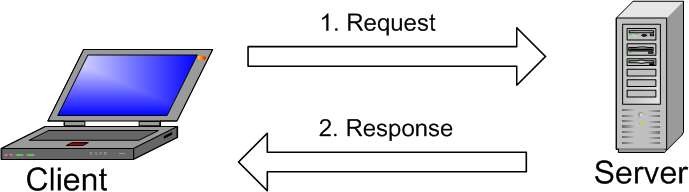


Figure 1.3 *Request and reply in client-server model*

1. **Communication medium:** Computer network is a powerful medium for communication. E-mail is very popularly used amongst company employees. Video- conferencing is other form of computer assisted communications.
2. **Electronic commerce:** Many companies doing business electronically with suppliers and customers. Customers can place order electronically; this assures fast delivery and efficient services.

## Home Applications

* Now a days, use of computer in home is widespread. Popular uses of computers in home are as under.
  1. Internet access ii) Personal communication iii) Entertainment iv) Electronic commerce
* Surfing on Internet may be for fun, to acquire information and for playing games. Information on every field is available on internet such as arts, science, technology, business, government, health, games, travels, music, cooking etc. Many newspapers are available on-line and selected articles can be downloaded.
* Various magazines, scientific journals, e-books are available on line. Many professional organizations also provide their journals, conference proceedings on-line.
* E-mail, instant messaging, chating, internet telephony, video phone provides personal communication by using Internet and [WWW.](http://WWW/)
* Entertainment applications include video on demand, interactive films and games, virtual reality games, line televisions where audience, participating in quiz show, choosing among contestants etc.
* E-commerce facilitates home shopping, catalogs of company products, on line technical support. E-commerce also popularly employed for bill payments, banking, investments, on line auctions. Commonly used forms of e-commerce and their typical applications are shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **B-Commerce** |  | **Applications** |
| 1. | Business – to – consumer | → | On line ordering |
| 2. | Business – to – Business | → | Supply chain management (Suppliers to  manufacturers) |
| 3. | Government – to – Consumer | → | Different government forms on Internet.  e.g., Income tax, Application forms. |
| 4. | Consumer – to – Consumer | → | Auctioning of second hand products. |
| 5. | Peer – to – peer | → | File sharing. |

## Mobile Computers

* Many professionals use desktop computers at office and want to be connected to the office network while travelling and at home also. This is possible by wireless networks, hence use of Lap-top, notebook computers and personal digital assistants (PDAs) is increased. With the help of wireless networks one can access internet, read and send e-mail. Wireless networks are used in:
  1. Taxis, delivery vehicles and other mobile vehicles for keeping contacts with their office.
  2. Geographical Information Systems (GIS).
  3. Military applications
  4. Airports
  5. Banking
  6. Weather reporting

## Requirements

* For designing the computer network, it is necessary to identify the requirements and constraints. Depending upon the user or organizations, requirements are changed.
* Following are the three parameters which affects the design:
  + - 1. Application programmer : Specify the list of services
      2. Network designer: List the properties of a cost-effective design
      3. Network Provider: List the properties of a system that is easy to administer and manage.

## Challenges for Building Networks

1. The Scalability and Extensible Network: The rapid growth of the Internet and network technologies has increased audio, video, image and graphic data applications, which consume large volumes of network bandwidth.
2. Security in Computer Networks: Part of the security challenge comes from increased use of a divergent platforms, end-systems and network protocols.
3. The biggest challenge of the implementation is in verifying network components to ensure they are capable of protecting security, privacy and reliability.
4. Difficulty in protecting data crossing over different network components.
5. Reliability: Reliability means availability and correctness. Systems for providing services must be always available and correct and commit to fulfill every request from the legitimate users.
6. Protocol: Protocol indicates criteria and mechanisms used in the network’s communication.

## Network Criteria

A network must be able to meet a certain number of criteria. The most important of these are

* Performance
* Reliability
* Security

### Performance

Performance can be measured based on transmit time and response time. Transmit time is the amount of time required for a message to travel from one device to another. Response time is the elapsed time between an inquiry and a response.

### Throughput

* Throughput is an actual measurement of how fast data can be transmitted whereas bandwidth is a potential measurement of link.
* Throughput is usually less than bandwidth.

### Latency

* Latency is also termed as delay. Latency is time required for a message to completely arrive at the destination from source. If has four components propagation time, transmission time, queuing time and processing delay.

### Reliability

The network reliability is measured by the frequency of failure and time taken to recover from a failure

### Security

Network security include protecting data from unauthorized access, protecting data from damage and development, and implementing policies and procedures for recovery from modifications and data losses.

## Physical Structures

### Type of Connection

A network is two or more devices connected through links. There are two possible types of connections:

* Point-to-point
* Multipoint.

### Point-to-Point

A point-to-point connection provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices. Most point-to- point connections use an actual length of wire or cable to connect the two ends.

### Multipoint

A multipoint (also called multidrop) connection is one in which more than two specific devices share a single link. In a multipoint environment, the capacity of the channel is shared, either spatially or temporally.

If several devices can use the link simultaneously, it is a *spatially shared* connection.

If users must take turns, it is a *timeshared* connection.

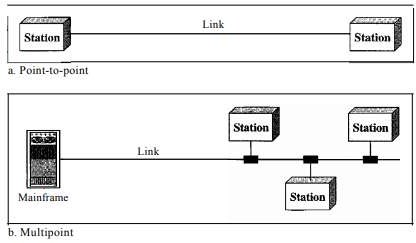


Figure 1.4 *Types of connections: point-to-point and multipoint*

* + - 1. *Physical Topology*

The term *physical topology* refers to the way in which a network

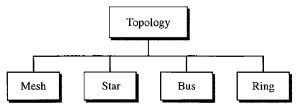


Figure 1.5 *Categories of topology*

**Mesh:** In a mesh topology, every device has a dedicated point-to-point link to every other device. The term *dedicated* means that the link carries traffic only between the two devices it connects. To find the number of physical links in a fully connected mesh network with *n* nodes, we first consider that each node must be connected to every other node. Node 1 must be connected to *n* - 1 nodes, node 2 must be connected to *n* – 1 nodes, and finally node *n* must be connected to *n* - 1 nodes. We need *n(n* - 1) physical links.

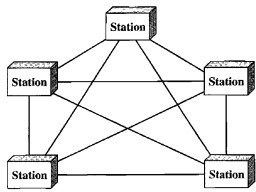
However, if each physical link allows communication in both directions (duplex mode), we can divide the number of links by 2. In other words, we can say that in a mesh topology, we need *n(n* -1) /2 duplex-mode links.

Figure 1.6 *A fully connected mesh topology (five devices)*

## Advantages

* + - * 1. The use of dedicated links guarantees that each connection can carry its own data load.
        2. A mesh topology is robust. If one link becomes unusable, it does not incapacitate the entire system.
        3. Third, there is the advantage of privacy or security. When every message travels along a dedicated line.
        4. Point-to-point links make fault identification and fault isolation easy..

## Disadvantages

1. Disadvantage of a mesh are related to the amount of cabling because every device must be connected to every other device, installation and reconnection are difficult.
2. Second, the sheer bulk of the wiring can be greater than the available space (in walls, ceilings, or floors) can accommodate.
3. The hardware required to connect each link (I/O ports and cable) can be prohibitively expensive.

**Star Topology**

In a star topology, each device has a dedicated point-to-point link only to a central controller, usually called a hub. The devices are not directly linked to one another. Unlike a mesh topology, a star topology does not allow direct traffic between devices. The controller acts as an exchange: If one device wants to send data to another, it sends the data to the controller, which then relays the data to the other connected device.

## Advantages

1. A star topology is less expensive than a mesh topology. In a star, each device needs only one link and one I/O port to connect it to any number of others. This factor also makes it easy to install and reconfigure.
2. Other advantages include robustness. If one link fails, only that link is affected. All other links remain active. This factor also lends itself to easy fault identification and fault isolation. As long as the hub is working, it can be used to monitor link problems and bypass defective links.

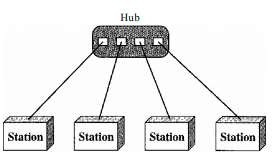


Figure 1.7 *A star topology connecting four stations*

## Disadvantage

The star topology is the dependency of the whole topology on one single point, the hub. If the hub goes down, the whole system is dead. Although a star requires far less cable than a mesh, each node must be linked to a central hub. For this reason, often more cabling is required in a star than in some other topologies (such as ring or bus).

## Bus Topology

A **bus topology,** is multipoint. One long cable acts as a **backbone** to link all the devices in a network. Bus topology was the one of the first topologies used in the design of early local area networks.

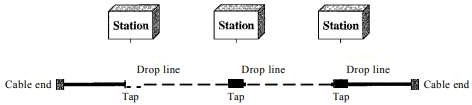


Figure 1.8 A bus topology connecting three stations

Nodes are connected to the bus cable by drop lines and taps. A drop line is a connection running between the device and the main cable. A tap is a connector that either splices into the main cable or punctures the sheathing of a cable to create a contact with the metallic core.

## Advantages

A bus topology is easy to installation. Only the backbone cable stretches through the entire facility.

## Disadvantages

It is difficult reconnection and fault isolation.

Adding new devices may therefore require modification or replacement of the backbone.

## Ring Topology

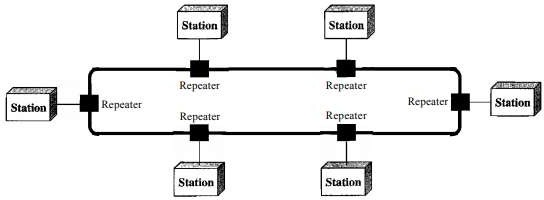
In a ring topology, each device has a dedicated point-to-point connection with only the two devices on either side of it. A signal is passed along the ring in one direction, from device to device, until it reaches its destination. Each device in the ring incorporates a repeater. When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along

Figure 1.9 *A ring topology connecting six stations*

## Advantages

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

requires changing only two connections. The only constraints are media and traffic considerations (maximum ring length and number of devices).

1. In addition, **fault isolation is simplified**. Generally in a ring, a signal is circulating at all times. If one device does not receive a signal within a specified period, it can issue an alarm. The alarm alerts the network operator to the problem and its location.

## Disadvantage

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

**Comparison between Bus and Ring Topology**

|  |  |  |
| --- | --- | --- |
| **Sl.No.** | **Bus topology** | **Ring topology** |
| 1. | Bus requires proper termination. Cable  cannot be left unterminated. | Termination is not required. |
| 2. | Bus is a passive network topology. | Ring is an active network topology. |
| 3. | There is loss in data integrity as the bus length increases. | Transmission errors are minimized because transmitted signal is  regenerated at each node. |
| 4. | It uses point to multipoint communication  links. | It uses point-to-point communication  links. |
| 5. | Recommended when large number of devices are to be attached. | Recommended when moderate number of devices are to be  attached. |

# Network Types: or Categories of Networks

## Local Area Networks (LAN)

Local area networks, generally called LANs, are privately-owned networks within a single building or campus of up to a few kilometer’s in size. They are widely used to connect personal computers and workstations in company offices and factories to share resources (e.g., printers) and exchange information. LANs are distinguished from other kinds of networks by three characteristics:

* Their size,
* Their transmission technology, and
* Their topology.

## Attributes of LAN

* The LAN transmits data amongst user stations
* The LAN transmission capacity is more than 1 Mbps.
* The LAN channel is typically privately owned by the organization using the facility
* The Geographical coverage of LANs is limited to area less than 5 square Kilometers

## Metropolitan Area Network (MAN)

A metropolitan area network, or MAN, covers a city. The best-known example of a MAN is the cable television network available in many cities. This system grew from earlier community antenna systems used in areas with poor over-the-air television reception. In these early systems, a large antenna was placed on top of a nearby hill and signal was then piped to the subscribers' houses. At first, these were locally-designed, ad hoc systems. Then companies began jumping into the business, getting contracts from city governments to wire up an entire city.

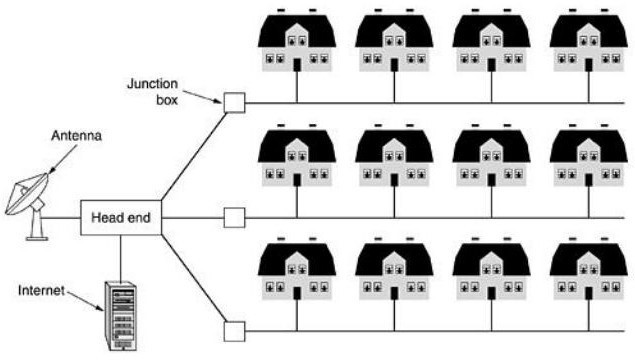


Figure 1.10 *Metropolitan area network based on cable TV.*

## Wide Area Network (WAN)

* A WAN provides long distance transmission of data and voice.
* A Network that covers a larger area such as a city, state, country or the world is called

## wide area network.

* The WAN contains host and collection of machines. User program is installed on the host and machines. All the hosts are connected by each other through communication subnet. Subnet carries messages from host to host.
* Fig. 1.11 shows the component of WAN.
* Subnet consists of transmission lines and switching elements. The transmission line is used for data transfer between two machines. Switching elements are used for connecting two transmission lines.
* Switching elements are specialized computers. It selects the proper outgoing line for

incoming data and forward the data on that line.

* The switching elements are basically computers and they are called packet switching nodes, intermediate systems and data switching exchanges. These switching elements are also called routers.

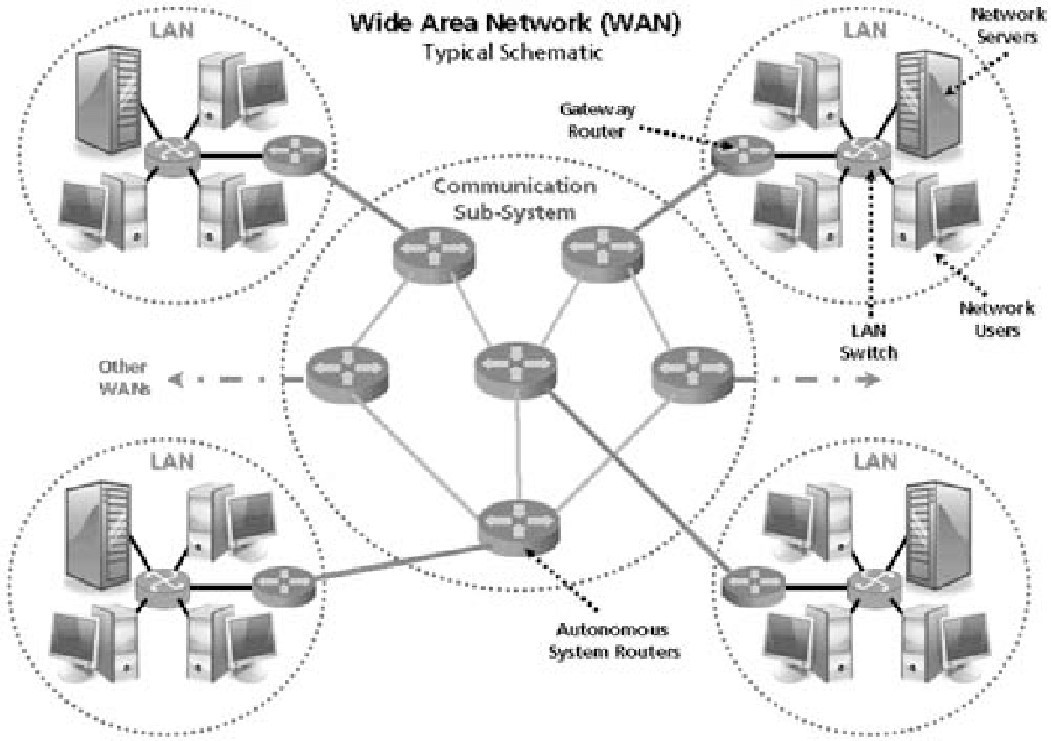


Figure 1.11 *Wide area network*

* Each host is connected to a LAN on which a router is present. Sometimes the host can be directly connected to the router. The interconnection of routers forms the subnet.
* In the WAN, when the packet is sent from one router to another via one or more intermediate routers, the packet is received at each intermediate router in its entirety. This packet is stored in that router until the required output line is free. The subnet which uses this principle is called point-to-point, store and forward, or packet switched subnet.
* Almost all the WANs use store and forward subnets.
* If the packets are small and of same size, they are also called cells.
* In the point-to-point subnet, the router interconnection topology becomes important. WANs can also use satellite or ground radio system. The routers have antenna, through which they can send or receive data, they can listen from satellite.
* WAN uses hierarchical addressing because they facilitate routing. Addressing is required to identify which network input is to be connected to which network output.

Comparison between LAN, WAN and MAN

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **LAN** | **WAN** | **MAN** |
| Area covered | Covers small area  i.e. within the building. | Covers large geographical area. | Covers larger than LAN & smaller than WAN. |
| Error rates | Lowest | Highest | Moderate |
| Transmission  speed | High speed. | Low speed | Moderate speed. |
| Equipment  cost | Uses inexpensive  equipment | Uses most expensive  equipment. | Uses moderately  expensive equipment. |

* + 1. **Comparison between LAN and WAN**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **LAN** | **WAN** |
| 1. | It covers small area. | WAN covers large geographical area. |
| 2. | LAN operates on the principal of  broadcasting. | WAN operates on the principal of point  to point. |
| 3. | Used for time critical application. | Not used for time critical application. |
| 4. | Transmission speed is high. | Transmission speed is low. |
| 5. | Easy to design and maintain. | Design and maintenance is not easy. |
| 6. | LAN is broadcasting in nature. | WAN is point-to-point in nature. |
| 7. | Transmission medium is co-axial  or UTP cable. | Transmission or communication  medium is PSTN or satellite link. |
| 8. | LAN does not suffer from  propagation delay. | WAN suffer from propagation delay. |

## Wireless Networks

* A wireless LAN or WLAN is a wireless local area network that uses radio waves as its carrier. The last link with the users is wireless, to give a network connection to all users in a building or campus. The backbone network usually uses cables.
* Wireless LANs operate in almost the same way as wired LANs, using the same networking protocols and supporting the most of the same applications.

How are WLANs Different?

1. They use specialized physical and data link protocols.
2. They integrate into existing networks through access points which provide a bridging function.
3. They let you stay connected as you roam from one coverage area to another.
4. They have unique security considerations.
5. They have specific interoperability requirements.
6. They require different hardware.
7. They offer performance that differs from wired LANs.

## Example 1. Consider a bus LAN with a number of equally spaced stations with a data rate of 9 Mbps and a bus length of 1 km. What is the mean time to send a frame of 500 bits to another station, measured from the beginning of transmission to the end of reception? Assume a propagation speed of 150 m/s. If two stations begin to monitor and transmit at the same time, how long does it need to wait before an interference is noticed?

**Solution:** We assume that the distance between two stations is 500 m

Mean time to send = Propagation time + Transmission time

= 500 m / 150 msec. + 500 bits / 9000000 bps.

= 3.33 msec. + 55.55 msec. = 58.88 msec.

If the two stations begin the transmission at exactly the same time the signal will interface after exactly 250 m.

Tinterface = (250 m + 250 m) / 150 m / msec = 3.33 msec …Ans.

# Layering and Protocols

* A computer network must provide general, cost effective, fair and robust connectivity among a large number of computers. Designing a network to meet these requirements is no small task.
* To deal with this complexity, network designers have developed general blue prints – usually called network architectures. It guides the design and implementation of networks.

## Layered Architecture

* Computer network is designed around the concept of layered protocols or functions. For exchange of data between computers, terminals or other data processing devices, there is data path between two computers, either directly or via a communication network.
* Protocols are the rules that govern network communication. Fig. 1.12 shows the five layer network.
* Layer n on one node carries on a conversation with layer n on other node.
* The entities comprising the corresponding layers on different machine and called peers.
* The actual data flow is from upper layer to its below layer and then from physical medium to destination layer.



Figure 1.12 *Layers, protocols and interfaces*

* Between each pair of adjacent layers is called interface. The interface defined which primitive operations and services the lower layer offers to the upper one.
* A set of layers and protocols is called network architecture.

# OSI Architecture

* The ISO was one of the first organizations to formally define a common way to connect computers in 1947. Their architecture, called the Open System Interconnection (OSI).
* The International organization for standardization developed the **Open System Interconnection (OSI)** reference model. OSI model is the most widely used model for networking.
* OSI model is a seven layer standard.
* OSI model provides following services.

1. Provides peer-to-peer logical services with layer physical implementation.
2. Provides standards for communication between system.
3. Defines point of interconnection for the exchange of information between system.
4. Each layer should perform a well defined function.
5. Narrows the options in order to increase the ability to communicate without expansion conversions and translations between products.

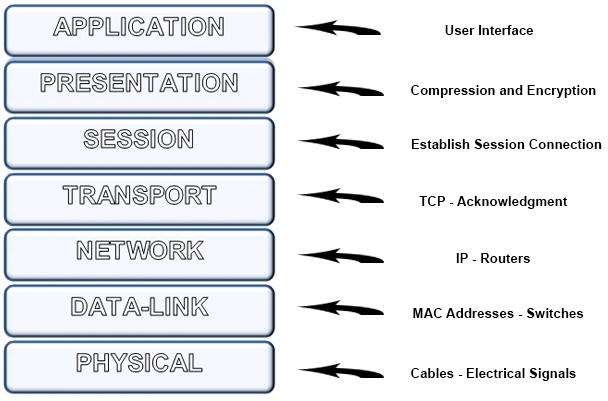


Figure 1.13 *OSI Layer model*

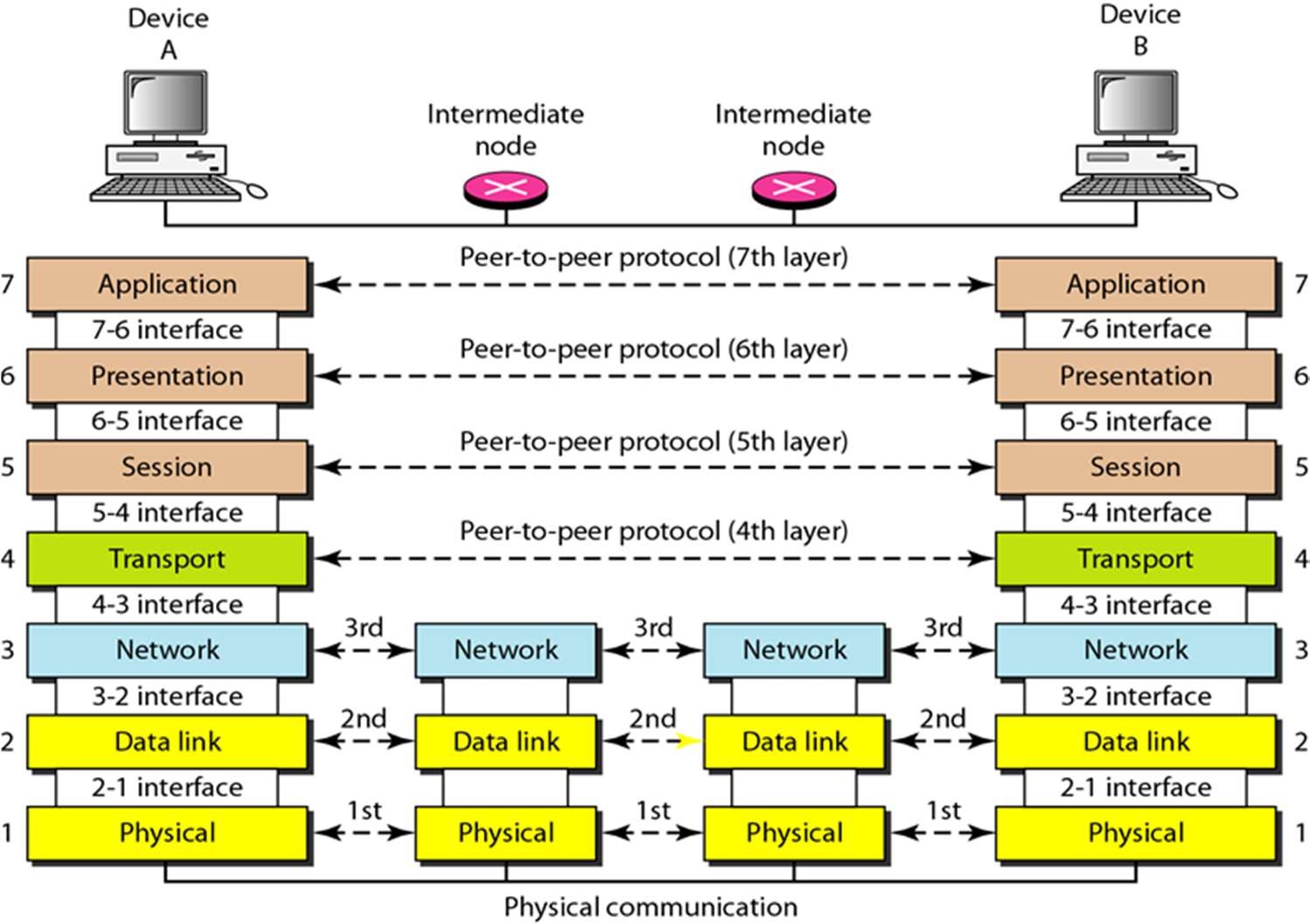


Figure 1.14 *OSI Layer model with Interface*

## Physical Layer

Physical Layer is the lowest layer of the OSI model. The physical layer coordinates the functions required to transmit a bit stream over a communication channel. It deals with the mechanical and electrical specifications of the interface and transmission medium. It also deals with procedures and functions required for transmission.

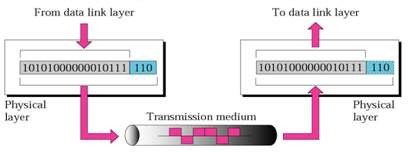


Figure 1.15 *Physical layer*

## Function of Physical layer

1. **Physical characteristics of interfaces and medium**. The physical layer defines the characteristics of the interface between the devices and the transmission medium.
2. **Representation of bits**. The physical layer data consists of a stream of bits (sequence of Os or 1s) with no interpretation. To be transmitted, bits must be encoded into signals--electrical or optical.
3. **Data rate.** The physical layer define the transmission rate-The number of bits sent each second
4. **Synchronization of bits.** The transmission rate and receiving rate must be same. This is done by synchronizing clock at sender and receiver.
5. **Line configuration.** The physical layer is concerned with the connection of devices to the media. In a point-to-point configuration, two devices are connected through a dedicated link. In a multipoint configuration, a link is shared among several devices.
6. **Physical topology**. The physical topology defines how devices are connected to make a network. Devices can be connected by using a mesh topology, star topology, a ring topology, a bus topology, or a hybrid topology (this is a combination of two or more topologies).
7. **Transmission mode**. The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex. In simplex mode, only one device can send; the other can only receive. The simplex mode is a one-way communication. In the half-duplex  two devices can send and receive, but not at the same time. In a full-duplex (or simply duplex) mode, two devices can send and receive at the same time.

***Data Link Layer***

## The data link layer is responsible for transmitting frames from one node to the

**next**.

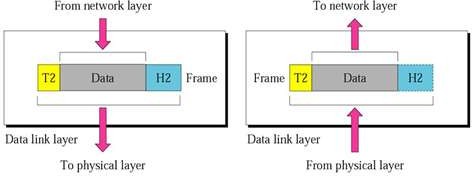


Figure 1.16 *Data link layer*

Other responsibilities of the data link layer include the following:

1. **Framing.** The data link layer divides the stream of bits received from the network layer into manageable data units called frames.
2. **Physical addressing.** If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender or receiver of the frame.
3. **Flow control.** When the rate of the data transmitted and rate of data reception by receiver is not same, same data may be lost.
4. **Error control.** The data link layer incorporate reliability to the physical layer. By adding mechanisms to detect and retransmit damaged or lost frames
5. **Access control**. When two or more devices are connected to the same link, data link layer determine which device has control over the link.

### Network Layer

**The network layer is responsible for the delivery of packet from source to destination possibly across multiple networks (links).** The network layer ensures that each packet gets from its point of origin to its final destination.

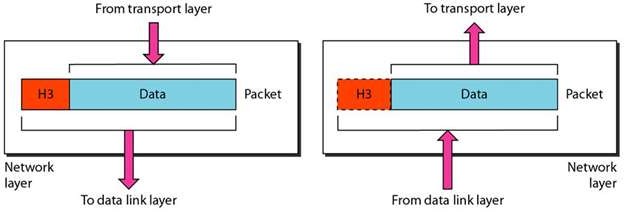


Figure 1.17 *Network layer*

Other responsibilities of the network layer include the following:

1. **Logical addressing.** The physical addressing implemented by the data link layer handles the addressing problem locally. If a packet passes the network boundary, we need another addressing system to help distinguish the source and destination systems. The network layer adds a header to the packet of upper layer includes the logical addresses of the sender and receiver.
2. **Routing.** Network layer, route or switch the packets to its final destination in an internetwork.

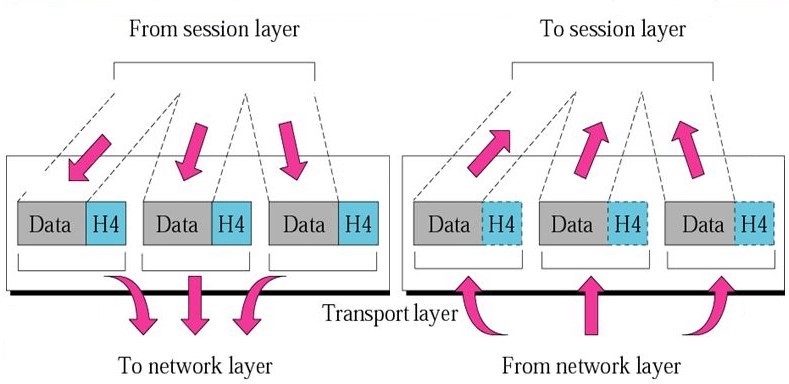
### Transport Layer

**The transport layer is responsible for delivery of message from one process to another process**. The network layer does the source-to-destination delivery of individual packets considering it as independent packet; it does not recognize any relationship between those packets. The transport layer ensures that the whole message arrives intact and in order with error control and flow control at the source-to-destination level.

Figure 2.10 shows the relationship of the transport layer to the network and session

layers.

Figure 1.18 *Transport layer*



Other responsibilities of the transport layer include the following:

1. **Service-point addressing.** Computers perform several operations simultaneously. For this reason, source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on one computer to a specific process (running program) on the other. The transport layer header must therefore include a type of address called a *service-point address* (or port address).
2. **Segmentation and reassembly.** A message is divided into segments, each segment containing a sequence number which enable the transport layer to reassemble at the destination.
3. **Connection control.** The transport layer performs connectionless or connection oriented service with the destination machine.
4. **Flow control.** Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.
5. **Error control.** Like the data link layer, the transport layer is responsible for error control. However, error control at this layer is performed end to end rather than across a single link.

### Session Layer

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

Specific responsibilities of the session layer include the following:

1. **Dialog control:** The communication between two processes to take place in either half duplex (one way at a time) or full-duplex (two ways at a time) mode. The session layer manages control for this communication.
2. **Synchronization.** The session layer adds checkpoints, or synchronization points, to a stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to insert.

Figure 1.19 illustrates the relationship of the session layer to the transport and presentation layers.

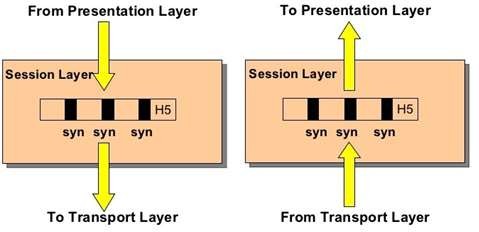


Figure 1.19 *Session layer*

### Presentation Layer

**The presentation layer deals with the syntax and semantics of the information exchanged between two systems. The presentation layer is responsible for translation, compression, and encryption**. Figure 1.20 shows the relationship between the presentation layer and the application and session layers.

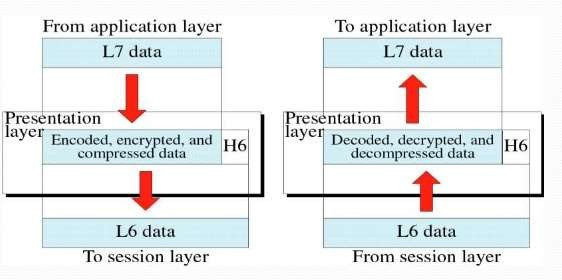


Figure 1.20 *Presentation layer*

Specific responsibilities of the presentation layer include the following:

* 1. **Translation**. The different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods.
  2. **Encryption.** Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.
  3. **Compression**. Data compression reduces the number of bits contained in the information. Data compression becomes particularly important in the transmission of multimedia such as text, audio, and video.

### Application Layer

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

Figure 1.21 shows the relationship of the application layer to the user and the presentation layer.

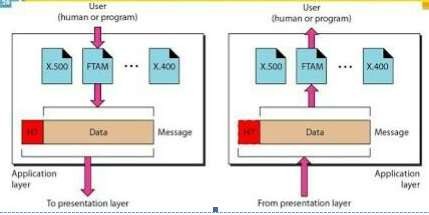


Figure 1.21 *Application layer*

Specific services provided by the application layer include the following:

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

### Summary of Layers

Figure 1.25 shows a summary of duties for each layer.

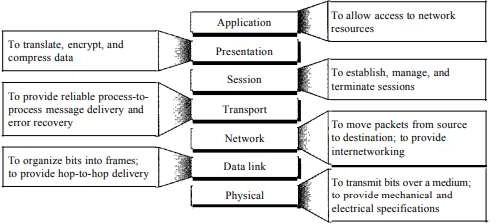


Figure1.25 *Summary of layers*

# TCP/IP Protocol Suite (Internet Architecture)

The TCPIIP protocol suite was developed prior to the OSI model. Therefore, the layers in the TCP/IP protocol suite do not exactly match those in the OSI model. The original TCP/IP protocol suite was defined as having four layers:

* Host-to-network Layer
* Internet Layer
* Transport Layer
* Application Layer.

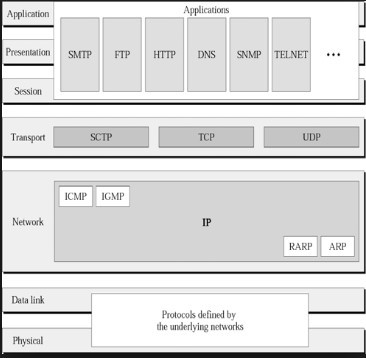


Figure 1.26 *TCP/IP and OSI model*

## Physical and Data Link Layers

At the physical and data link layers, *TCP/IP* does not define any specific protocol. It supports all the standard and proprietary protocols. A network in a *TCP/IP* internetwork can be a local-area network or a wide-area network.

## Network Layer

At the network layer (or, more accurately, the internetwork layer), *TCP/IP* supports the Internetworking Protocol. IP, in turn, uses four supporting protocols: ARP, RARP, ICMP, and IGMP.

### Internetworking Protocol (IP)

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

### Address Resolution Protocol

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

### Reverse Address Resolution Protocol

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

### Internet Control Message Protocol

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

### Internet Group Message Protocol

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

### Transport Layer

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

### User Datagram Protocol

The User Datagram Protocol (UDP) is the simpler of the two standard TCP/IP transport protocols. It is a process-to-process protocol that adds only port addresses, checksum error control, and length information to the data from the upper layer ***Transmission Control Protocol (TCP)***

It provides full transport-layer services to applications. TCP is a reliable stream transport protocol. And it is connection-oriented: A connection must be established between both ends of a transmission before either can transmit data.

### Stream Control Transmission Protocol

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

## Application Layer

The *application layer* in TCP/IP is equivalent to the combined session, presentation, and application layers in the OSI model.

## Comparison of the OSI and TCP/IP Protocol Suite

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **OSI Model** | | **TCP/IP Model** | | | |
| 1. | 7 layers | | 4 layers | | | |
| 2. | Model was first defined  implementation takes place. | before | Model defined  implemented. | after | protocols | were |

|  |  |  |
| --- | --- | --- |
| 3. | OSI model based on three concept  i.e. service, interface and protocol. | TCP/IP model did not originally clearly distinguish between service, interface and  protocol. |
| 4. | OSI model gives guarantee of  reliable delivery of packet. | Transport layer does not always  guarantee the reliable delivery of packet. |
| 5. | OSI does not support internet  working | TCP/IP support. |
| 6. | Strict layering | Lossely layered. |
| 7. | Support connections less and connection-oriented communication  in the network layer. | Support only connection-oriented communication in the transport layer. |

* + 1. **Addressing**
* An Internet employing TCP / IP protocols uses four levels of addresses:
  + - 1. Physical (Link) addresses 2. Logical (IP) addresses

3. Port addresses 4. Specific addresses

* Each address type is related to a specific layer in TCP / IP architecture. Fig.1.27 shows the relationship of layers and addresses in TCP / IP.

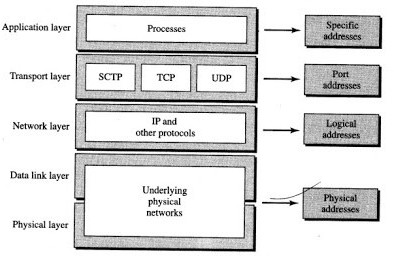
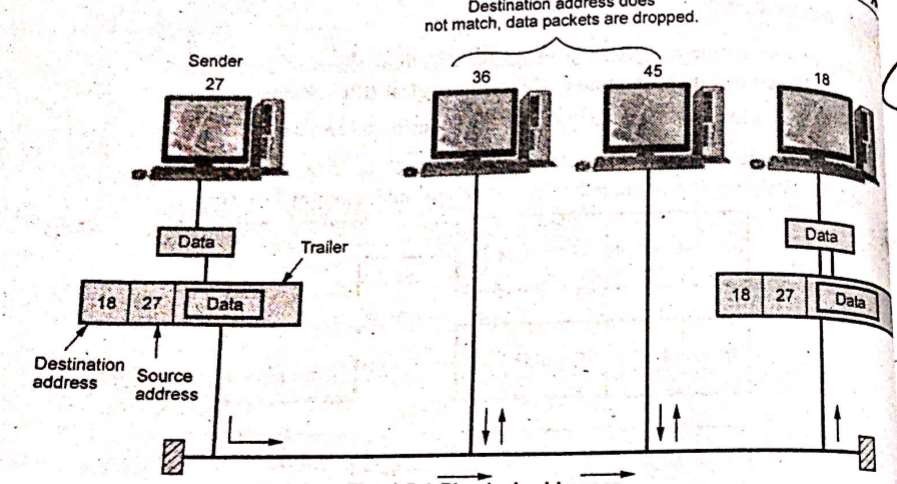


Figure 1.27 *TCP / IP layers and associated addresses*

## Physical Addresses

* The physical address is the lowest level address and is also refereed as link address. They physical address of a node is defined by its LAN or WAN. The physical address is included in the frame by the data link layer.
* The size and format of physical addresses vary depending on the network. It has authority over the network. At data link layer the frame contains physical (link) addresses in the header.

## Logical Addresses

* Logical addresses are independent of underlying physical networks. Since different networks can have different address formats hence a universal address system is required which can identify each host uniquely irrespective of underlying physical networks. Logical addresses are necessary for universal communications. It is 32-bit address which uniquely defines host connected to Internet.

## Fig. 1.28. Physical addresses

* The physical addresses changes from hop to hop, but the logical address usual remains the same.

## Port Addresses

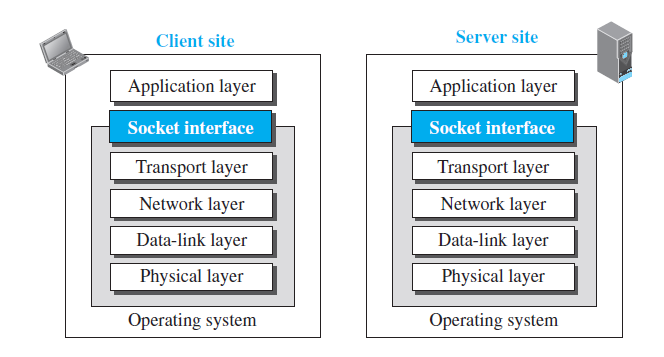
* The IP address and physical address are necessary for data to travel from source to destination. But a communication process involves TELNET and FTP which requires addresses. In TCP/IP architecture, the label assigned to a process is called port address. In TCP/IP the port address is of 16-bit.

## Specific Addresses

* Specific addresses are designed by users for some applications. For example [evilaas@in.com](mailto:evilaas@in.com) and the Universal Resource Locator (URL), [www.vtubooks.com.](http://www.vtubooks.com/) The first example defines the recipient of e-mail and second example is used to find a document on the world wide web.
* The specific addresses gets changed to corresponding port and logical addresses by the station or host who sends it.

**INTRODUCTION TO SOCKETS**

Socket interface started in the early 1980s at UC Berkeley as part of a UNIX environment. The socket interface is a set of instructions that provide communication between the application layer and the operating system, as shown in Figure 25.4. It is a set of instructions that can be used by a process to communicate with another process. The idea of sockets allows us to use the set of all instructions already designed in a programming language for other sources and sinks. For example, in most computer languages, like C, C++, or Java, we have several instructions that can read and write data to other sources and sinks such as a keyboard (a source), a monitor (a sink), or a file (source and sink). We can use the same instructions to read from or write to sockets. In other words, we are adding only new sources and sinks to the programming language

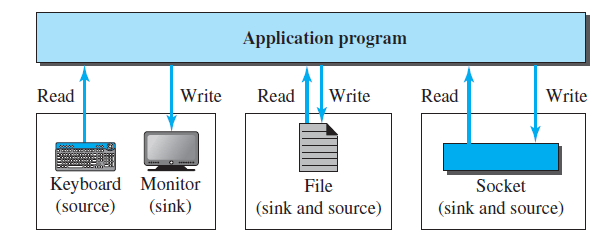


*Position of the socket interface*

without changing the way we send data or receive data. Figure 25.5 shows the idea and compares the sockets

with other sources and sinks.

**Figure 25.5** *Sockets used the same way as other sources and sinks*



*Sockets used the same way as other sources and sinks*

***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 an object that is created and used by the application program.
* We can say that, as far as the application layer is concerned, communication between a client process

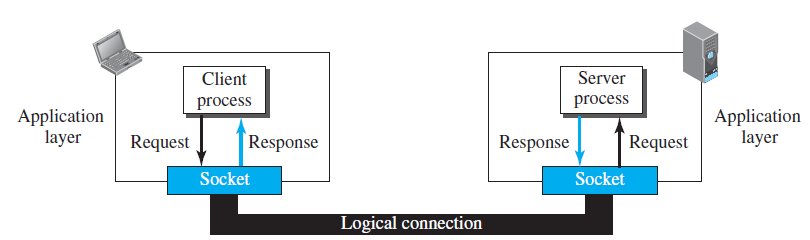
and a server process is communication between two sockets, created at two ends

The client thinks that the socket is theentity that receives the request and gives the response; the server thinks

that the socket is the one that has a request and needs the response. If we create two sockets, one at each end, and

define the source and destination addresses correctly, we can use the available instructions to send and receive

data. The rest is the responsibility of the operating system and the embedded TCP/IP protocol.



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

***Socket Addresses***

* The interaction between a client and a server is two-way communication.
* In a two-way communication, we need a pair of addresses: local (sender) and remote (receiver).
* The local address in one direction is the remote address in the other direction and vice versa.
* 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.

* However, we need to define a socket address in terms of identifiers used in the TCP/IP protocol

suite.

* A socket address should first define the computer on which a client or a server is running.
* A computer in the Internet is uniquely defined by its IP address, a 32-bit integer in the current

Internet version.

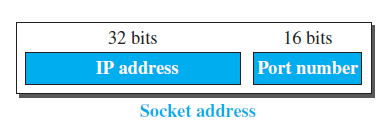
* However, several client or server processes may be running at the same time on a computer,

which means that we need another identifier to define the specific client or server involved

in the communication.

* An application program can be defined by a port number, a 16-bit integer. This means that

a socket address should be a combination of an IP address and a port number as shown



*A socket address*

Since a socket defines the end-point of the communication, we can say that a socket is identified by a pair

of socket addresses, a local and a remote.

***Finding Socket Addresses***

How can a client or a server find a pair of socket addresses for communication? The situation is different for each

site.

***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, however, 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.

* For example, the assigned port number for a Hypertext Transfer Protocol (HTTP) is the integer

80, which cannot be used by any other process.

* If the server process is not standard, the designer of the server process can choose a port number,

in the range defined by the Internet authority, and assign it to the process. When a server starts

running, it knows the local socket address.

***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 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, however, 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, however, needs to be assigned from a set of integers defined by the

Internet authority and called the ephemeral (temporary) port numbers.

* The operating system, however, needs to guarantee that the new port number is not used by any

other running client process.

* The operating system needs to remember the port number to be able to redirect the response

received from the server process to the client process that sent the request.

***Remote Socket Address*** Finding the remote (server) socket address for a client, however, 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.

Sometimes, 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 andwe want to test them.

For example, at the end of this chapter we write some simple client and server programs and

we test them using this approach. In this situation, the programmer can provide these two pieces of information

when he runs the client program.

❑ Although 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 web page, send an e-mail to a friend, copy a file from a remote site, and so on. In these situations, the server has a name, an identifier that uniquely defines the server process. 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)*. We will discuss DNS in Chapter 26, but it is enough to know that it acts as a directory in the Internet. Compare the situation with the telephone directory. We want to call someone whose name we know but whose telephone number can be obtained from the telephone directory. The telephone directory maps the name to the telephone number; DNS maps the server name to the IP address of the computer running that server.

# WWW

The World Wide Web (WWW) is a repository of information linked together from points all over the world. The WWW has a unique combination of flexibility, portability, and user-friendly features that distinguish it from other services provided by the Internet

## Architecture

The WWW today is a distributed client/server service, in which a client using a browser can access a service using a server. However, 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 can contain a link to other pages in the same site or at other sites. The request, among other information, includes the address of the site and the Web page, called the URL.

The server at site A finds the document and sends it to the client. When the user views the document, she finds some references to other documents, including a Web page at site B. The reference has the URL for the new site. The user is also interested in seeing this document. The client sends another request to the new site, and the new page is retrieved.

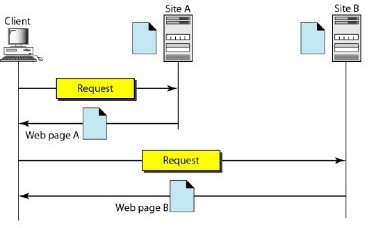


Figure 5.1 Architecture of WWW

## Client (Browser)

A variety of vendors offer commercial browsers that interpret and display a Web document, and all use nearly the same architecture. Each browser usually consists of three parts: **a controller, client protocol, and interpreters.**

## Server

The Web page is stored at the server. Each time a client request arrives, the corresponding document is sent to the client.

## Uniform Resource Locator

A client that wants to access a Web page needs the address. To facilitate the access of documents distributed throughout the world, HTTP uses locators. The uniform resource locator (URL) is a standard for specifying any kind of information on the Internet. The URL defines four things: **protocol, host computer, port, and path.**

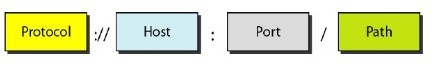


Figure 5.3 URL

The protocol is the client/server program used to retrieve the document. The most commonly used Protocol today is HTTP.

The host is the computer on which the information is located. The URL can optionally contain the port number of the server. Path is the pathname of the file where the information is located. **Cookies**

* A cookies is small file. Frequently access browsers information are stored in a cookies directory.
* A cookie may contain upto five fields.

a) Domain b) Path c) Content

d) Expires e) Secure

1. **Domain** : It tells where the cookies came from.
2. **Path** : The path is a path in the server's directory structure that identifies which parts of the server's file tree may use the cookie.
3. **Content :**It takes the form name value.
4. **Expires :**The expires field specifies when the cookies expires.
5. **Secure :**This field can be set to indicate that the browser may only return the cookie to secure server.

## Examples of cookies

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Domin** | **Path** | **Content** | **Expires** | **Secure** |
| toms-casino.com | / | customer ID=497793521 | 15-10-02, 17:00 | Yes |
| joes-store.com | / | cart=1-00501;1-07031;2-13721 | 11-10-02,15:20 | No |
| Sneaky.com | / | User ID=3456789 | 30-12-06, 11:00 | Yes |

* + 1. **WEB DOCUMENTS**

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

## 1. Static Documents

Static documents are fixed-content documents that are created and stored in a server. The client can get only a copy of the document. In other words, the contents of the file are determined when the file is created, not when it is used.

## HTML

Hypertext Markup Language (HTML) is a language for creating Web pages.

* HTML documents are in plain text format that contain embedded HTML tags. Documents can be created in any text editor. There are also many other tools, including editors, designed specifically to assist in creating HTML documents. To HTML document, the user needs a browser.
* A document will be ready by both graphical and character based web browser. The three basic tagging pairs used to create the highest level of structure in an HTML documents are as follows :

<HTML> HTML documents </HTML>

<HEAD> Header information of documents </HEAD>

<BODY> Body of the HTML document </BODY> The general structure of the HTML is

<HTML>

<HEAD>

<TITLE>

Title here

</TITLE>

</HEAD>

<BODY>

Body element and content

</BODY>

</HTML>

A simple HTML document is given below.

<HTML>

<HEAD>

<TITLE> Communication Networks </TITLE>

</HEAD>

<BODY>

<H/> Information about the communication networks </H/>

<P> Information about the communication networks is available

<A HREF :[http://www.technicalpublicationspune.com](http://www.technicalpublicationspune.com/)></A></P>

</BODY>

</HTML>

* Structural elements in the document are identified by Start andEnd tags. For example the

<TITLE> and </TITLE> tags are used to specify the title of the document.

* The <H/> and </H/> tags' are used to define the first level heading. Headings are generated by an <Hn> tags, where n is a digit in the range 1 to 6. <H/> is the most important heading and <H6> is the less important. Typically the lower numbered heading will be displayed in a larger and heavier font.
* The browser may also choose to use different colors for each level of heading. Typically

<H1> headings are large and bold face with at least one blank line above and below.

* In contrast <H2> headings are in a smaller font, and with less space above and below. The <BR>, <P> and <HR> tags all indicate a boundary between sections of text.
* The precise format can be determined by the style sheet associated with the page. The

<BR> tag just forces a line break. <P> starts a paragraph, which might for example, insert a blank line and possibly some indentation. <HR> (horizontal-rule) tag forces the browser to generate a horizontal rule or line, across the display. It breaks pages into logical sections and is useful when creating forms. There is no equivalent vertical rule.

## Advantages and Disadvantages of HTML

1. **Advantages of HTML**
2. Applications are quickly developed
3. Web applications are easy to maintain and update.

## Disadvantages

1. **Locking**: HTML is not a compiled data format. .
2. **Security :** Information is easily accessible and travels unimpeded between hosts and desktops.

## Dynamic Web Documents

Server Side dynamic web page generation using the various scripting languages

### Scripting Technologies for Dynamic Documents

A few technologies have been involved in creating dynamic documents using scripts. Among the most common are Hypertext Preprocessor (pHP), which uses the Perl language; Java Server Pages (JSP), which uses the Java language for scripting; Active Server Pages (ASP), a Microsoft

product which uses Visual Basic language for scripting; and ColdFusion, which embeds SQL database queries in the HTML document.

### 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.

### Java Applets

One way to create an active document is to use Java applets.

### JavaScript

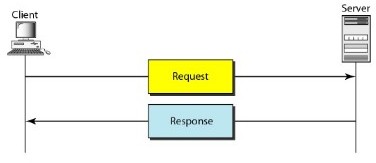
The idea of scripts in dynamic documents can also be used for active documents. If the active part of the document is small, it can be written in a scripting language; then it can be interpreted and run by the client at the same time.

# HTTP

The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World.

## HTTP Transaction

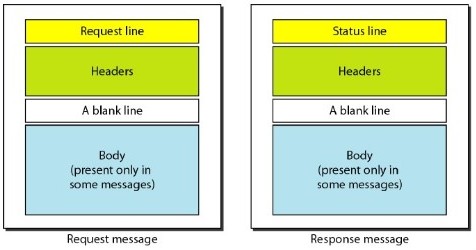
Although HTTP uses the services of TCP, HTTP itself is a stateless protocol. The client initializes the transaction by sending a request message. The server replies by sending a response.



HTTP transaction

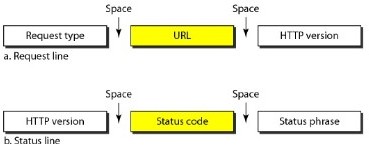
## Messages

The formats of the request and response messages are similar. A request message consists of a request line, a header, and sometimes a body. A response message consists of a status line, a header, and sometimes a body.



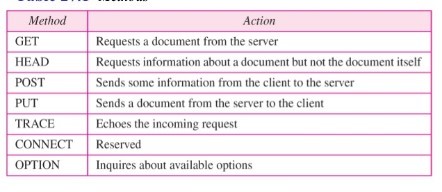
Request and response message

**Request and Status Lines** The first line in a request message is called a request line; the first line in the response message is called the status line.



Request and status lines

**Request type:** This field is used in the request message. In version1.1of HTTP, several request types are defined. The request types is categorized into methods as follow



**Version:** The most current version of HTTP is 1.1.

**Status code**: This field is used in the response message. It consists of three digits.

**Status phrase**: This field is used in the response message. It explains the status code in text form.

**Header:** The header exchanges additional information between the client and the server.

**Difference between Persistent and Non-persistent**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Persistent HTTP** | **Non-persistent HTTP** |
| 1. | Persistent version is 1.1 | Non-persistent HTTP version is 1.0 |
| 2. | It uses one RTT. | It uses two RTT. |
| 3. | TCP connection is not closed. | TCP connection is closed after every  request response |
| 4. | Client make multiple request over the  same TCP connection. | Client make multiple request over the  multiple TCP connection. |
| 5. | It is default mode. | It is not default mode. |
| 6. | Request methods are GET, HEAD,  POST, PUT, DELETE, TRACE and OPTIONS. | Request methods used are GET, POST and HEAD. |

# ELECTRONIC MAIL

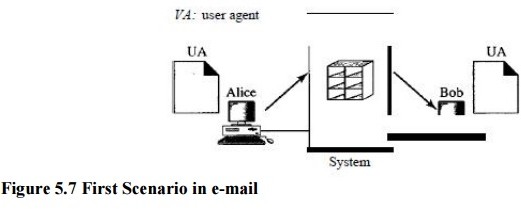
One of the most popular Internet services is electronic mail (e-mail).

## Architecture

To explain the architecture of e-mail, we give four scenarios. We begin with the simplest situation and add complexity as we proceed. The fourth scenario is the most common in the exchange of email.

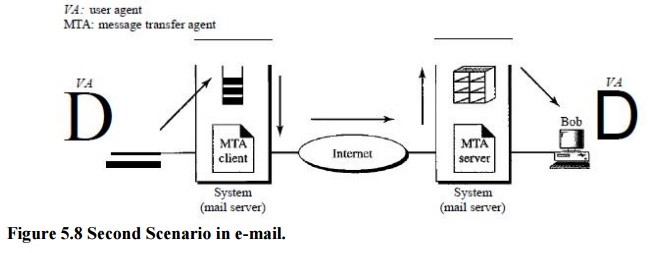
## First Scenario

In the first scenario, the sender and the receiver of the e-mail are users (or application programs) on the same system; they are directly connected to a shared system. When Alice, a user, needs to send a message to Bob, another user, Alice runs a user agent (VA) program to prepare the message and store it in Bob's mailbox. The message has the sender and recipient mailbox addresses (names of files). Bob can retrieve and read the contents of his mailbox at his convenience, using a user agent.



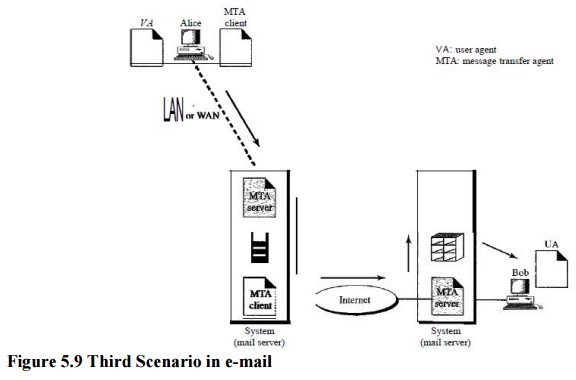
## Second Scenario

In the second scenario, the sender and the receiver of the e-mail are users (or application programs) on two different systems. The message needs to be sent over the Internet. Here we need user agents (VAs) and message transfer agents (MTAs).



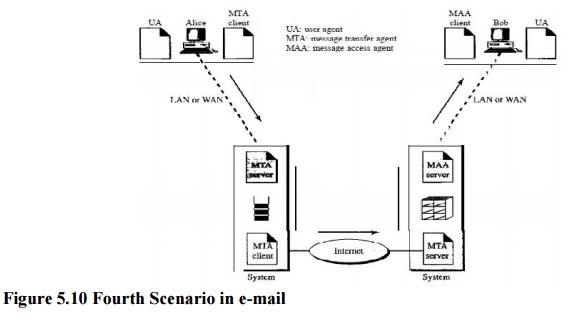
## Third Scenario

In the third scenario, Bob, as in the second scenario, is directly connected to his system. Alice, however, is separated from her system. Either Alice is connected to the system via a point-to- point WAN, such as a dial-up modem, a DSL, or a cable modem; or she is connected to a LAN in an organization that uses one mail server for handling e-mails-all users need to send their messages to this mail server.



## Fourth Scenario

In the fourth and most common scenario, Bob is also connected to his mail server by a WAN or a LAN. After the message has arrived at Bob's mail server, Bob needs to retrieve it. Here, we need another set of client/server agents, which we call message access agents (MAAs). Bob uses an MAA client to retrieve his messages. The client sends a request to the MAA server, which is running all the time, and requests the transfer of the messages.

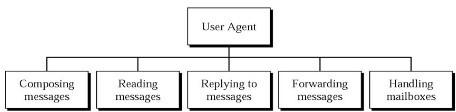


## User Agent

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.

## Services Provided by a User Agent

A user agent is a software package (program) that composes reads, replies to, and forwards messages. It also handles mailboxes.



Service of user agent

**Composing Messages** A user agent helps the user compose the e-mail message to be sent out. Most user agents provide a template on the screen to be filled in by the user. Some even have a built-in editor that can do spell checking, grammar checking, and other tasks expected from a sophisticated word processor.

**Reading Messages** The second duty of the user agent is to read the incoming messages. When a user invokes a user agent, it first checks the mail in the incoming mailbox. Most user agents show a one-line summary of each received mail. Each e-mail contains the following fields.

1. A number field.
2. A flag field that shows the status of the mail such as new, already read but not replied to, or read and replied to.
3. The size of the message.
4. The sender.
5. The optional subject field.

**Replying to Messages** After reading a message, a user can use the user agent to reply to a message.

**Forwarding Messages** Replying is defined as sending a message to the sender a message to the sender or recipients of the copy. Forwarding is defined as sending the message to a third party.

## User Agent Types

There are two types of user agents: command-driven and GUI-based.

## Command-Driven

Command-driven user agents belong to the early days of electronic mail. They are still present as the underlying user agents in servers. A command-driven user agent normally accepts a one- character command from the keyboard to perform its task.

**GUI-Based Modem** 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.

## Sending Mail

To send mail, the user, through the UA, creates mail that looks very similar to postal mail. 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 (such as encoding type, as we see shortly)..

## Receiving Mail

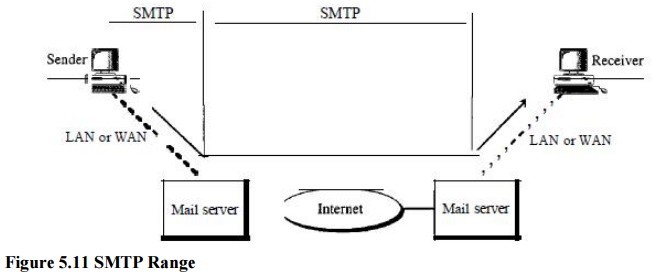
The user agent is triggered by the user (or a timer). If a user has mail, the VA informs the user with a notice. If the user is ready to read the mail..

## Message Transfer Agent: SMTP

The actual mail transfer is done through message transfer agents.

To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA.

The formal protocol that defines the MTA client and server in the Internet is called the 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.

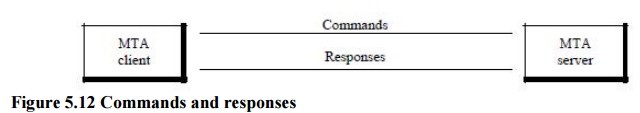
SMTP simply defines how commands and responses must be sent back and forth.

## Commands and Responses

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

**Commands:** Commands are sent from the client to the server. It consists of a keyword followed by zero or more arguments.

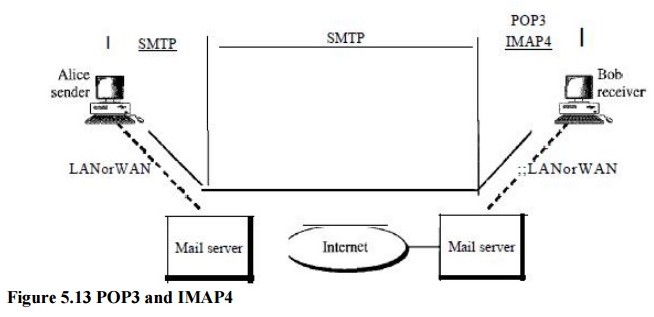
**Responses:** Responses are sent from the server to the client. A response is a three digit code that may be followed by additional textual information.



## Mail Transfer Phases

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

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



## Post Office Protocol (POP)

* Post Office Protocol 3 (POP3) is used to transfer e-mail messages from a mail server to mail client software.
* POP3 begins when the user agent opens a TCP connection to the mail server on port 110.
* After TCP connection established, POP3 progresses three phases :

i) Authorization ii) Transaction iii) Update

* In **authorization phase**, user agent sends a user name and a password to authenticate the user downloading the mail.
* In **transaction phase**, the user agent retrieves messages. In this phase, user agent can also mark messages for deletion, remove deletion marks.
* **In update phase**, it occurs after the client has issued the quit command, ending the POP3 session.
* POP3 has two modes : **Delete mode and the keep mode.**
* In the delete mode, mail is deleted from the mailbox after each retrieval.
* In the keep mode, the mail remains in the mailbox after retrieval.
* Fig. 5.4.7 shows downloading using POP3.

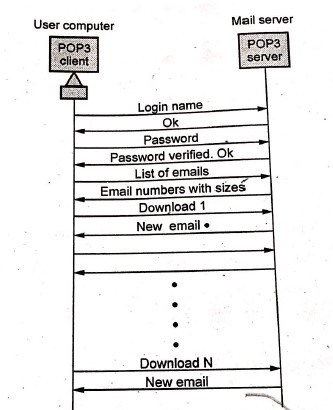


Fig. 5.4.7 *POP3*

## IMAP

* IMAP is the Internet Mail Access Protocol. IMAP4 is more power and more complex. IMAP is similar to SMTP.
* It was designed to help the user who uses multiple computers.
* An IMAP client connects to a server by using TCP.
* IMAP supports the following modes for accessing e-ma me

i) Offline mode ii) Online mode iii) Disconnected mode

**Offline mode:** A client periodically connects to the server to download e-mail messages. After downloading, messages are deleted from the server.

**Online mode** : Client process e-mail messages on the server. The e-mail messages are stored on the server itself but are processed by an application on the client’s end.

**Disconnected mode :** In this mode, both offline and online modes are supported.

## IMAP4 provides the following extra functions.

1. User can check the e-mail header prior to downloading.
2. User can partially download e-mail.
3. A user can create, delete or rename mailboxes on the mail server.
4. A user can create a hierarchy of mailboxes in a folder for e-mail storage.
5. User can search the contents of the e-mail for a specific string of characters.

* Fig. 5.4.8 shows IMAP state transition diagram.

1. **Not authenticated:** Client provides authentication information to the server.
2. **Authenticated:** Server verify the information and client is now allowed to perform operations on a mailbox.
3. **Selected:** Client is allowed to access of manipulate individual messages within the mailbox.
4. **Logout:** Client send logout command for closing IMAP session.

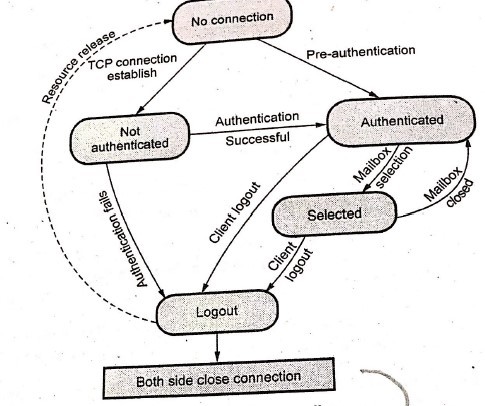


Fig.5.4.8 IMAP *state diagram*

# Domain Name System (DNS)

The client/server programs can be divided into two categories: those that can be directly used by the user, such as e-mail, and those that support other application programs. The Domain Name System (DNS) is a supporting program that is used by other programs such as e-mail.

## Components of DNS

* DNS includes following components

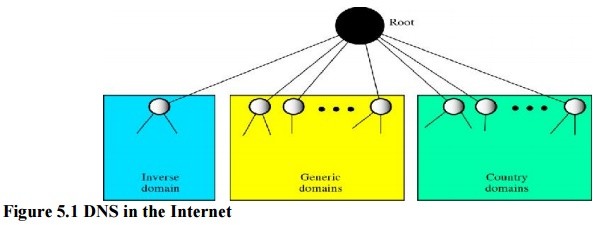
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. | Domain | 2. Domain name 3. Name server |  | | |
| 4. | Name resolver | 5. Name cache 6. Zone |
| 1) | For | example, google.com. Here | com | is | the |
|  | domain. |  |  |  |  |

1. google.com could be domain name.
2. In name server, software (program) that maps names to addresses.
3. Name resolver is a software that functions as a client interacting with a name server.
4. Name cache is the storage used by the name resolver to store reformation frequently used.
5. Zone is a contiguous part of a domain.

## 2. DNS in the Internet:

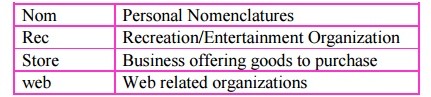
DNS is a protocol that can be used in different platforms. In the Internet, the domain name space is divided into three sections are

1. Generic domains
2. Country domains and
3. Inverse domain

Generic domain:

There are 14 generic domains, each specifying an organization type. The generic domain defines registered hosts according to their generic behaviour.

These labels describe the organization types as shown below



## Country domains:

Each country domain specifies a country. Such as in for india, jp for japan , uk for United kingdom and us for United State , etc

## Inverse domain:

The inverse domain finds a domain name for a given IP address. This is called address-to-name resolution. It is used to map an address to a name.

## 3. Types of Records:

There are two types of DNS records:

1. Question records
2. Resource records

## Question Records:

The question records are used in the question section of the query and response messages. It is used by the client to get information from a server.

## Resource Records:

Every domain whether it is a single host or a top level domain, can have a set of resource records associated with it. For a single host, the most common resource record is just its IP address, but many other kinds also exist. When a resolver gives a domain name to DNS, what it gets back are the resource records associated with that name. Thus, the primary function of DNS is to map domain names onto resource records. The server database consists of resource records. This record is used in the answer, authoritative and additional information sections of the response message.

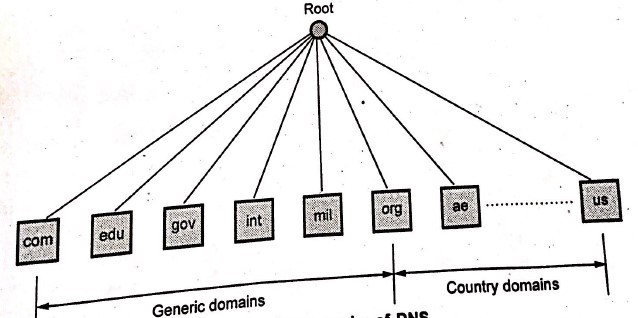
## Name Spaces

* Name spaces are of two types: Flat name spaces and Hierarchical names.

## Flat name spaces

* A name is assigned to an address.

## Hierarchical names

* Hierarchical namespaces provides a simple yet flexible naming structure.
* The namespace is partitioned at the top level. The top level domains are divided into three areas :
  1. Arpa is a special domain used for the address-to-name mappings.
  2. The 3 character domains are called the generic domains.
  3. The 2 character domains are based on the counter codes found in ISO 3166. These are called the country domains.
* Fig. 5.7.5shows the hierarchy of DNS.

## Hierarchy of Name Servers

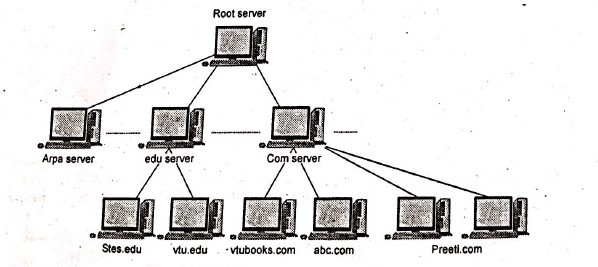


Fig.5.7.7 *Hierarchy of name server*

* To distribute the information among many computers, DNS servers are used.Creates many domains as there are first level nodes. Fig. 5.7.7 shows hierarchy of name servers.
* **Root server**: If zone consists of the full tree then that zone server is called root server. Root server do not maintain any information about domains.

DNS uses two types of servers:

1. Primary server 2. Secondary server

* **Primary server**: This server keeps a file about the zone for which it is responsible and have authority. It performs operation on zone file like create, update and maintaining.
* **Secondary server**: It loads all information from the primary server. Secondary server cannot perform any operation on zone file.

## Message Format

* Messages are sent between domain clients and domain servers with a specific format.
* DNS has two types of messages: Query and Response. Both types have the same format.
* The query message consists of the header and the question records, the response message consists of a header, question record, answer record, authoritative record and additional records.
* Fig. 5.7.11 shows the query and response messages.

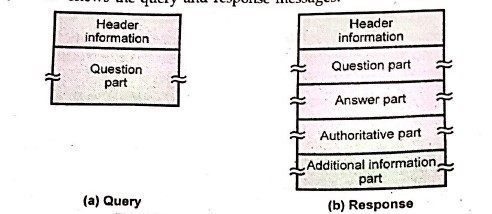


Fig. 5.7.11 *Query and response message*

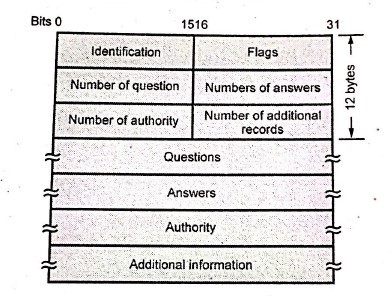
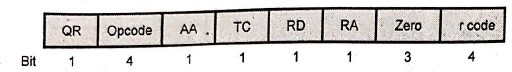
* Fig. 5.7.12 shows the header format of the DNS.

Fig. 5.7.12 *General format of DNS*

* **Identification:** It is 16 bits fields and unique value used by the client to match responses to queries.
* **Flags:** It is the collection of subfields that define the type of messages and type of the answers requested and so on.
* Number of question record contains the number of queries in the question section of the message.
* Number of answer record contains the number of answer records in the answer section of the response message.
* Number of authority record contains the number of authority records in the authoritative section of the response message.
* Number of additional records contains the number of additional records in the additional section of the response message. The message has a fixed 12-byte header followed by 4

variable length fields. The identification field is set by client and returned by the server. It lets the client, match responses to requests.

* Fig. 5.7.13 flag fields in DNS header.
* The flags field is divided into 8 parts.

|  |  |  |  |
| --- | --- | --- | --- |
| QR | = | 0 | For message is a query |
|  | = | 1 | It is response |
| Opcode | = | 0 | Standard query |
|  | = | 1 | Inverse query |
|  | = 2 | | Server status request |
| AA | = | | Authoritative answer |
| TC | = | | Truncated |
| RD | = | | Recursive query |
| RA | = | | Recursion available |
| r code | = | | Return code |

## Advantages of DNS

1. DNS has hierarchical structure and database.
2. DNS has small and manageable zones.
3. It is scalable.
4. DNS helps in eliminating host tables.
5. It is consistent on all hosts.
6. The Internet couldn't exist without it.
7. Easy to implement with minimal configuration changes in DNS server.

## NETWORK MANAGEMENTSYSTEM

Network Management System is a collection of tools for network monitoring and control. A network management system consists of hardware and software addition implemented among existing components.

A network management system can be divided into five broad categories: configuration management, fault management, performance management, security management, and accounting management, as shown in Figure 28.1.

### Simple Network Management Protocol(SNMP)

SNMP uses the concept of manager and agent. That is, a manager, usually a host, controls and monitors a set of agents, usually routers

### Managers and Agents

A management station, called a manager, is a host that runs the SNMP client program. A managed station, called an agent, is a router (or a host) that runs the SNMP server program. Management is achieved through simple interaction between a manager and an agent.

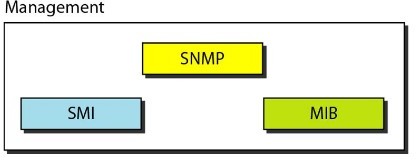
The agent keeps performance information in a database. The manager has access to the values in the database. For example, a router can store in appropriate variables the number of packets received and forwarded. The manager can fetch and compare the values of these two variables to see if the router is congested or not.

In other words, management with SNMP is based on three basic ideas:

1. A manager checks an agent by requesting information that reflects the behaviour of the agent.
2. A manager forces an agent to perform a task by resetting values in the agent database.
3. An agent contributes to the management process by warning the manager of an un usualsituation.

### Management Components

To do management tasks, SNMP uses two other protocols: Structure of Management Information (SMI) and Management Information Base (MIB). In other words, management on the Internet is done through the cooperation of the three protocols SNMP, SMI, and MIB, as shown in Figure 28.3.



### Role of SNMP

SNMP defines the format of packets exchanged between a manager and an agent. It reads and changes the status (values) of objects (variables) in SNMP packets.

### Role of SMI

SMI defines the general rules for naming objects, defining object types (including range and length), and showing how to encode objects and values.

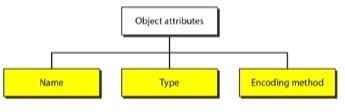
### Role of MIB

MIB creates a collection of named objects, their types, and their relationships to each other in an entity to be managed.

## Structure of Management Information(SMI)

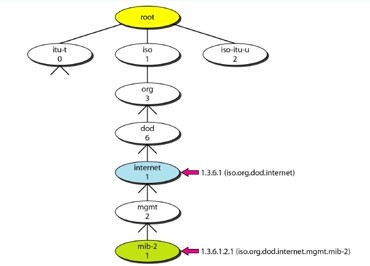
Its functions are

1. To name objects
2. To define the type of data that can be stored in an object
3. To show how to encode data for transmission over the network

SMI is a guideline for SNMP. It emphasizes three attributes to handle an object: name, data type, and encoding method (see Figure 28.5).

### Name

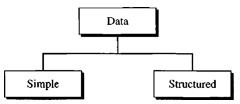
SMI requires that each managed object (such as a router, a variable in a router, a value) have a unique name. To name objects globally, SMI uses an object identifier, which is a hierarchical identifier based on a tree structure (see Figure28.6).



### Type

The second attribute of an object is the type of data stored in it. To define the data type, SMI uses fundamental Abstract Syntax Notation 1 (ASN.l) definitions and adds some new definitions. .

SMI has two broad categories of data type: *simple* and *structured.*



Simple Type The simple data types are atomic data types. Table28.1*Datatypes*

|  |  |  |
| --- | --- | --- |
| *Type* | *Size* | *Description* |
| INTEGER | 4 bytes | An integer with a value between \_231 and 231 - 1 |
| Integer32 | 4 bytes | Same as INTEGER |
| Unsigned32 | 4 bytes | Unsigned with a value between 0 and 232 - 1 |
| OCTET STRING | Variable | Byte string up to 65,535 bytes long |
| OBJECT IDENTIFIER | Variable | An object identifier |
| IPAddress | 4 bytes | An IP address made of four integers |
| Counter32 | 4 bytes | An integer whose value can be incremented from 0 to 232; when it reaches its maximum  value, it wraps back to O. |
| Counter64 | 8 bytes | 64-bit counter |
| Gauge32 | 4 bytes | Same as Counter32, but when it reaches its maximum value, it does not wrap; it remains  there until it is reset |
| TimeTicks | 4 bytes | A counting value that records time in s |
| BITS |  | A string of bits |
| Opaque | Variable | Uninterpreted string |

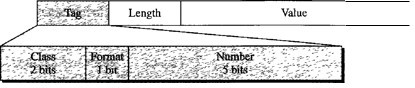
Structured Type: By combining simple and structured data types. SMI defines two structured data types: *sequence* and *sequence of.*

**Sequence**. A *sequence* data type is a combination of simple data types, not necessarily of the same type.

**Sequence of.** A *sequence of* data type is a combination of simple data types all of the same type.

### Encoding Method

SMI uses another standard, Basic Encoding Rules (BER), to encode data to be trans mitted over the network.



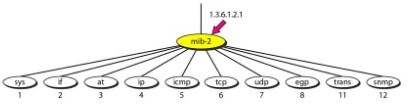
**Tag**. The tag is a 1 -byte field that defines the type of data. It is composed of three subfields:*class*(2*bits),format*(1bit),and*number*(5bits).Theclasssubfielddefines the scope of the data..

**Length.**The length field is 1 or more bytes. If it is 1 byte, the most significant bit must be O. The other 7 bits define the length of the data. If it is more than 1 byte, the most significant bit of the first byte must be 1.

**Value**. The value field codes the value of the data according to the rules defined in BER.

## Management Information Base (MIB)

The Management Information Base, version 2 (MIB2) is the second component used in network management. Each agent has its own MIB2, which is a collection of all the objects that the manager can manage. The objects in MIB2 are categorized under 10different groups: system, interface, address translation, ip, icmp, tcp, udp, egp, trans• mission, and snmp. These groups are under the mib-2 object in the object identifier tree (see Figure 28.15). Each group has defined variables and/or tables.



The following is a brief description of some of the objects:

**Sys:**Thisobject *(system)* defines general information about the node (system), such as the name, location, and lifetime.

**if** :This object *(interface)* defines information about all the interfaces of the node including interface number, physical address, and IPaddress.

**at :**Thisobject*(addresstranslation)*definestheinformationabouttheARPtable.

**ip:**Thisobject defines information related to IP, such as the routing table and the IPaddress.

**icmp** :This object defines information related to ICMP, such as the number of packets sent and received and total errors created.

**tcp**: This object defines general information related to TCP, such as the connection table, time- out value, number of ports, and number of packets sent and received.

**Udp :**This object defines general information related to UDP, such as the number of ports and number of packets sent and received.

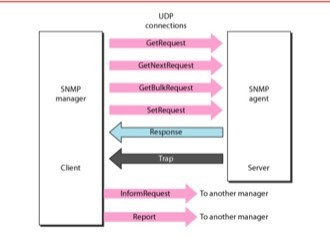
**Snmp** :This object defines general information related to SNMP itself.

## SNMP

1. A manager to retrieve the value of an object defined in an agent
2. A manager to store a value in an object defined in an agent
3. An agent to send an alarm message about an abnormal situation to the manager

### PDUs

SNMPv3 defines eight types of packets (or PDUs): GetRequest, GetNextRequest, GetBulkRequest,SetRequest,Response,Trap,InformRequest,andReport(seeFigure28.20).



**GetReques**tTheGetRequest PDU is sent from the manager (client) to the agent (server) to retrieve the value of a variable or a set of variables.

**GetNextRequest**TheGetNextRequestPDU is sent from the manager to the agent to retrievethevalueofavariable.

**GetBulkRequest**The GetBulkRequest POD is sent from the manager to the agent to retrieve a amount of data. It can be used instead of multiple GetRequest and GetNextRequestPODs.

**SetRequest**The SetRequest PDD is sent from the manager to the agent to set (store) a value in a variable.

**Response** The Response PDD is sent from an agent to a manager in response to GetRequest or GetNextRequest. It contains the value(s) of the variable(s) requested by the manager.

**Trap**The Trap (also called SNMPv2 Trap to distinguish it from SNMPv1 Trap) POD is sent from the agent to the manager to report an event. For example, if the agent is rebooted, it informs themanager and reports the time of rebooting.

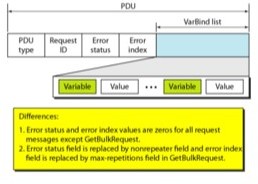
**InformRequest** The Inform Request POD is sent from one manager to another remote manager to get the value of some variables from agents under the control of the remote manager. The remote manager responds with a Response POD.

**Report** The Report POD is designed to report some types of errors between managers.

It is not yet in use.

### Format

The format for the eight SNMP PODs is shown in Figure 28.21. The GetBulkRequest POD differs from the others in two areas, as shown in the figure.



The fields are listed below:

**PDU type.** This field defines the type of the POD

**Request ID**.This field is a sequence number used by the manager in a Request POD and repeated by the agent in a response. It is used to match a request to a response.

**Error status.** This is an integer that is used only in Response PDUs to show the types of errors reported by the agent. Its value is 0 in Request PDUs.

**Nonrepeaters.**Thisfieldused only in GetBulkRequestand replaces the error status field, which is empty in RequestPDUs.

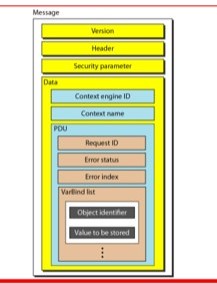
**Error index.** The error index is an offset that tells the manager which variable Caused the error.

**Max repetition :**This field is also used only in GetBulkRequest and replaces the error index field, which is empty in Request PDUs.

**VarBindlist.** This is a set of variables with the corresponding values the manager wants to retrieve or set.

## Messages

SNMP does not send only a PDU, it embeds the PDU in a message. A message in SNMPv3 is made of four elements: version, header, security parameters, and data (which include the encoded PDU), as shown in Figure 28.22.



## Strength of SNMP

* 1. It is simple to implement.
  2. Agents are widely implemented.
  3. Agent level overhead isminimal.
  4. It is robust and extensible.
  5. Polling approach is good for LAN based managed object.
  6. It offers the best direct manager agent interface.
  7. SNMP meet a critical need.

## Weakness of SNMP

1. It is too simple and does not scale well.
2. There is no object oriented data view.
3. It has no standard control definition.
4. It has many implementation specific (private MIB) extensions.
5. It has high communication overhead due to polling.
   1. **SSH**

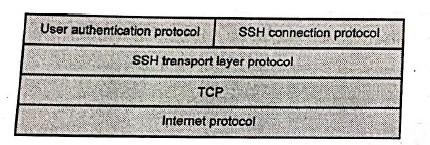
* SSH is a protocol for secure remote login and other secure network services over an insecure network.
* Secure Shell (SSH) is a protocol for secure network communications designed to be relatively simple and inexpensive to implement.
* Secure shell provides strong authentication and encrypted data communications between two computers connecting over an open network such as the internet.
* SSH uses the client-server model, connecting a secure shell client application, the end at which the session is displayed, with an SSH server, the end at which the session runs.
* Fig. 5.9.1 shows SSH protocol stack.

Fig. 5.9.1 *SSH protocol stack*

* The SSH protocol consists of three major components :

1. **SSH transport layer protocol :** Provides server authentication, confidentiality and integrity with perfect forward secrecy.
2. **SSH user authentication protocol :** Authenticates the client to the server. It runs over the transport layer protocol.
3. **SSH connection protocol :**Multiplexes the encrypted tunnel into several logical channels.

## SSH port forwarding

* Port-Forwarding is also called tunnelling.
* The main benefit of port forwarding is that the tunnelled traffic between the user's computer and the remote server is encrypted through the SSH protocol.
* The SSH protocol V2 offers three types of port forwarding :
  1. **Local-to-remote forwarding:** Local (user side) port is created and all traffic is forwarded to a predefined destination server and port.
  2. **Remote-to-local forwarding:** A remote (server side) port is created and traffic from connections to that port are routed to the local (user) computer and is forwarded to a destination and port from there.
  3. **Dynamic port forwarding:** A local (user side) port is created and all traffic is forwarded to a destination server and port. The

server and port can be chosen at connection time.

* Fig. 5.9.2 shows setup flow of a secure shell connection.

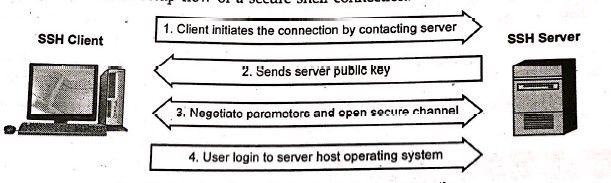


Fig. 5.9.2 *Setup flow of a secure shell connection*

* The protocol works in the client-server model, which means that the connection is established by the SSH client connecting to the SSH server.
* The SSH client drives the connection setup process and uses public key cryptography to verify the identity of the SSH server.
* After the setup phase the SSH protocol uses strong symmetric encryption and hashing algorithms to ensure the privacy and integrity of the data that is exchanged between the client and server.
* Once a connection has been established between the SSH client and server, the data that is transmitted is encrypted according to the parameters negotiated in the setup.
* During the negotiation the client and server agree on the symmetric encryption algorithm to be used and generate the encryption key that will be used.
* The keys used for authentication are called SSH keys.
* The protocol is used in networks for :

1. Providing secure access for users and automated processes.
2. Interactive and automated file transfers.
3. Issuing remote commands.
4. Managing network infrastructure and other mission-critical system components.

## Difference between SSH Version 1 and SSH Version 2

|  |  |
| --- | --- |
| It uses one monolithic protocol | It uses separate transport, authentication and  connection protocols. |
| Weak CRC-32 integrity check. | Strong cryptographic integrity check. |
| Exactly one session channel per connection. | Any number of session channels per  connection. |
| Negotiates only the bulk cipher; all others are fixed. | Full negotiation of modular cryptographic and compression algorithms, including bulk  encryption, MAC and public-key. |
| The same algorithms and keys are used in both directions | Encryption, MAC and compression are  negotiated separately for each direction, with independent keys. |
| Server key used for forward secrecy on the  session key. | Use of Diffie-Hellman key agreement  removes the need for a server key. |