Computer Networks notes

Information Technology (Jomo Kenyatta University of Agriculture and Technology)

1.1 Introduction to Computer Networks

A **computer network**, often simply referred to as a network, is a collection of hardware components and computers interconnected by communication channels that allow sharing of resources and information. Where at least one process in one device is able to send/receive data to/from at least one process residing in a remote device, then the two devices are said to be in a network. Networks may be classified according to a wide variety of characteristics such as the medium used to transport the data, communications protocol used, scale, topology, and organizational scope. Some basic types of computer networks include:

- A local area network (often called a LAN) connects two or more computers, and may be called a corporate network in an office or business setting.
- An "internetwork", sometimes called a Wide Area Network (because of the **wide** distance between networks) connects two or more smaller networks together. The largest internetwork is called the Internet.

Computers can be part of several different networks. Networks can also be parts of bigger networks. The local area networking a small business is usually connected to the corporate network of the larger company. Any connected machine at any level of the organization may be able to access the Internet, for example to demonstrate computers in the store, display its catalogue through a web server, or convert received orders into shipping instructions.

Microsoft Windows, Linux and most other operating systems use TCP/IP for networking. Apple Macintosh computers used AppleTalk in the past, but it uses TCP/IP now.

To set up a network an appropriate media is required. This can be wired or wireless. Twisted-pair, co-axial or fiber-optic are examples of cable and infra-red, blue-tooth, radio-wave, micro-wave etc. are wireless media used for networking. When you are working with a mere LAN, computers, media and peripherals are sufficient. But when you are working with a wider range you have use some additional devices like bridge, gateway or router to connect different small or large networks. And obviously a protocol must be maintained.

To set up a network you have to select an appropriate topology to arrange the hardware devices using the media. Topologies generally used are bus-topology, ring-topology, star-topology, tree-topology, object-oriented topology etc. Among these star-topology and tree-topology are most popular nowadays.

Properties

Computer networks:

i. ***Facilitate communications***
   - Using a network, people can communicate efficiently and easily via email, instant messaging, chat rooms, telephone, video telephone calls, and video conferencing.

ii. ***Permit sharing of files, data, and other types of information***
   - In a network environment, authorized users may access data and information stored on other computers on the network. The capability of providing access to data and information on shared storage devices is an important feature of many networks.

iii. ***Share network and computing resources***

In a networked environment, each computer on a network may access and use resources provided by devices on the network, such as printing a document on a shared network printer. Distributed computing uses computing resources across a network to accomplish tasks.

iv. ***May be insecure***
   - A computer network may be used by computer hackers to deploy computer viruses or computer worms on devices connected to the network, or to prevent these devices from normally accessing the network (denial of service).

v. ***May interfere with other technologies***
   - Power line communication strongly disturbs certain forms of radio communication, e.g., amateur radio. It may also interfere with last mile access technologies such as ADSL (Asymmetric digital subscriber line) and VDSL (Very-high-bit-rate digital subscriber line).

vi. ***May be difficult to set up***

- A complex computer network may be difficult to set up. It may also be very costly to set up an effective computer network in a large organization or company.

Type of Networks

1. Peer-to-peer

Peer-to-peer (**P2P**) computing or networking is a distributed application architecture that partitions tasks or workloads among peers. Peers are equally privileged, equipotent participants in the application. They are said to form a peer-to-peer network of nodes.

Or

A **peer-to-peer (P2P) network** is a type of decentralized and distributed network architecture in which individual nodes in the network (called *"peers"*) act as both suppliers and consumers of resources, in contrast to the centralized client–server model where client nodes request access to resources provided by central servers.

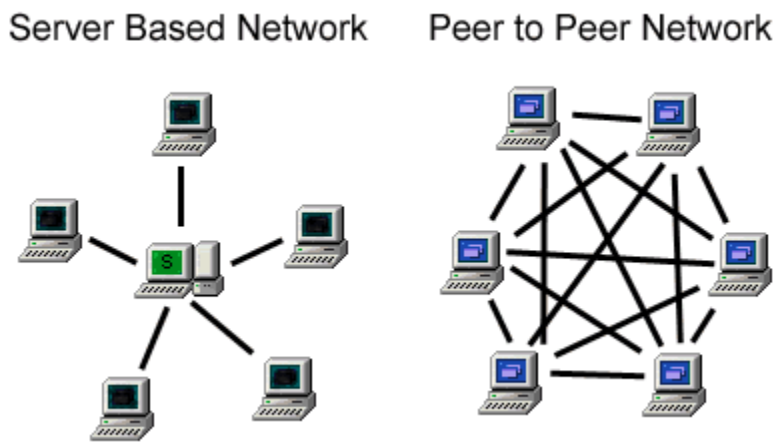
Peers make a portion of their resources, such as processing power, disk storage or network bandwidth, directly available to other network participants, without the need for central coordination by servers or stable hosts. Peers are both suppliers and consumers of resources, in contrast to the traditional client–server model where only servers supply (send), and clients consume (receive).

Networks in which all computers have equal status are called peer-to-peer or P2P networks.

In a peer-to-peer network, tasks (such as searching for files or streaming audio/video) are shared amongst multiple interconnected peers who each make a portion of their resources (such as processing power, disk storage or network bandwidth) directly available to other network participants, without the need for centralized coordination by servers.

The peer-to-peer application structure was popularized by file sharing systems like Napster. The concept has inspired new structures and philosophies in many areas of human interaction. Peer-to-peer networking is not restricted to technology, but covers also social processes with a peer-to-

peer dynamic. In such context, social peer-to-peer processes are currently emerging throughout society.



Advantages of a peer-to-peer network:

- **Less initial expense -** No need for a dedicated server.
- **Setup -** An operating system (such as Windows XP) already in place may only need to be reconfigured for peer-to-peer operations.

Disadvantages of a peer-to-peer network:

- **Decentralized -** No central repository for files and applications.
- **Security -** Does not provide the security available on a client/server network.

Current applications

Peer-to-peer networks underlie numerous applications. The most commonly known application is file sharing, which popularized the technology.

Communications

- Instant messaging systems and online chat networks.

Content delivery

In P2P networks, clients provide resources as well as using them. This means that unlike client-server systems, the content serving capacity of peer-to-peer networks can actually increase as more users begin to access the content (especially with protocols such as Bit torrent that require users to share). This property is one of the major advantages of using P2P networks because it makes the setup and running costs very small for the original content distributor.

File-sharing networks

Many file peer-to-peer file sharing networks, such as Gnutella, G2, and the eDonkey network popularized peer-to-peer technologies. From 2004 on, such networks form the largest contributor of network traffic on the Internet.

- Peer-to-peer content delivery networks
- Peer-to-peer content services, e.g. caches for improved performance
- Software publication and distribution (Linux distribution, several games); via file sharing networks.

Streaming media

- Streaming media. P2PTV
- Applications include TVUPlayer, Joost, CoolStreaming, Cybersky-TV, PPLive, LiveStation, Giraffic and Didiom
- Spotify uses a peer-to-peer network along with streaming servers to stream music to its desktop music player
- Peercasting for multicasting streams
- Osiris (Serverless Portal System) allows its users to create anonymous and autonomous web portals distributed via P2P network.

Architecture

A peer-to-peer network is designed around the notion of equal peer nodes simultaneously functioning as both "clients" and "servers" to the other nodes on the network. This model of network arrangement differs from the client–server model where communication is usually to and from a central server. A typical example of a file transfer that uses the client-server model is the File Transfer Protocol (FTP) service in which the client and server programs are distinct: the clients initiate the transfer, and the servers satisfy these requests.

Three categories can easily be seen: -

- **In pure peer-to-peer systems** the entire network consists solely of equipotent peers. There is only one routing layer, as there are no preferred nodes with any special infrastructure function.
- **Hybrid peer-to-peer systems** allow such infrastructure nodes to exist often called super-nodes.
- **In centralized peer-to-peer systems**, a central server is used for indexing functions and to bootstrap the entire system. Although this has similarities with a structured architecture, the connections between peers are not determined by any algorithm.

a. Unstructured networks

Unstructured peer-to-peer networks do not impose a particular structure on the overlay network by design, but rather are formed by nodes that randomly form connections to each other. (Gnutella, Gossip, and Kazaa are examples of unstructured P2P protocols)

Because there is no structure globally imposed upon them, unstructured networks are easy to build and allow for localized optimizations to different regions of the overlay. Also, because the role of all peers in the network is the same, unstructured networks are highly robust in the face of high rates of "churn"—that is, when large numbers of peers are frequently joining and leaving the network.

However the primary limitations of unstructured networks also arise from this lack of structure. In particular, when a peer wants to find a desired piece of data in the network, the search query must be flooded through the network to find as many peers as possible that share the data. Flooding causes a very high amount of signaling traffic in the network, uses more CPU/memory (by requiring every peer to process all search queries), and does not ensure that search queries will always be resolved. Furthermore, since there is no correlation between a peer and the content managed by it, there is no guarantee that flooding will find a peer that has the desired data.

b. Structured Networks

Structured P2P networks employ a globally consistent protocol to ensure that any node can efficiently route a search to some peer that has the desired file, even if the file is extremely rare. Such a guarantee necessitates a more structured pattern of overlay links. By far the most common type of structured P2P network is the distributed hash table (DHT), in which a variant of consistent hashing is used to assign ownership of each file to a particular peer, in a way

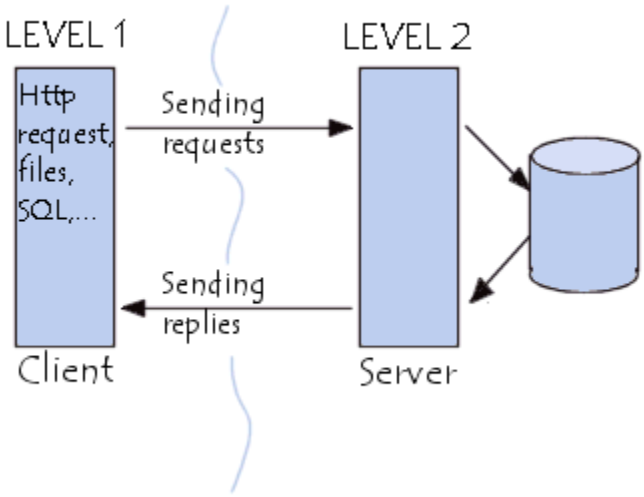
analogous to a traditional hash table's assignment of each key to a particular array slot. Though the term DHT is commonly used to refer to the structured overlay, in practice, DHT is a data structured implemented on top of a structured overlay

2. Client–server Model

The client–server model is a computing model that acts as distributed application which partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server machine is a host that is running one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.

The client–server characteristic describes the relationship of cooperating programs in an application. The server component provides a function or service to one or many clients, which initiate requests for such services.

Functions such as email exchange, web access and database access, are built on the client–server model. Users accessing banking services from their computer use a web browser client to send a request to a web server at a bank. That program may in turn forward the request to its own database client program that sends a request to a database server at another bank computer to retrieve the account information. The balance is returned to the bank database client, which in turn serves it back to the web browser client displaying the results to the user. The client–server model has become one of the central ideas of network computing. Many business applications being written today use the client–server model. So do the Internet's main application protocols, such as HTTP (Hypertext Transfer Protocol), SMTP (Simple Mail Transfer Protocol), Telnet, and DNS (Domain Name System).



Advantages of a client/server network:

- **Centralized -** Resources and data security are controlled through the server
- **Scalability -** Any or all elements can be replaced individually as needs increase
- **Flexibility -** New technology can be easily integrated into system
- **Interoperability -** All components (client/network/server) work together
- **Accessibility -** Server can be accessed remotely and across multiple platforms

Disadvantages of a client/server network:

- **Expense -** Requires initial investment in dedicated server
- **Maintenance -** Large networks will require a staff to ensure efficient operation
- **Dependence -** When server goes down, operations will cease across the network

Client and server roles

The *client–server* characteristic describes the relationship of cooperating programs in an application. The server component provides a function or service to one or many clients, which initiate requests for such services.

Servers are classified by the services they provide. For instance, a web server serves web pages and a file server serves computer files. A shared resource may be any of the server computer's software and electronic components,
from programs and data to processors and storage devices. The sharing of resources of a server constitutes a *service*.

Whether a computer is a client, a server, or both, is determined by the nature of the application that requires the service functions. For example, a single computer can run web server and file

server software at the same time to serve different data to clients making different kinds of requests. Client software can also communicate with server software within the same computer. Communication between servers, such as to synchronize data, is sometimes called *inter-server* or *server-to-server* communication.

Client and server communication

In general, a service is an abstraction of computer resources and a client does not have to be concerned with how the server performs while fulfilling the request and delivering the response. The client only has to understand the response based on the well-known application protocol, i.e. the content and the formatting of the data for the requested service.

Clients and servers exchange messages in a request-response messaging pattern:

- The client sends a request, and the server returns a response. This exchange of messages is an example of inter-process communication.
- To communicate, the computers must have a common language, and they must follow rules so that both the client and the server know what to expect. The language and rules of communication are defined in a communications protocol. All client-server protocols operate in the application layer. The application-layer protocol defines the basic patterns of the dialogue. To formalize the data exchange even further, the server may implement an API (such as a web service). The API is an abstraction layer for such resources as databases and custom software. By restricting communication to a specific content format, it facilitates parsing. By abstracting access, it facilitates cross-platform data exchange.
- A server may receive requests from many different clients in a very short period of time. Because the computer can perform a limited number of tasks at any moment, it relies on a scheduling system to prioritize incoming requests from clients in order to accommodate them all in turn. To prevent abuse and maximize uptime, the server's software limits how a client can use the server's resources. Even so, a server is not immune from abuse. A denial of service attack exploits a server's obligation to process requests by bombarding it with requests incessantly. This inhibits the server's ability to responding to legitimate requests.

Example

When a bank customer accesses online banking services with a web browser (the client), they initiate a request to the bank's web server. Since the customer's login credentials are stored in a database, the web server runs a program to access a database server. This database server may, in turn, fetch financial transaction records from another database server. An application server interprets the returned data by following the bank's business logic, and provides the output to the web server. Finally, the web server sends the result to the web browser, which interprets the data.

Each server listed above acts as a client when it submits data in a request to another server for processing. In each step of this sequence of client–server message exchanges, a computer processes a request and returns data. This is the request-response messaging pattern. When all the requests are met, the sequence is complete and the web browser presents the data to the customer.

This example illustrates a design pattern applicable to the client–server model: separation of concerns.

Comparison with peer-to-peer architecture

In addition to the client-server model, distributed computing applications often use the peer-to-peer application architecture.

In the client–server model, the server is often designed to be a centralized system that serves many clients. The computing power, memory and storage requirements of a server must be scaled appropriately to the expected work load, i.e. the number of clients connecting simultaneously. Load balancing and failover systems are often employed to scale the server implementation.

In a peer-to-peer (P2P) network, two or more computers (*peers*) pool their resources and communicate in a decentralized system. Peers are coequal or equipotent nodes in a non-hierarchical network. Unlike clients in a client–server or client–queue–client network, peers communicate with each other directly. In peer-to-peer networking, an algorithm in the peer-to-peer communications protocol balances load, and even peers with modest resources can help to share the load. If a node becomes unavailable, its shared resources remain available as long as other peer offers it. Ideally, a peer does not need to achieve high availability because other,

redundant peers make up for any resource downtime; as the availability and load capacity of peers change, the protocol reroutes requests.

CHAPTER 2

Network Topologies

Network topology is the arrangement of the various elements (links, nodes, etc.) of a computer network. Essentially, it is the topological structure of a network, and may be depicted physically or logically.

Physical topology refers to the placement of the network's various components, including device location and cable installation, while *logical* topology shows how data flows within a network, regardless of its physical design. Distances between nodes, physical interconnections, transmission rates, and/or signal types may differ between two networks, yet their topologies may be identical.

A good example is a local area network (LAN): Any given node in the LAN has one or more physical links to other devices in the network; graphically mapping these links results in a geometric shape that can be used to describe the physical topology of the network. Conversely, mapping the data flow between the components determines the logical topology of the network.

Topology

There are two basic categories of network topologies:

1. Physical topologies
2. Logical topologies

The shape of the cabling layout used to link devices is called the physical topology of the network. This refers to the layout of cabling, the locations of nodes, and the interconnections between the nodes and the cabling. The physical topology of a network is determined by the capabilities of the network access devices and media, the level of control or fault tolerance desired, and the cost associated with cabling or telecommunications circuits.

The logical topology in contrast, is the way that the signals act on the network media, or the way that the data passes through the network from one device to the next without regard to the physical interconnection of the devices. A network's logical topology is not necessarily the same as its physical topology. For example, the original twisted pair Ethernet using repeater hubs was a logical bus topology with a physical star topology layout. Token Ring is a logical ring topology, but is wired a physical star from the Media Access Unit.

The logical classification of network topologies generally follows the same classifications as those in the physical classifications of network topologies but describes the path that the *data* takes between nodes being used as opposed to the actual *physical* connections between nodes. The logical topologies are generally determined by network protocols as opposed to being determined by the physical layout of cables, wires, and network devices or by the flow of the electrical signals, although in many cases the paths that the electrical signals take between nodes may closely match the logical flow of data, hence the convention of using the terms *logical topology* and *signal topology* interchangeably.

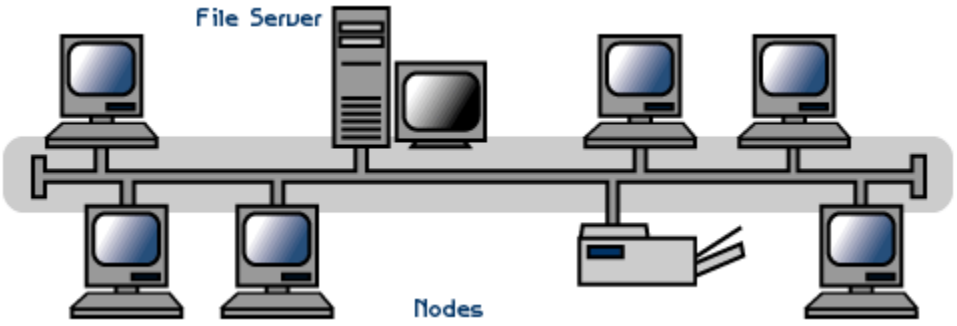
Logical topologies are often closely associated with Media Access Control methods and protocols. Logical topologies are able to be dynamically reconfigured by special types of equipment such as routers and switches.

Main Types of Physical Topologies

Linear Bus

A linear bus topology consists of a main run of cable with a terminator at each end

All nodes (file server, workstations, and peripherals) are connected to the linear cable.



Advantages of a Linear Bus Topology

- Easy to connect a computer or peripheral to a linear bus
- Requires less cable length than a star topology

Disadvantages of a Linear Bus Topology

- Entire network shuts down if there is a break in the main cable
- Terminators are required at both ends of the backbone cable
- Difficult to identify the problem if the entire network shuts down
- Not meant to be used as a stand-alone solution in a large building

Star

A star topology is designed with each node (file server, workstations, and peripherals) connected directly to a central network hub, switch, or concentrator.

Data on a star network passes through the hub, switch, or concentrator before continuing to its destination. The hub, switch, or concentrator manages and controls all functions of the network. It also acts as a repeater for the data flow. This configuration is common with twisted pair cable; however, it can also be used with coaxial cable or fiber optic cable.

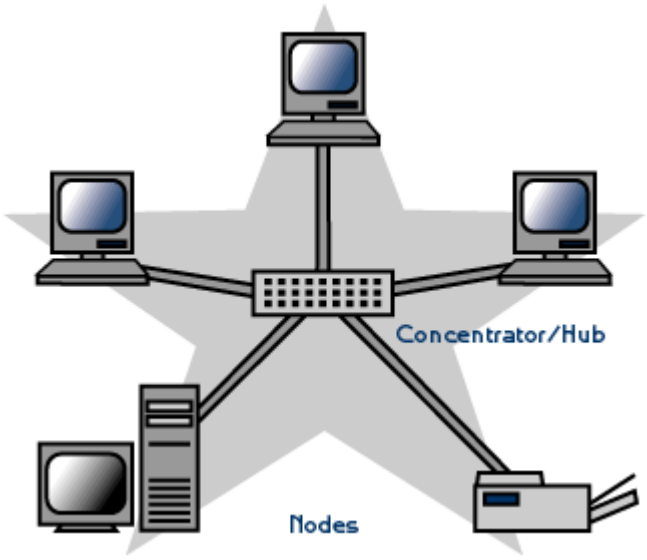


Fig. 2. Star topology

Advantages of a Star Topology

- Easy to install and wire
- No disruptions to the network when connecting or removing devices
- Easy to detect faults and to remove parts

Disadvantages of a Star Topology

- Requires more cable length than a linear topology.
- If the hub, switch, or concentrator fails, nodes attached are disabled.
- More expensive than linear bus topologies because of the cost of the hubs, etc.

Tree or Expanded Star

A tree topology combines characteristics of linear bus and star topologies. It consists of groups of star-configured workstations connected to a linear bus backbone cable (See fig. 3). Tree topologies allow for the expansion of an existing network, and enable schools to configure a network to meet their needs.

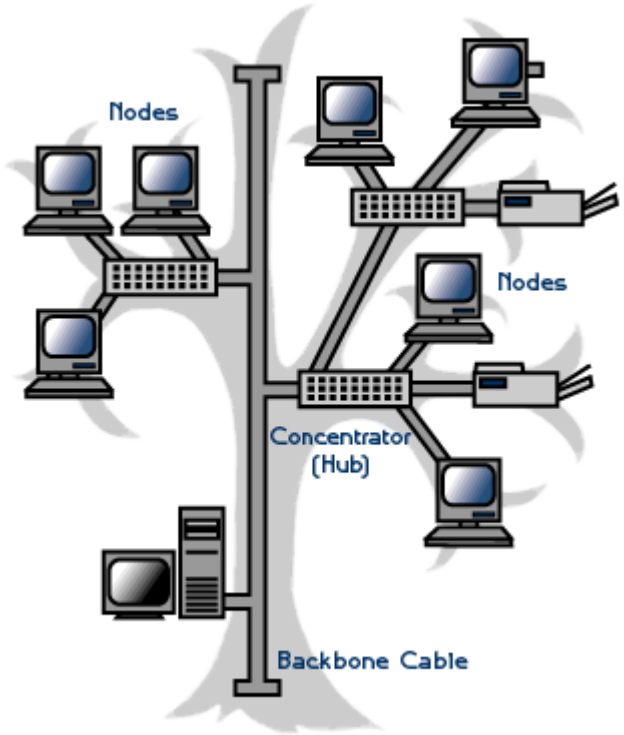


Fig. 3. Tree topology

Advantages of a Tree Topology

- Point-to-point wiring for individual segments
- Supported by several hardware and software venders

Disadvantages of a Tree Topology

- Overall length of each segment is limited by the type of cabling used
- If the backbone line breaks, the entire segment goes down
- More difficult to configure and wire than other topologies

5-4-3 Rule

A consideration in setting up a tree topology using Ethernet protocol is the 5-4-3 rule. One aspect of the Ethernet protocol requires that a signal sent out on the network cable reach every part of the network within a specified length of time. Each concentrator or repeater that a signal goes through adds a small amount of time. This leads to the rule that between any two nodes on the network there can only be a maximum of 5 segments, connected through 4 repeaters/concentrators. In addition, only 3 of the segments may be populated (trunk) segments if they are made of coaxial cable. A populated segment is one that has one or more nodes attached to it. The 5-4-3 rule is adhered to. The furthest two nodes on the network have 4 segments and 3 repeaters/concentrators between them.

NOTE: This rule does not apply to other network protocols or Ethernet networks where all fiber optic cabling or a combination of a fiber backbone with UTP cabling is used. If there is a combination of fiber optic backbone and UTP cabling, the rule would translate to a 7-6-5 rule. The speed of networking switches is vastly improved over older technologies, and while every effort should be made to limit network segment traversal, efficient switching can allow much larger numbers of segments to be traversed with little or no impact to the network.

Considerations When Choosing a Topology

- **Money**: - A linear bus network may be the least expensive way to install a network; you do not have to purchase concentrators
- **Length of cable needed: -** The linear bus network uses shorter lengths of cable
- **Future growth: -** With a star topology, expanding a network is easily done by adding another concentrator
- C**able type: -** The most common cable in schools is unshielded twisted pair, which is most often used with star topologies

CHAPTER 3

NETWORK HARD & SOFTWARE'S

Network Hardware

Networking hardware may also be known as **network equipment** or **computer networking devices**. Units which are the last receiver or generate data are called hosts or data terminal equipment.

All these terms refer to devices facilitating the use of a computer network. Specifically, they mediate data in a computer network.

Specific devices

- **Gateway:** this device is placed at a network node and interfaces with another network that uses different protocols. It works on OSI layers 4 to 7.
- **Router:** a specialized network device that determines the next network point to which it can forward a data packet towards the ultimate destination of the packet. Unlike a gateway, it cannot interface different protocols. It works on OSI layer 3.
- **Switch:** a device that allocates traffic from one network segment to certain lines (intended destination(s)) which connect the segment to another network segment. Unlike a hub, a switch splits the network traffic and sends it to different destinations rather than to all systems on the network. It works on OSI layer 2.
- **Bridge:** a device that connects multiple network segments along the data link layer. It works on OSI layer 2.
- **Hub:** a device that connects multiple Ethernet segments, making them acts as a single segment. When using a hub, every attached device shares the same broadcast domain and the same collision domain. Therefore, only one computer connected to the hub is able to transmit at a time. Depending on the network topology, the hub provides a basic level 1 OSI model connection among the network objects (workstations, servers, etc.). It provides bandwidth which is shared among all the objects, in contrast to switches, which provide a connection between individual nodes. It works on OSI layer 1.
- **Repeater:** a device which amplifies or regenerates digital signals received while sending them from one part of a network into another. It works on OSI layer 1.

Some hybrid network devices:

- **Multilayer switch:** a switch which, in addition to switching on OSI layer 2, provides functionality at higher protocol layers.
- **Protocol converter:** a hardware device that converts between two different types of transmission, such as asynchronous and synchronous transmissions.
- **Bridge router (Brouter):** a device that combines router and bridge functionality and therefore works on OSI layers 2 and 3.

Hardware or software components that typically sit on the connection point of different networks, e.g. between an internal network and an external network:

- **Proxy server:** computer network service which allows clients to make indirect network connections to other network services.
- **Firewall:** a piece of hardware or software put on the network to prevent some communications forbidden by the network policy.
- **Network address translator (NAT):** network service provided as hardware or software that converts internal to external network addresses and vice versa.

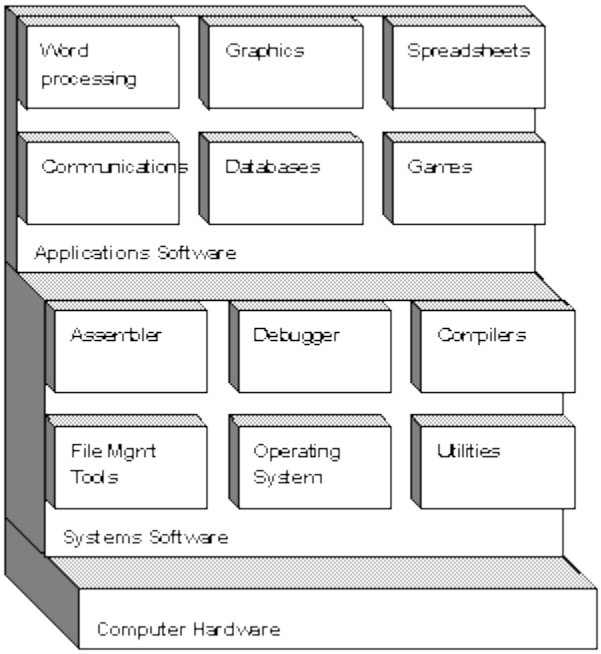
Other hardware for establishing networks or dial-up connections:

- **Multiplexer:** a device that combines several electrical signals into a single signal
- Network interface controller: a device connecting the attached computer to a wire-based computer network
- **Wireless network interface controller:** a device connecting the attached computer to a radio-based computer network
- **Modem:** device that modulates an analog "carrier" signal (such as sound) to encode digital information, and that also demodulates such a carrier signal to decode the transmitted information, such as a computer communicating with another computer over a telephone network
- **ISDN (Integrated Services for Digital Network) terminal adapter (TA):** a specialized gateway for ISDN
- **Line driver:** a device to increase transmission distance by amplifying the signal; used in base-band networks only.

Network Software

A general phrase for software that is designed to help set up, manage, and/or monitor computer networks. Networking software applications are available to manage and monitor networks of all sizes, from the smallest home networks to the largest enterprise networks.

Application (application software)



An application is a program, or group of programs, that is designed for the end user. Application software can be divided into two general classes: *systems software* and *applications software*. Applications software (also called *end-user programs*) includes such things as database programs, word processors, Web browsers and spreadsheets.

Network operating system (NOS)

Network operating system refers to software that implements an operating system of some kind that is oriented to computer networking. For example, one that runs on a server and enables the server to manage data, users, groups, security, applications, and other networking functions. The network operating system is designed to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks.

Use in routers

Network operating systems can be embedded in a router or hardware firewall that operates the functions in the network layer (layer 3) of the OSI model.

Examples:

- JUNOS, used in routers and switches from Juniper Networks,
- Cisco IOS (formerly "Cisco Internetwork Operating System").
- TiMOS, used in routers from Alcatel-Lucent
- VRP (Versatile Routing Platform), used in routers from Huawei
- RouterOS, software which turns a PC or MikroTik hardware into a dedicated router
- ZyNOS, used in network devices made by ZyXEL.
- ExtremeXOS, used in network devices made by Extreme Networks. Also called EXOS.
- Embedded Linux, in distributions like Openwrt and DD-WRT which run on low-cost platforms such as the Linksys WRT54G.

Peer-to-Peer

In a peer-to-peer network operating system users are allowed to share resources and files located on their computers and access shared resources from others. This system is not based with having a file server or centralized management source. A peer-to-peer network sets all connected computers equal; they all share the same abilities to use resources available on the network.

Examples:

- AppleShare used for networking connecting Apple products.
- Windows for Workgroups used for networking peer-to-peer windows computers.

Advantages

- Ease of setup
- Less hardware needed, no server needs to be purchased.

Disadvantages

- No central location for storage
- Lack of security that a client/server type offers

Client/Server

Network operating systems can be based on a client/server architecture in which a server enables multiple clients to share resources. Client/server network operating systems allow the network to centralize functions and applications in one or more dedicated file servers. The server is the center of the system, allowing access to resources and instituting security. The network operating system provides the mechanism to integrate all the components on a network to allow multiple users to simultaneously share the same resources regardless of physical location.

Examples:

- Novell NetWare
- Windows Server
- Banyan VINES

Advantages

- Centralized servers are more stable
- Security is provided through the server
- New technology and hardware can be easily integrated into the system
- Servers are able to be accessed remotely from different locations and types of systems

Disadvantages

- Cost of buying and running a server are high
- Dependence on a central location for operation
- Requires regular maintenance and updates

Network Management

In computer networks, network management refers to the activities, methods, procedures, and tools that pertain to the operation, administration, maintenance, and provisioning of networked systems. Network management is essential to command and control practices and is generally carried out of a network operations center.

- Operation deals with keeping the network (and the services that the network provides) up and running smoothly. It includes monitoring the network to spot problems as soon as possible, ideally before users are affected.

- Administration deals with keeping track of resources in the network and how they are assigned. It includes all the "housekeeping" that is necessary to keep the network under control.
- Maintenance is concerned with performing repairs and upgrades—for example, when equipment must be replaced, when a router needs a patch for an operating system image, when a new switch is added to a network. Maintenance also involves corrective and preventive measures to make the managed network run "better", such as adjusting device configuration parameters.
- Provisioning is concerned with configuring resources in the network to support a given service. For example, this might include setting up the network so that a new customer can receive voice service, real time communications etc.

A common way of characterizing network management functions is FCAPS—Fault, Configuration, Accounting, Performance and Security.

Functions that are performed as part of network management accordingly include controlling, planning, allocating, deploying, coordinating, and monitoring the resources of a network, network planning, frequency allocation, predetermined traffic routing to support load balancing, cryptographic key distribution authorization, configuration management, fault management, security management, performance management, bandwidth management, Route analytics and accounting management.

Data for network management is collected through several mechanisms, including agents installed on infrastructure, synthetic monitoring that simulates transactions, logs of activity, sniffers and real user monitoring. In the past network management mainly consisted of monitoring whether devices were up or down; today performance management has become a crucial part of the IT team's role which brings about a host of challenges—especially for global organizations.

Note: Network management does not include user terminal equipment.

CHAPTER 4

Data communications (Introduction)

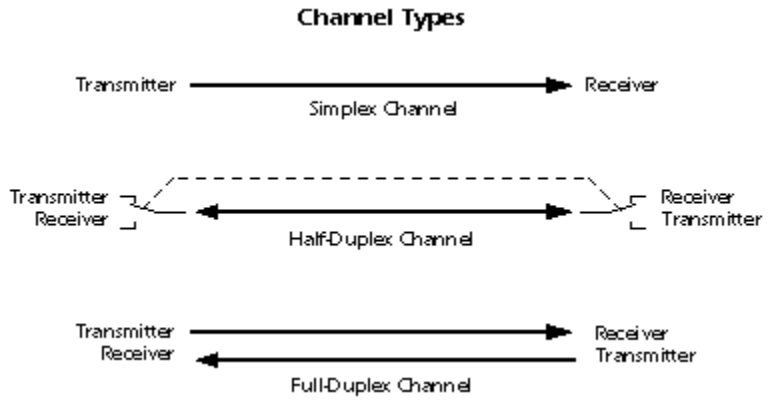
Data Communications concerns the transmission of digital messages to devices external to the message source. "External" devices are generally thought of as being independently powered circuitry that exists beyond the chassis of a computer or other digital message source. As a rule, the maximum permissible transmission rate of a message is directly proportional to signal power, and inversely proportional to channel noise. It is the aim of any communications system to provide the highest possible transmission rate at the lowest possible power and with the least possible noise.

Communications Channels

A communications channel is a pathway over which information can be conveyed. It may be defined by a physical wire that connects communicating devices, or by a radio, laser, or other radiated energy source that has no obvious physical presence. Information sent through a communications channel has a source from which the information originates, and a destination to which the information is delivered. Although information originates from a single source, there may be more than one destination, depending upon how many receive stations are linked to the channel and how much energy the transmitted signal possesses.

In a digital communications channel, the information is represented by individual data bits, which may be encapsulated into multibit message units. A byte, which consists of eight bits, is an example of a message unit that may be conveyed through a digital communications channel. A collection of bytes may itself be grouped into a frame or other higher-level message unit. Such multiple levels of encapsulation facilitate the handling of messages in a complex data communications network.

Any communications channel has a direction associated with it:



The message source is the transmitter, and the destination is the receiver. A channel whose direction of transmission is unchanging is referred to as a **simplex channel**. For example, a radio station is a simplex channel because it always transmits the signal to its listeners and never allows them to transmit back.

A half-duplex channel: - is a single physical channel in which the direction may be reversed. Messages may flow in two directions, but never at the same time, in a half-duplex system. In a telephone call, one party speaks while the other listens. After a pause, the other party speaks and the first party listens. Speaking simultaneously results in garbled sound that cannot be understood.

A full-duplex channel: - allows simultaneous message exchange in both directions. It really consists of two simplex channels, a forward channel and a reverse channel, linking the same points. The transmission rate of the reverse channel may be slower if it is used only for flow control of the forward channel.

Asynchronous vs. Synchronous Transmission

Serialized data is not generally sent at a uniform rate through a channel. Instead, there is usually a burst of regularly spaced binary data bits followed by a pause, after which the data flow resumes. Packets of binary data are sent in this manner, possibly with variable-length pauses between packets, until the message has been fully transmitted. In order for the receiving end to know the proper moment to read individual binary bits from the channel, it must know exactly when a packet begins and how much time elapses between bits. When this timing information is known, the receiver is said to be synchronized with the transmitter, and accurate data transfer

becomes possible. Failure to remain synchronized throughout a transmission will cause data to be corrupted or lost.

Two basic techniques are employed to ensure correct synchronization. In synchronous systems, separate channels are used to transmit data and timing information. The timing channel transmits clock pulses to the receiver. Upon receipt of a clock pulse, the receiver reads the data channel and latches the bit value found on the channel at that moment. The data channel is not read again until the next clock pulse arrives. Because the transmitter originates both the data and the timing pulses, the receiver will read the data channel only when told to do so by the transmitter (via the clock pulse), and synchronization is guaranteed.

Techniques exist to merge the timing signal with the data so that only a single channel is required. This is especially useful when synchronous transmissions are to be sent through a modem. Two methods in which a data signal is self-timed are nonreturn-to-zero and biphase Manchester coding. These both refer to methods for encoding a data stream into an electrical waveform for transmission.

In asynchronous systems, a separate timing channel is not used. The transmitter and receiver must be preset in advance to an agreed-upon baud rate. A very accurate local oscillator within the receiver will then generate an internal clock signal that is equal to the transmitter's within a fraction of a percent. For the most common serial protocol, data is sent in small packets of 10 or 11 bits, eight of which constitute message information. When the channel is idle, the signal voltage corresponds to a continuous logic '1'. A data packet always begins with a logic '0' (the start bit) to signal the receiver that a transmission is starting. The start bit triggers an internal timer in the receiver that generates the needed clock pulses. Following the start bit, eight bits of message data are sent bit by bit at the agreed upon baud rate. The packet is concluded with a parity bit and stop bit.

Basic Hardware Components

Apart from the physical communications media themselves, networks comprise additional basic hardware building blocks interconnecting their terminals, such as network interface cards (NICs), hubs, bridges, switches, and routers

Network Interface Cards

A network card, network adapter, or NIC (network interface card) is a piece of computer hardware designed to allow computers to physically access a networking medium. It provides a low-level addressing system through the use of MAC addresses.

Each Ethernet network interface has a unique MAC address which is usually stored in a small memory device on the card, allowing any device to connect to the network without creating an address conflict.

Repeaters and Hubs

A repeater is an electronic device that receives a signal, cleans it of unnecessary noise, regenerates it, and retransmits it at a higher power level, or to the other side of an obstruction, so that the signal can cover longer distances without degradation. In most twisted pair Ethernet configurations, repeaters are required for cable that runs longer than 100 meters. A repeater with multiple ports is known as a hub. Repeaters work on the Physical Layer of the OSI model. Repeaters require a small amount of time to regenerate the signal. This can cause a propagation delay which can affect network communication when there are several repeaters in a row. Many network architectures limit the number of repeaters that can be used in a row (e.g. Ethernet's 5-4-3 rule).

Bridges

A network bridge connects multiple network segments at the data link layer (layer 2) of the OSI model. Bridges broadcast to all ports except the port on which the broadcast was received. However, bridges do not promiscuously copy traffic to all ports, as hubs do, but learn which MAC addresses are reachable through specific ports. Once the bridge associates a port and an address, it will send traffic for that address to that port only.

Bridges come in three basic types:

- **Local bridges:** Directly connect LANs

- **Remote bridges:** Can be used to create a wide area network (WAN) link between LANs. Remote bridges, where the connecting link is slower than the end networks, largely have been replaced with routers.
- **Wireless bridges:** Can be used to join LANs or connect remote stations to LANs.

Switches

A network switch is a device that forwards and filters OSI layer 2 datagrams (chunks of data communication) between ports (connected cables) based on the MAC addresses in the packets. A switch is distinct from a hub in that it only forwards the frames to the ports involved in the communication rather than all ports connected. A switch breaks the collision domain but represents itself as a broadcast domain. Switches make forwarding decisions of frames on the basis of MAC addresses. A switch normally has numerous ports, facilitating a star topology for devices, and cascading additional switches. Some switches are capable of routing based on Layer 3 addressing or additional logical levels; these are called multi-layer switches.

Routers

A router is an internetworking device that forwards packets between networks by processing information found in the datagram or packet (Internet protocol information from Layer 3 of the OSI Model). In many situations, this information is processed in conjunction with the routing table (also known as forwarding table). Routers use routing tables to determine what interface to forward packets (this can include the "null" also known as the "black hole" interface because data can go into it, however, no further processing is done for said data).

Data transmission - Transmission modes

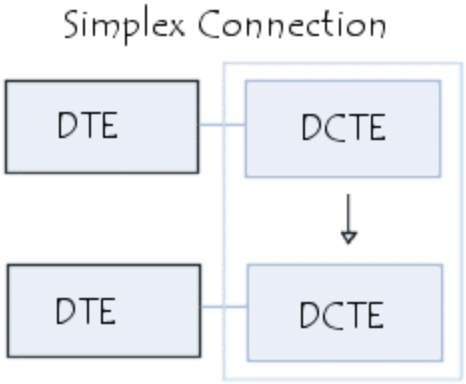
A given transmission on a communications channel between two machines can occur in several different ways. The transmission is characterised by:

- The direction of the exchanges
- The transmission mode: the number of bits sent simultaneously
- Synchronization between the transmitter and receiver

There are 3 different transmission modes characterized according to the direction of the exchanges:

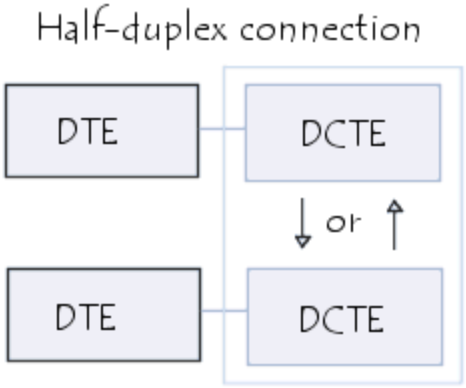
a. A simplex connection is a connection in which the data flows in only one direction, from the transmitter to the receiver. This type of connection is useful if

the data do not need to flow in both directions (for example, from your computer to the printer or from the mouse to your computer...).

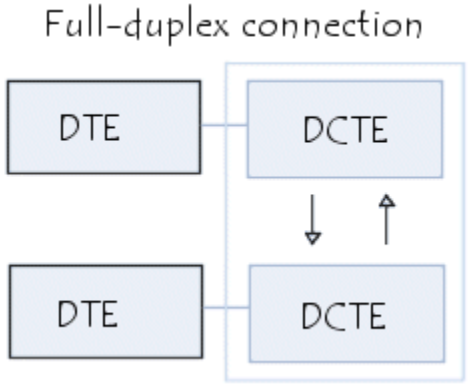


Nb: *DTE - Data terminal equipment*
DCTE - data circuit-terminating equipment

b. A half-duplex connection (sometimes called an alternating connection orsemi-duplex) is a connection in which the data flows in one direction or the other, but not both at the same time. With this type of connection, each end of the connection transmits in turn. This type of connection makes it possible to have bidirectional communications using the full capacity of the line.



c. **A full-duplex connection** is a connection in which the data flow in both directions simultaneously. Each end of the line can thus transmit and receive at the same time, which means that the bandwidth is divided in two for each direction of data transmission if the same transmission medium is used for both directions of transmission.

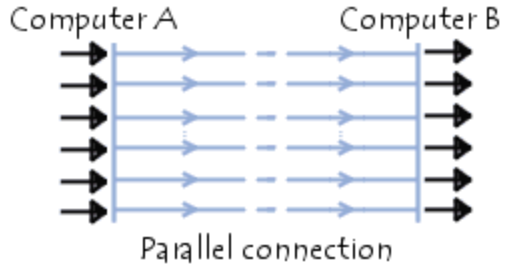


Serial and Parallel Transmission

The transmission mode refers to the number of elementary units of information (bits) that can be simultaneously translated by the communications channel. In fact, processors (and therefore computers in general) never process (in the case of recent processors) a single bit at a time; generally they are able to process several (most of the time it is 8: one byte), and for this reason the basic connections on a computer are parallel connections.

Parallel connection

Parallel connection means simultaneous transmission of N bits. These bits are sent simultaneously over N different channels (a channel being, for example, a wire, a cable or any other physical medium). The parallel connection on PC-type computers generally requires 10 wires.



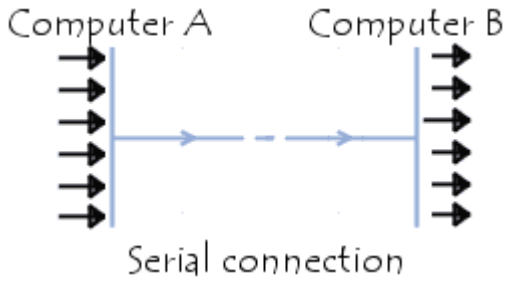
These channels may be:

- N physical lines: in which case each bit is sent on a physical line (which is why parallel cables are made up of several wires in a ribbon cable)
- One physical line divided into several sub-channels by dividing up the bandwidth. In this case, each bit is sent at a different frequency...

Since the conductive wires are close to each other in the ribbon cable, interference can occur (particularly at high speeds) and degrade the signal quality...

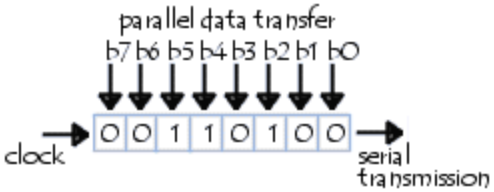
Serial connection

In a serial connection, the data are sent one bit at a time over the transmission channel. However, since most processors process data in parallel, the transmitter needs to transform incoming parallel data into serial data and the receiver needs to do the opposite.

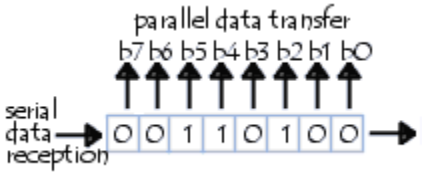


These operations are performed by a communications controller (normally aUART (Universal Asynchronous Receiver Transmitter) chip). The communications controller works in the following manner:

- The parallel-serial transformation is performed using a shift register. The shift register, working together with a clock, will shift the register (containing all of the data presented in parallel) by one position to the left, and then transmit the most significant bit (the leftmost one) and so on:



- The serial-parallel transformation is done in almost the same way using a shift register. The shift register shifts the register by one position to the left each time a bit is received, and then transmits the entire register in parallel when it is full:



Synchronous and asynchronous transmission

Given the problems that arise with a parallel-type connection, serial connections are normally used. However, since a single wire transports the information, the problem is how to synchronize the transmitter and receiver, in other words, the receiver can not necessarily distinguish the characters (or more generally the bit sequences) because the bits are sent one after the other. There are two types of transmission that address this problem:

- **An asynchronous connection**, in which each character is sent at irregular intervals in time (for example a user sending characters entered at the keyboard in real time). So, for example, imagine that a single bit is transmitted during a long period of silence... the receiver will not be able to know if this is 00010000, 10000000 or 00000100...

To remedy this problem, each character is preceded by some information indicating the start of character transmission (the transmission start information is called a START bit) and ends by sending end-of-transmission information (called STOP bit, there may even be several STOP bits).

- In a synchronous connection, the transmitter and receiver are paced by the same clock. The receiver continuously receives (even when no bits are transmitted) the information at the same rate the transmitter send it. This is why the transmitter and receiver are paced at the same speed. In addition, supplementary information is inserted to guarantee that there are no errors during transmission.

During synchronous transmission, the bits are sent successively with no separation between each character, so it is necessary to insert synchronization elements; this is called character**-level synchronization**.
The main disadvantage of synchronous transmission is recognizing the data at the receiver, as there may be differences between the transmitter and receiver clocks. That is why each data transmission must be sustained long enough for the receiver to distinguish it. As a result, the transmission speed cannot be very high in a synchronous link.

CHAPTER 5

Signal Transmission

In telecommunications, **transmission** (abbreviation: **Tx**) is the process of sending and propagating an analogue or digital information signal over a physical point-to-point or point-to-multipoint transmission medium, either wired, optical fiber or wireless. Transmission technologies and schemes typically refer to physical layer protocol duties such as modulation, demodulation, line coding, equalization, error control, bit synchronization and multiplexing, but the term may also involve higher-layer protocol duties, for example, digitizing an analog message signal, and source coding (compression).

Transmission of a digital message, or of a digitized analog signal, is known as data transmission or digital communication.

One transmission is the sending of a signal with limited duration, for example a block or packet of data, a phone call, or an email.

Analog signal

An **analog** or **analogue signal** is any continuous signal for which the time varying feature (variable) of the signal is a representation of some other time varying quantity, i.e., analogous to another time varying signal. For example, in an analog audio signal, the instantaneous voltage of the signal varies continuously with the pressure of the sound waves. It differs from a digital signal, in which a continuous quantity is represented by a discrete function which can only take on one of a finite number of values. The term analog signal usually refers to electrical signals; however, mechanical, pneumatic, hydraulic, and other systems may also convey analog signals.

An analog signal uses some property of the medium to convey the signal's information. For example, an aneroid barometer uses rotary position as the signal to convey pressure information. In an electrical signal, the voltage, current, or frequency of the signal may be varied to represent the information.

Any information may be conveyed by an analog signal; often such a signal is a measured response to changes in physical phenomena, such as sound, light, temperature, position, or pressure. The physical variable is converted to an analog signal by a transducer. For example, in sound recording, fluctuations in air pressure (that is to say, sound) strike the diaphragm of

a microphone which induces corresponding fluctuations in the current produced by a coil in an electromagnetic microphone, or the voltage produced by a condenser microphone. The voltage or the current is said to be an "analog" of the sound.

An analog signal has a theoretically infinite resolution. In practice an analog signal is subject to electronic noise and distortion introduced by communication channels and signal processing operations, which can progressively degrade the signal-to-noise ratio. In contrast, digital signals have a finite resolution. Converting an analog signal to digital form introduces a constant low-level noise called quantization noise into the signal which determines the noise floor, but once in digital form the signal can in general be processed or transmitted without introducing additional noise or distortion. Therefore as analog signal processing systems become more complex, they may ultimately degrade signal resolution to such an extent that their performance is surpassed by digital systems. This explains the widespread use of digital signals in preference to analog in modern technology. In analog systems, it is difficult to detect when such degradation occurs. However, in digital systems, degradation can not only be detected but corrected as well.

Disadvantages

The primary disadvantage of analog signal is that any system has noise – i.e., random unwanted variation. As the signal is copied and re-copied, or transmitted over long distances, or electronically processed, the unavoidable noise introduced by each step in the signal path is additive, progressively degrading the signal-to-noise ratio, until in extreme cases the signal can be overwhelmed. This is called generation loss. Noise can show up as 'hiss' and inter-modulation distortion in audio signals, or "snow" in video signals. This degradation is impossible to recover, since there is no sure way to distinguish the noise from the signal; amplifying the signal to recover attenuated parts of the signal amplifies the noise (distortion/interference) as well. Since digital signals can be transmitted, stored and processed without introducing noise, even if the resolution of an analog signal is higher than a comparable digital signal, after enough processing the analog signal to noise ratio will be lower.

Electrically, analog signal noise can be diminished by shielding, good connections, and several cable types such as coaxial or twisted pair.

Modulation

Another method of conveying an analog signal is to use modulation. In this, some base signal (e.g., a sinusoidal carrier signal) has one of its properties modulated: amplitude modulation involves altering the amplitude of a sinusoidal voltage waveform by the source information, frequency modulation changes the frequency. Other techniques, such as changing the phase of the base signal also work.

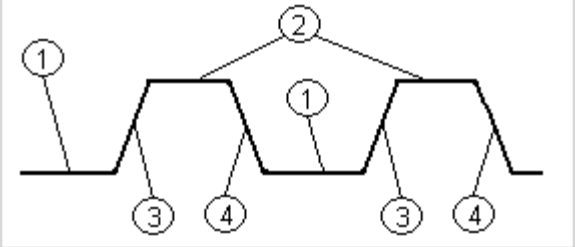
Analog circuits do not involve quantization of information into digital format. The concept being measured over the circuit, whether sound, light, pressure, temperature, or an exceeded limit, remains from end to end.

Digital signal

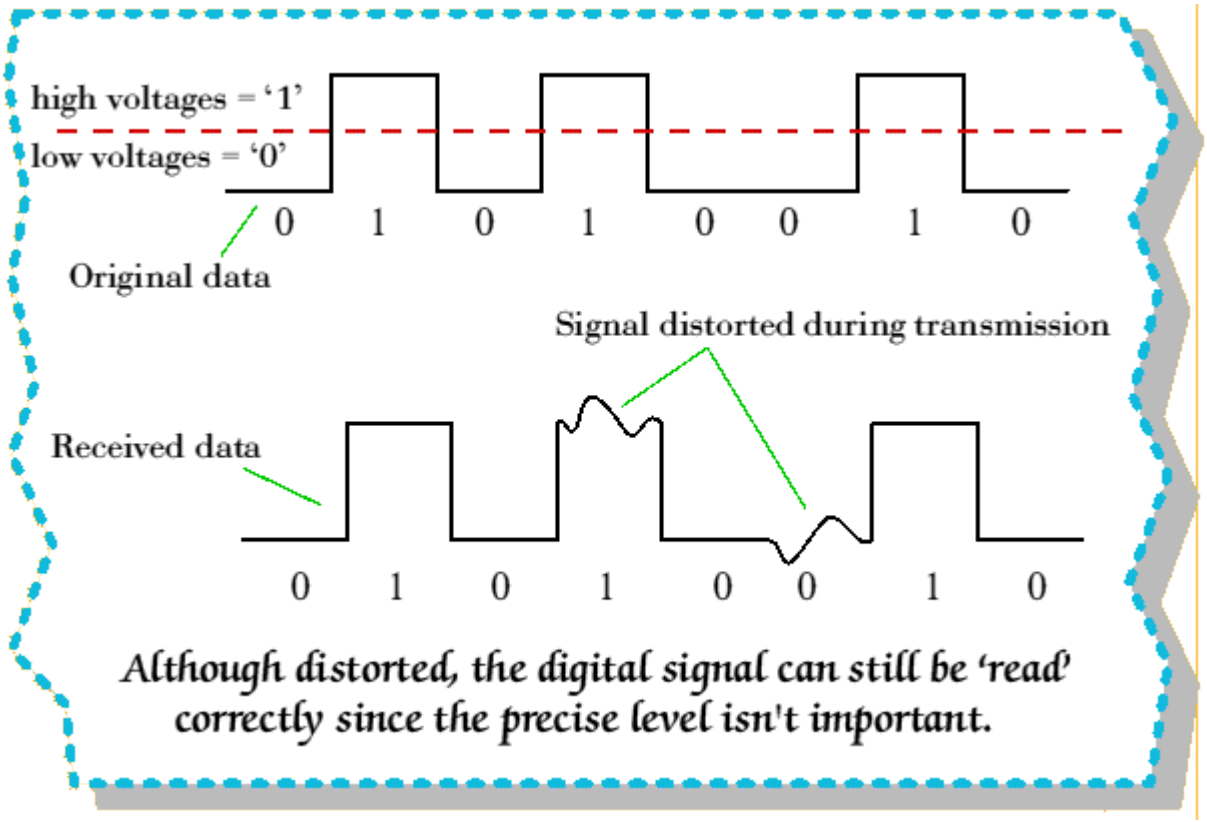
A **digital signal** is a physical signal that is a representation of a sequence of discrete values (a quantified discrete-time signal), for example of an arbitrary bit stream, or of a digitized (sampled and analog-to-digital converted) analog signal. *The term digital signal can refer to either of the following:*

1. Any continuous-time waveform signal used in digital communication, representing a bit stream or other sequence of discrete values
2. A pulse train signal that switches between a discrete number of voltage levels or levels of light intensity, also known as a line coded signal or baseband transmission, for example a signal found in digital electronics or in serial communications, or a pulse code modulation (PCM) representation of a digitized analog signal.

A signal that is generated by means of a digital modulation method (digital passband transmission), to be transferred between modems, is in the first case considered as a digital signal, and in the second case as converted to an analog signal.



A digital signal waveform: (1) low level, (2) high level, (3) rising edge, and (4) falling edge.



Waveforms in digital systems

In computer architecture and other digital systems, a waveform that switches between two voltage levels representing the two states of a Boolean value (0 and 1) is referred to as a *digital signal*, even though it is an analog voltage waveform, since it is interpreted in terms of only two levels.

The clock signal is a special digital signal that is used to synchronize digital circuits. The image shown can be considered the waveform of a clock signal. Logic changes are triggered either by the rising edge or the falling edge.

The given diagram is an example of the practical pulse and therefore we have introduced two new terms that are:

- **Rising edge:** the transition from a low voltage (level 1 in the diagram) to a high voltage (level 2)
- **Falling edge**: the transition from a high voltage to a low one.

Although in a highly simplified and idealized model of a digital circuit we may wish for these transitions to occur instantaneously, no real world circuit is purely resistive and therefore no circuit can instantly change voltage levels. This means that during a short, finite transition

time the output may not properly reflect the input, and indeed may not correspond to either a logically high or low voltage.

Analog vs. Digital

Analog and **digital** signals are used to transmit information, usually through electric signals. In both these technologies, the information, such as any audio or video, is transformed into electric signals. The **difference between analog and digital** technologies is that in analog technology, information is translated into electric pulses of varying amplitude. In digital technology, translation of information is into binary format (zero or one) where each bit is representative of two distinct amplitudes.

	Analog	**Digital**
Signal	Analog signal is a continuous signal which represents physical measurements.	Digital signals are discrete time signals generated by digital modulation.
Waves	Denoted by sine waves	Denoted by square waves
Representation	Uses continuous range of values to represent information	Uses discrete or discontinuous values to represent information
Example	Human voice in air, analog electronic devices.	Computers, CDs, DVDs, and other digital electronic devices.
Technology	Analog technology records waveforms as they are.	Samples analog waveforms into a limited set of numbers and records them.
Data transmissions	Subjected to deterioration by noise during transmission and write/read cycle.	Can be noise-immune without deterioration during transmission and write/read cycle.
Response to Noise	More likely to get affected reducing accuracy	Less affected since noise response are analog in nature
Flexibility	Analog hardware is not flexible.	Digital hardware is flexible in implementation.
Uses	Can be used in analog devices	Best suited for Computing and digital

	Analog	Digital
	only. Best suited for audio and video transmission.	electronics.
Applications	Thermometer	PCs, PDAs (Personal Data Assistants)
Bandwidth	Analog signal processing can be done in real time and consumes less bandwidth.	There is no guarantee that digital signal processing can be done in real time and consumes more bandwidth to carry out the same information.
Memory	Stored in the form of wave signal	Stored in the form of binary bit
Power	Analog instrument draws large power	Digital instrument drawS only negligible power
Cost	Low cost and portable	Cost is high and not easily portable
Impedance	Low	High order of 100 megaohm
Errors	Analog instruments usually have a scale which is cramped at lower end and give considerable observational errors.	Digital instruments are free from observational errors like parallax and approximation errors.

Properties of Digital vs Analog signals

Digital information has certain properties that distinguish it from analog communication methods. These include

- **Synchronization** – digital communication uses specific synchronization sequences for determining synchronization.
- **Language** – digital communications requires a language which should be possessed by both sender and receiver and should specify meaning of symbol sequences.
- **Errors** – disturbances in analog communication causes errors in actual intended communication but disturbances in digital communication does not cause errors enabling error free communication. Errors should be able to substitute, insert or delete symbols to be expressed.
- **Copying** – analog communication copies are quality wise not as good as their originals while due to error free digital communication, copies can be made indefinitely.
- **Granularity** – for a continuously variable analog value to be represented in digital form there occur quantization error which is difference in actual analog value and

digital representation and this property of digital communication is known as granularity.

Communications Protocol

In telecommunications, a **communications protocol** is a system of digital rules for data exchange within or between computers.

Communicating systems use well-defined formats for exchanging messages. Each message has an exact meaning intended to elicit a response from a range of possible responses pre-determined for that particular situation. Thus, a protocol must define the syntax, semantics, and synchronization of communication; the specified behavior is typically independent of how it is to be implemented. A protocol can therefore be implemented as hardware, software, or both. Communications protocols have to be agreed upon by the parties involved. To reach agreement a protocol may be developed into a technical standard.

Communicating Systems

The information exchanged between devices—through a network, or other media—is governed by rules and conventions that can be set out in technical specifications called communication protocol standards. The nature of a communication, the actual data exchanged and any state-dependent behaviors, is defined by its specification.

- In digital computing systems, the rules can be expressed by algorithms and data structures. Expressing the algorithms in a portable programming language makes the protocol software operating system independent.
- Operating systems usually consist of a set of cooperating processes that manipulate shared data to communicate with each other. This communication is governed by well-understood protocols, which can be embedded in the process code itself.
- In contrast, because there is no common memory, communicating systems have to communicate with each other using a shared transmission medium. Transmission is not necessarily reliable, and individual systems may use different hardware and/or operating systems.
- To implement a networking protocol, the protocol software modules are interfaced with a framework implemented on the machine's operating system.

This framework implements the networking functionality of the operating system. The best known frameworks are the TCP/IP model and the OSI model.

- At the time the Internet was developed, layering had proven to be a successful design approach for both compiler and operating system design and, given the similarities between programming languages and communication protocols, layering was applied to the protocols as well. This gave rise to the concept of layered protocols which nowadays forms the basis of protocol design.
- Systems typically do not use a single protocol to handle a transmission. Instead they use a set of cooperating protocols, sometimes called a protocol family or protocol suite. Some of the best known protocol suites include: IPX/SPX, X.25, AX.25, AppleTalk and TCP/IP.
- The protocols can be arranged based on functionality in groups, for instance there is a group of transport protocols. The functionalities are mapped onto the layers, each layer solving a distinct class of problems relating to, for instance: application-, transport-, internet- and network interface-functions. To transmit a message, a protocol has to be selected from each layer, so some sort of multiplexing/de-multiplexing takes place. The selection of the next protocol is accomplished by extending the message with a protocol selector for each layer.

Basic requirements of protocols

Messages are sent and received on communicating systems to establish communications. Protocols should therefore specify rules governing the transmission. *In general, much of the following should be addressed:*

- *Data formats for data exchange*. Digital message bitstrings are exchanged. The bitstrings are divided in fields and each field carries information relevant to the protocol. Conceptually the bitstring is divided into two parts called the *header area* and the *data area*. The actual message is stored in the data area, so the header area contains the fields with more relevance to the protocol. Bitstrings longer than the maximum transmission unit (MTU) are divided in pieces of appropriate size.
- *Address formats for data exchange*. Addresses are used to identify both the sender and the intended receiver(s). The addresses are stored in the header area of the bitstrings, allowing the receivers to determine whether the bitstrings are intended for themselves and

should be processed or should be ignored. A connection between a sender and a receiver can be identified using an address pair *(sender address, receiver address)*. Usually some address values have special meanings. An all-*1*s address could be taken to mean an addressing of all stations on the network, so sending to this address would result in a broadcast on the local network. The rules describing the meanings of the address value are collectively called an *addressing scheme*.

- ***Address mapping.*** Sometimes protocols need to map addresses of one scheme on addresses of another scheme. For instance to translate a logical IP address specified by the application to an Ethernet hardware address. This is referred to as *address mapping*.
- ***Routing.*** When systems are not directly connected, intermediary systems along the *route* to the intended receiver(s) need to forward messages on behalf of the sender. On the Internet, the networks are connected using routers. This way of connecting networks is called *internetworking*.
- ***Detection of transmission errors*** is necessary on networks which cannot guarantee error-free operation. In a common approach, CRCs of the data area are added to the end of packets, making it possible for the receiver to detect differences caused by errors. The receiver rejects the packets on CRC differences and arranges somehow for retransmission.
- ***Acknowledgements*** of correct reception of packets is required for connection-oriented communication. Acknowledgements are sent from receivers back to their respective senders.
- ***Loss of information - timeouts and retries.*** Packets may be lost on the network or suffer from long delays. To cope with this, under some protocols, a sender may expect an acknowledgement of correct reception from the receiver within a certain amount of time. On timeouts, the sender must assume the packet was not received and retransmit it. In case of a permanently broken link, the retransmission has no effect so the number of retransmissions is limited. Exceeding the retry limit is considered an error.
- ***Direction of information flow*** needs to be addressed if transmissions can only occur in one direction at a time as on half-duplex links. This is known as Media Access Control. Arrangements have to be made to accommodate the case when two parties want to gain control at the same time.
- ***Sequence control.*** We have seen that long bitstrings are divided in pieces, and then sent on the network individually. The pieces may get lost or delayed or take different routes to

their destination on some types of networks. As a result pieces may arrive out of sequence. Retransmissions can result in duplicate pieces. By marking the pieces with sequence information at the sender, the receiver can determine what was lost or duplicated, ask for necessary retransmissions and reassemble the original message.

- ***Flow control*** is needed when the sender transmits faster than the receiver or intermediate network equipment can process the transmissions. Flow control can be implemented by messaging from receiver to sender.

Getting the data across a network is only part of the problem for a protocol. The data received has to be evaluated in the context of the progress of the conversation, so a protocol has to specify rules describing the context. These kinds of rules are said to express the *syntax* of the communications. Other rules determine whether the data is meaningful for the context in which the exchange takes place. These kind of rules are said to express the *semantics* of the communications.

Switching

- Messages Switching
- Circuit Switching
- Packet Switching

Packet switching

- **Packet switching** is a digital networking communications method that groups all transmitted data – regardless of content, type, or structure – into suitably sized blocks, called *packets*.
- Packet switching features delivery of variable bit-rate data streams (sequences of packets) over a shared network which allocates transmission resources as needed using statistical multiplexing or dynamic bandwidth allocation techniques. When traversing network adapters, switches, routers, and other network nodes, packets are buffered and queued, resulting in variable delay and throughput depending on the network's capacity and the traffic load on the network.
- Packet switching contrasts with another principal networking paradigm, circuit switching, a method which sets up a limited number of dedicated connections of constant bit rate and constant delay between nodes for exclusive use during the

communication session. In cases where traffic fees are charged (as opposed to flat rate), for example in cellular communication services, circuit switching is characterized by a fee per unit of connection time, even when no data is transferred, while packet switching is characterized by a fee per unit of information transmitted (characters, packets, messages,).

- Packet mode communication may be utilized with or without intermediate forwarding nodes (packet switches or routers). Packets are normally forwarded by intermediate network nodes asynchronously using first-in, first-out buffering, but may be forwarded according to some scheduling discipline for fair queuing, traffic shaping, or for differentiated or guaranteed quality of service, such as weighted fair queuing or leaky bucket. In case of a shared physical medium (radio, 10BASE5 or thick Ethernet,), the packets may be delivered according to a multiple access scheme.

Connectionless and connection-oriented packet switching

Two major packet switching modes exist:

1. Connectionless packet switching, also known as datagram switching; and
2. Connection-oriented packet switching, also known as virtual circuit switching.

Connection-oriented networks

In connection-oriented networks, each packet is labeled with a connection ID rather than an address. Address information is only transferred to each node during a connection set-up phase, when the route to the destination is discovered and an entry is added to the switching table in each network node through which the connection passes. The signaling protocols used allow the application to specify its requirements and the network to specify what capacity etc. is available, and acceptable values for service parameters to be negotiated. Routing a packet is very simple, as it just requires the node to look up the ID in the table. The packet header can be small, as it only needs to contain the ID and any information (such as length, timestamp, or sequence number) which is different for different packets.

Connection oriented packet-switching protocols include X.25, Frame relay, Multiprotocol Label Switching (MPLS), and TCP.

Connectionless networks

In connectionless networks, each packet is labeled with a destination address, source address, and port numbers; it may also be labeled with the sequence number of the packet. This precludes the need for a dedicated path to help the packet find its way to its destination, but means that much more information is needed in the packet header, which is therefore larger, and this information needs to be looked up in power-hungry content-addressable memory. Each packet is dispatched and may go via different routes; potentially, the system has to do as much work for every packet as the connection-oriented system has to do in connection set-up, but with less information as to the application's requirements. At the destination, the original message/data is reassembled in the correct order, based on the packet sequence number. Thus a virtual connection, also known as a virtual circuit or byte stream is provided to the end-user by a transport layer protocol, although intermediate network nodes only provides a connectionless network layer service.

Each packet includes complete addressing or routing information. The packets are routed individually, sometimes resulting in different paths and out-of-order delivery. a connection is defined and pre-allocated in each involved node during a connection setup phase before any packet is transferred. The packets include a connection identifier rather than address information and are delivered in order.

Some connectionless protocols are Ethernet, IP, and UDP (**User Datagram Protocol**).

Circuit switching

Circuit switching is a methodology of implementing a telecommunications network in which two network nodes establish a dedicated communications channel (circuit) through the network before the nodes may communicate. The circuit guarantees the full bandwidth of the channel and remains connected for the duration of the communication session. The circuit functions as if the nodes were physically connected as with an electrical circuit.

The defining example of a circuit-switched network is the early analog telephone network. When a call is made from one telephone to another, switches within the telephone exchanges create a continuous wire circuit between the two telephones, for as long as the call lasts.

Circuit switching contrasts with packet switching which divides the data to be transmitted into packets transmitted through the network independently. In packet switching, instead of being dedicated to one communication session at a time, network links are shared by packets from

multiple competing communication sessions, resulting in the loss of the quality of service guarantees that are provided by circuit switching.

In circuit switching, the bit delay is constant during a connection, as opposed to packet switching, where packet queues may cause varying and potentially indefinitely long packet transfer delays. No circuit can be degraded by competing users because it is protected from use by other callers until the circuit is released and a new connection is set up. Even if no actual communication is taking place, the channel remains reserved and protected from competing users.

Virtual circuit switching is a packet switching technology that emulates circuit switching, in the sense that the connection is established before any packets are transferred, and packets are delivered in order.

While circuit switching is commonly used for connecting voice circuits, the concept of a dedicated path persisting between two communicating parties or nodes can be extended to signal content other than voice. Its advantage is that it provides for continuous transfer without the overhead associated with packets making maximal use of available bandwidth for that communication. Its disadvantage is that it can be relatively inefficient because unused capacity guaranteed to a connection cannot be used by other connections on the same network.

Compared to datagram packet switching

Circuit switching contrasts with packet switching which divides the data to be transmitted into small units, called packets, transmitted through the network independently. Packet switching shares available network bandwidth between multiple communication sessions.

Multiplexing multiple telecommunications connections over the same physical conductor has been possible for a long time, but nonetheless each channel on the multiplexed link was either dedicated to one call at a time, or it was idle between calls.

In circuit switching, and virtual circuit switching, a route and bandwidth is reserved from source to destination. Circuit switching can be relatively inefficient because capacity is guaranteed on connections which are set up but are not in continuous use, but rather momentarily. However, the connection is immediately available while established.

Packet switching is the process of segmenting a message/data to be transmitted into several smaller packets. Each packet is labeled with its destination and a sequence number for ordering related packets, precluding the need for a dedicated path to help the packet find its way to its

destination. Each packet is dispatched independently and each may be routed via a different path. At the destination, the original message is reassembled in the correct order, based on the packet number. Datagram packet switching networks do not require a circuit to be established and allow many pairs of nodes to communicate concurrently over the same channel.

Message switching

A Message is a logical unit of information and can be of any length. In this method, if a station or a switching office wishes to send a message to another, it first attaches the destination address to message. When the sender has a block of data to be sent, it is stored in the first switching office and then forwarded later. This method is known as store-and-forward. Each block is received entirely, checked for errors and then retransmitted.

In message switching, no physical connection is required between the source and destination.

But one disadvantage is that the message length is unlimited i.e. each switching node must have sufficient storage to buffer message and another one is that a message is delay at each node because of time required to receive the message plus a queuing delay waiting for a chance to retransmit message to next node.

Network Architecture

Network architecture is the design of a communications network. It is a framework for the specification of a network's physical components and their functional organization and configuration, its operational principles and procedures, as well as data formats used in its operation.

In telecommunication, the specification of a network architecture may also include a detailed description of products and services delivered via a communications network, as well as detailed rate and billing structures under which services are compensated.

The network architecture of the Internet is predominantly expressed by its use of the Internet Protocol Suite, rather than a specific model for interconnecting networks or nodes in the network, or the usage of specific types of hardware links.

OSI Network Model

The Open Systems Interconnection model (OSI model) is a product of the Open Systems Interconnection effort at the International Organization for Standardization. It is a way of sub-dividing a communications system into smaller parts called layers. A layer is a collection of similar functions that provide services to the layer above it and receives services from the layer below it. On each layer, an instance provides services to the instances at the layer above and requests service from the layer below.

Physical (Layer 1)

This layer conveys the bit stream - electrical impulse, light or radio signal -- through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier, including defining cables, cards and physical aspects. Fast Ethernet, RS232, and ATM are protocols with physical layer components.

Layer 1 Physical examples include Ethernet, FDDI, B8ZS, V.35, V.24, RJ45.

Data Link (Layer 2)

At this layer, data packets are encoded and decoded into bits. It furnishes transmission protocol knowledge and management and handles errors in the physical layer, flow control and frame synchronization. The data link layer is divided into two sub layers: The Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. The MAC sub layer controls how a computer on the network gains access to the data and permission to transmit it. The LLC layer controls frame synchronization, flow control and error checking.

Layer 2 Data Link examples include PPP, FDDI, ATM, IEEE 802.5/ 802.2, IEEE 802.3/802.2, HDLC, Frame Relay.

Network (Layer 3)

This layer provides switching and routing technologies, creating logical paths, known as virtual circuits, for transmitting data from node to node. Routing and forwarding are functions of this layer, as well as addressing, internetworking, error handling, congestion control and packet sequencing.

Layer 3 Network examples include AppleTalk DDP, IP, IPX.

Transport (Layer 4)

This layer provides transparent transfer of data between end systems, or hosts, and is responsible for end-to-end error recovery and flow control. It ensures complete data transfer.

Layer 4 Transport examples include SPX, TCP, UDP.

Session (Layer 5)

This layer establishes, manages and terminates connections between applications. The session layer sets up, coordinates, and terminates conversations, exchanges, and dialogues between the applications at each end. It deals with session and connection coordination.

Layer 5 Session examples include NFS, NetBios names, RPC, SQL.

Presentation (Layer 6)

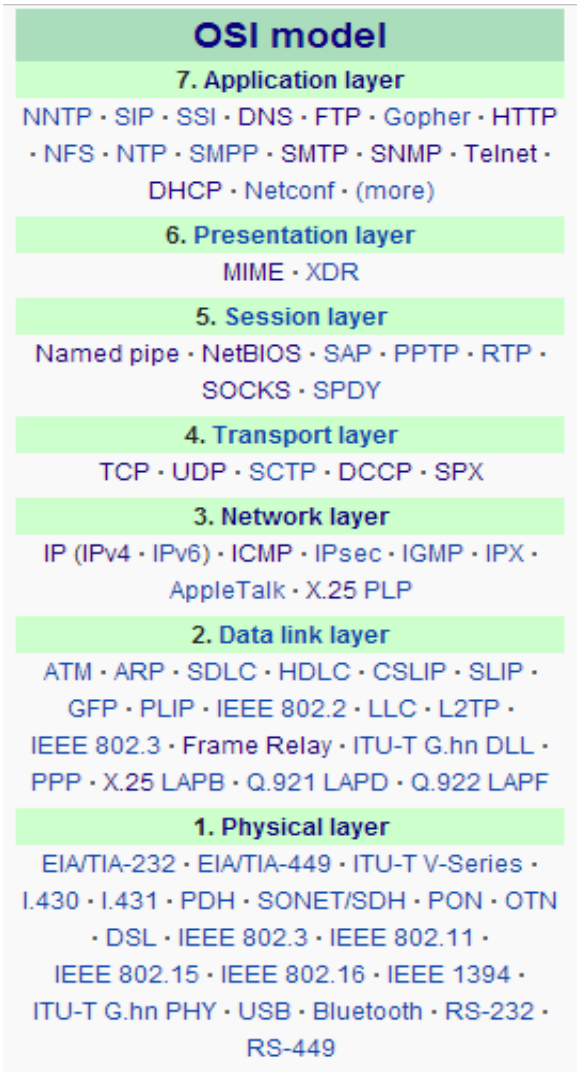
This layer provides independence from differences in data representation (e.g., encryption) by translating from application to network format, and vice versa. The presentation layer works to transform data into the form that the application layer can accept. This layer formats and encrypts data to be sent across a network, providing freedom from compatibility problems. It is sometimes called the syntax layer.

Layer 6 Presentation examples include encryption, ASCII, EBCDIC, TIFF, GIF, PICT, JPEG, MPEG, MIDI.

Application (Layer 7)

This layer supports application and end-user processes. Communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. Everything at this layer is application-specific. This layer provides application services for file transfers, e-mail, and
other network software services. Telnet and FTP are applications that exist entirely in the application level. Tiered application architectures are part of this layer**.**

Layer 7 Application examples include WWW browsers, NFS, SNMP, Telnet, HTTP, FTP



Distributed computing

In distinct usage in distributed computing, the term *network architecture* often describes the structure and classification of distributed application architecture, as the participating nodes in a distributed application are often referred to as a *network*. For example, the applications architecture of the public switched telephone network (PSTN) has been termed the Advanced Intelligent Network. There are any number of specific classifications but all lie on a continuum between the dumb network (e.g., Internet) and the intelligent computer network (e.g., the telephone network). Other networks contain various elements of these two classical types to make them suitable for various types of applications. Recently the context aware network, which is a synthesis of two, has gained much interest with its ability to combine the best elements of both.

A popular example of such usage of the term in distributed applications, as well as PVCs (permanent virtual circuits), is the organization of nodes in peer-to-peer (P2P) services and

networks. P2P networks usually implement overlay networks running over an underlying physical or logical network. These overlay network may implement certain organizational structures of the nodes according to several distinct models, the network architecture of the system.

Network architecture is a broad plan that specifies everything necessary for two application programs on different networks on an Internet to be able to work together effectively.