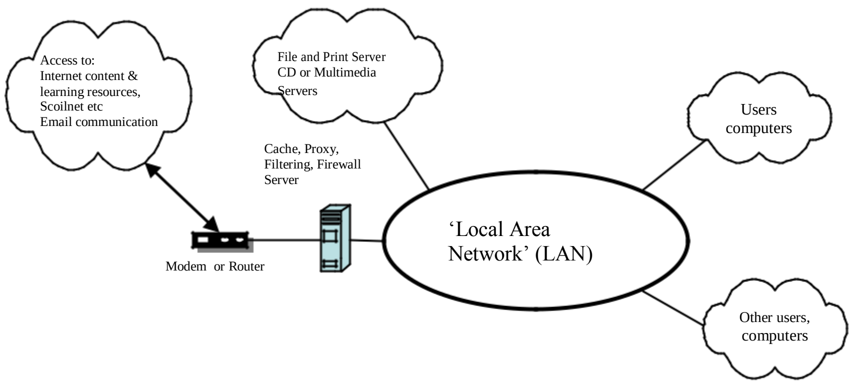
INTRODUCTION TO COMPUTER NETWORKS

A computer network consists of a collection of computers, peripherals and other equipment that are connected together so that they can communicate with each other on the network. The figure below gives an example of a network in a school comprising of a local area network or LAN connecting computers with each other, the internet, and various servers.



Peer-to-peer networks are more commonly implemented where less than ten computers are involved and where strict security is not necessary. All computers have the same status, hence the term **'peer'**, and they communicate with each other on an equal footing. Files, such as word processing or spreadsheet documents, can be shared across the network and all the computers on the network can share devices, such as printers or scanners, which are connected to any one computer.

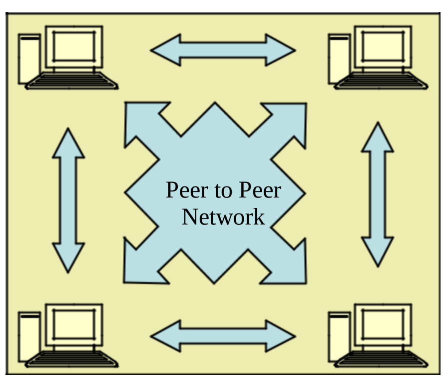


Fig 2: Peer to Peer Networking

Client/server networks are more suitable for larger networks. A central computer, or 'server', acts as the storage location for files and applications shared on the network. Usually the server is a higher than average performance computer. The server also controls the network access of the other computers which are referred to as the 'client' computers. Typically, teachers and students in a school will use the client computers for their work and only the network administrator (usually a designated staff member) will have access rights to the server.

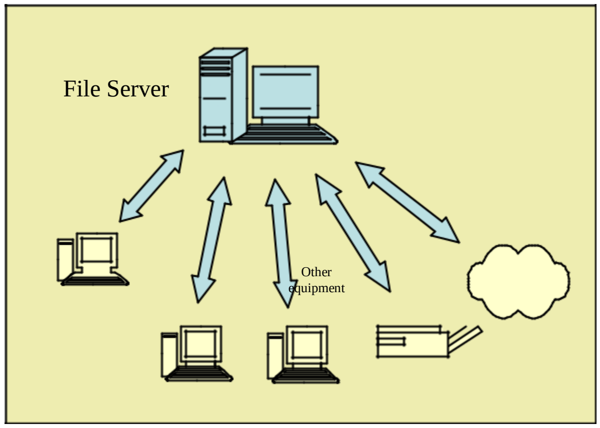


Fig 3: Client - Server Networking

The table below provides a summary comparison between Peer-to-Peer and Client/Server Networks.

Peer-to-Peer Networks vs Client/Server Networks	
Peer-to-Peer Networks	**Client/Server Networks**
Easy to set up	More difficult to set up
Less expensive to install	More expensive to install
Can be implemented on a wide range of operating systems	A variety of operating systems can be supported on the client computers, but the server needs to run an operating system that supports networking
More time consuming to maintain the software being used (as computers must be managed individually)	Less time consuming to maintain the software being used (as most of the maintenance is managed from the server)
Very low levels of security supported or none at all. These can be very cumbersome to set up, depending on the operating system being used	High levels of security are supported, all of which are controlled from the server. Such measures prevent the deletion of essential system files or the changing of settings
Ideal for networks with less than 10 computers	No limit to the number of computers that can be supported by the network
Does not require a server	Requires a server running a server operating system
Demands a moderate level of skill to administer the network	Demands that the network administrator has a high level of IT skills with a good working knowledge of a server operating system

Basic Components IN A Network

There are many components that can be part of a network, for example personal computers, servers, networking devices, and cabling. These components can be grouped into four main categories:

- Hosts
- Shared peripherals
- Networking devices
- Networking media

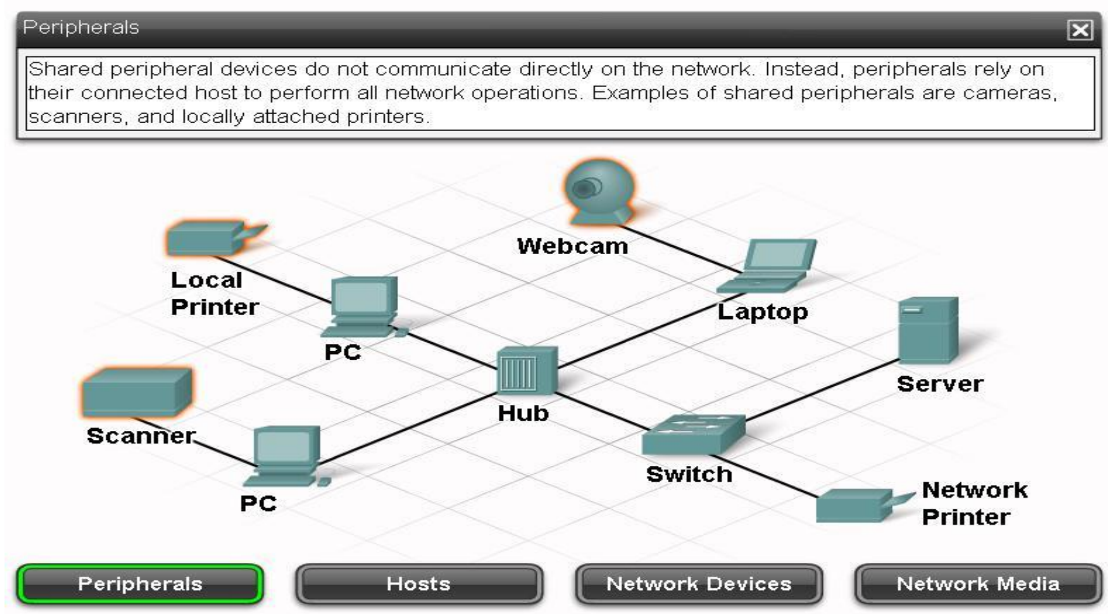
The network components that people are most familiar with are hosts and shared peripherals. Hosts are devices that send and receive messages directly across the network.

Shared peripherals are not directly connected to the network, but instead are connected to hosts. The host is then responsible for sharing the peripheral across the network. Hosts have computer software configured to enable people on the network to use the attached peripheral devices.

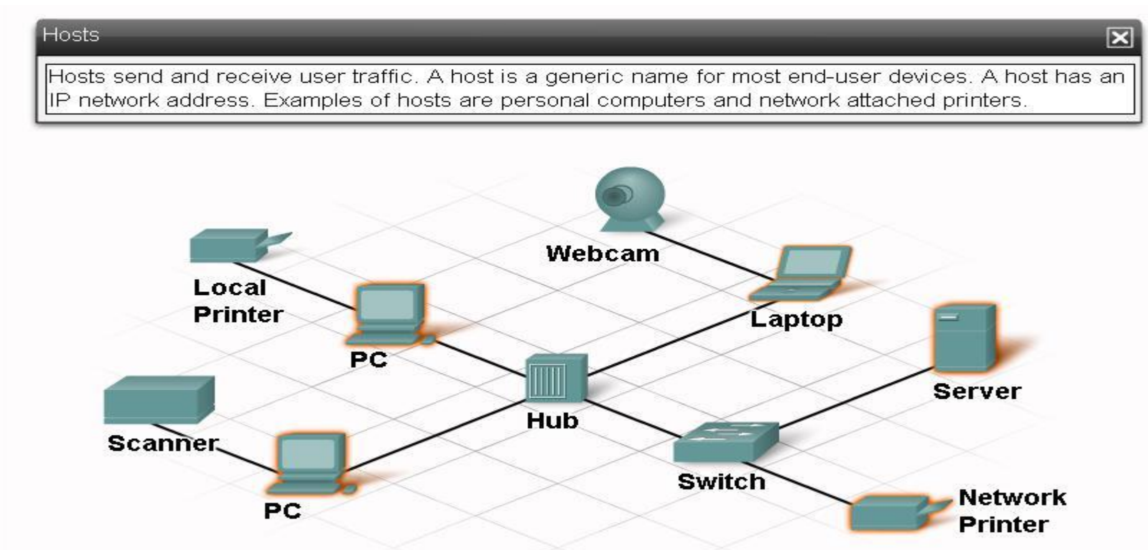
The network devices, as well as networking media, are used to interconnect hosts.

Some devices can play more than one role, depending on how they are connected. For example, a printer directly connected to a host (local printer) is a peripheral. A printer directly connected to a network device and participates directly in network communications is a host.

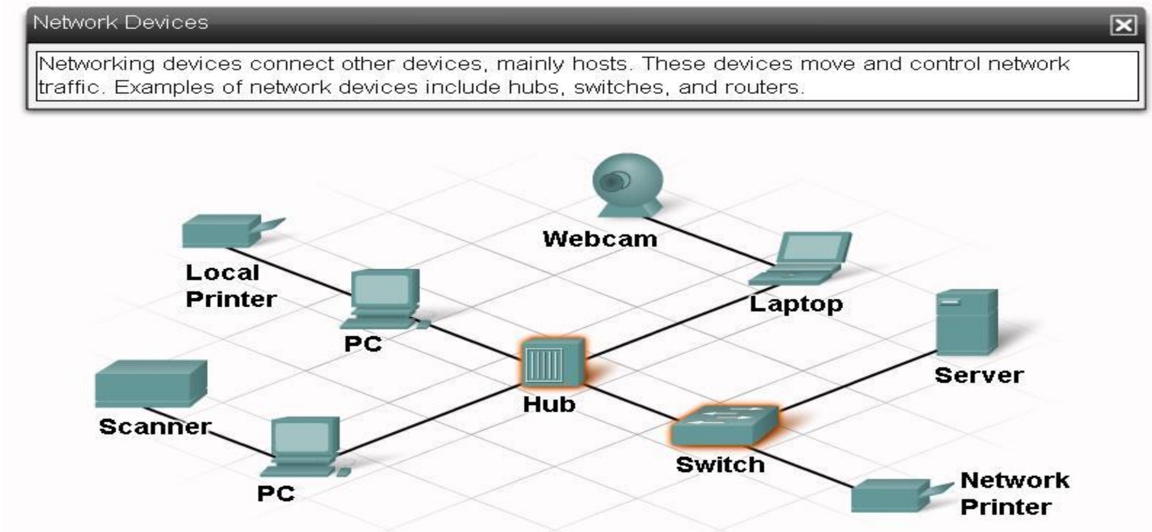
1. PERIPHERALS



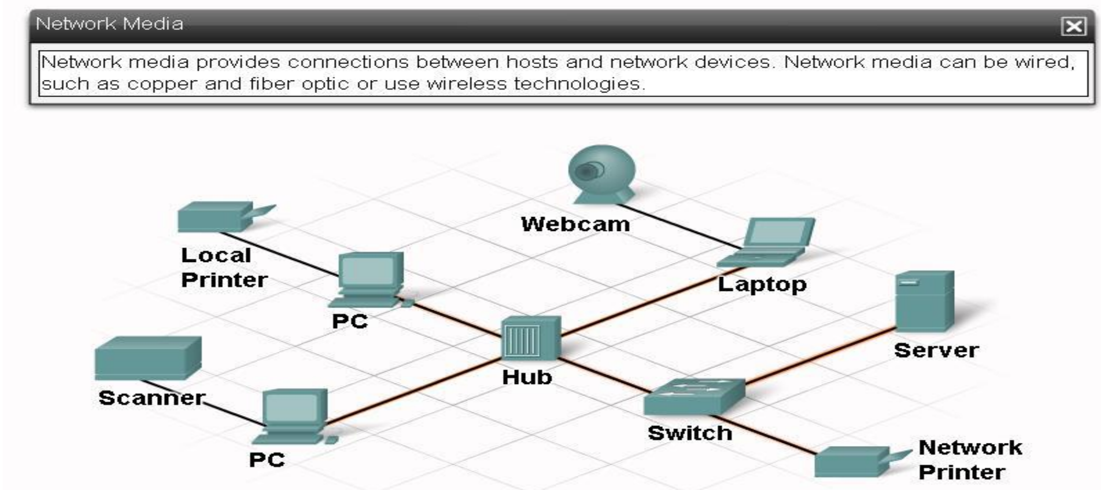
2. HOSTS



3. NETWORK DEVICES



4. NETWORK MEDIA



OSI REFERENCE MODEL

The Open Systems Interconnection reference model is a layered, abstract representation created as a guideline for network protocol design. The OSI model divides the networking process into seven logical layers, each of which has unique functionality and to which are assigned specific services and protocols.

In this model, information is passed from one layer to the next, starting at the Application layer on the transmitting host, and proceeding down the hierarchy to the Physical layer, then passing over the communications channel to the destination host, where the information proceeds back up the hierarchy, ending at the Application layer. The figure depicts the steps in this process.

The Application layer, Layer seven, is the top layer of both the OSI and TCP/IP models. It is the layer that provides the interface between the applications we use to communicate and the underlying network over which our messages are transmitted. Application layer protocols are used to exchange data between programs running on the source and destination hosts. There are many Application layer protocols and new protocols are always being developed.

Initially the OSI model was designed by the International Organization for Standardization (ISO) to provide a framework on which to build a suite of open systems protocols. The vision was that this set of protocols would be used to develop an international network that would not be dependent on proprietary systems.

As a reference model, the OSI model provides an extensive list of functions and services that can occur at each layer. It also describes the interaction of each layer with the layers directly above and below it.

1. Physical Layer

The OSI Physical layer provides the means to transport across the network media the bits that make up a Data Link layer frame. This layer accepts a complete frame from the Data Link layer and encodes it as a series of signals that are transmitted onto the local media. The encoded bits that comprise a frame are received by either an end device or an intermediate device.

The delivery of frames across the local media requires the following Physical layer elements:

- The physical media and associated connectors
- A representation of bits on the media
- Encoding of data and control information
- Transmitter and receiver circuitry on the network devices

At this stage of the communication process, the user data has been segmented by the Transport layer, placed into packets by the Network layer, and further encapsulated as frames by the Data Link layer. The purpose of the Physical layer is to create the electrical, optical, or microwave signal that represents the bits in each frame. These signals are then sent on the media one at a time.

It is also the job of the Physical layer to retrieve these individual signals from the media, restore them to their bit representations, and pass the bits up to the Data Link layer as a complete frame.

Protocols of Physical Layer

Commonly used protocol here is the Ethernet such as 10BaseT or 100BaseTX which specifies the type of cables that can be used, the optimal topology (star vs. bus, etc.), the maximum length of cables, etc.

2. The Data link Layer

Transmits frames of data from computer to computer on the same network segment. Ensures the reliability of the physical link established at layer 1. Standards define how data frames are recognized and provide the necessary flow control and error handling at the frame set.

The data link layer is divided into two sublayers: **The Media Access Control (MAC)** layer and the Logical Link Control (LLC) layer. The **MAC sublayer** 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.

Protocols in Layer 2

Actual Layer 2 protocol used depends on the logical topology of the network and the implementation of the Physical layer. Given the wide range of physical media used across the range of topologies in networking, there are a correspondingly high number of Layer 2 protocols in use and they include:-

- Ethernet
- Point-to-Point Protocol (PPP)
- High-Level Data Link Control (HDLC)
- Frame Relay
- Asynchronous Transfer Mode (ATM)

Each protocol performs media access control for specified Layer 2 logical topologies. This means that a number of different network devices can act as nodes that operate at the Data Link layer when implementing these protocols. These devices include the network adapter or network interface cards (NICs) on computers as well as the interfaces on routers and Layer 2 switches.

The Layer 2 protocol used for a particular network topology is determined by the technology used to implement that topology. The technology is, in turn, determined by the size of the network - in terms of the number of hosts and the geographic scope - and the services to be provided over the network.

3. The Network Layer

The Network layer, or OSI Layer 3, provides services to exchange the individual pieces of data over the network between identified end devices. To accomplish this end-to-end transport, Layer 3 uses four basic processes:

- **Addressing**

First, the Network layer must provide a mechanism for addressing these end devices. If individual pieces of data are to be directed to an end device, that device must have a unique address. In an IPv4 network, when this address is added to a device, the device is then referred to as a host.

- **Encapsulation *(Wrapping of data in a particular protocol header)***

Second, the Network layer must provide encapsulation. Not only must the devices be identified with an address, the individual pieces - the Network layer PDUs (Protocol Data Unit) - must also contain these addresses. During the encapsulation process, Layer 3 receives the Layer 4 PDU and adds a Layer 3 header, or label, to create the Layer 3 PDU. When referring to the Network layer, we call this PDU a packet. When a packet is created, the header must contain the address of the host to which it is being sent. This address is referred to as the destination address. The Layer 3 header also contains the address of the originating host. This address is called the source address.

After the Network layer completes its encapsulation process, the packet is sent down to the Data Link layer to be prepared for transportation over the media.

- **Routing**

Next, the Network layer must provide services to direct these packets to their destination host. The source and destination hosts are not always connected to the same network. In fact, the packet might have to travel through many different networks. Along the way, each packet must be guided through the network to reach its final destination and these networks are connected by routers. The role of the router is to select paths for and direct packets toward their destination. This process is known as routing.

During the routing through an internetwork, the packet may traverse many intermediary devices. Each route that a packet takes to reach the next device is called a hop. As the packet is forwarded, its contents (the Transport layer PDU), remain intact until the destination host is reached.

- **Decapsulation**

Finally, the packet arrives at the destination host and is processed at Layer 3. The host examines the destination address to verify that the packet was addressed to this device. If the address is correct, the packet is decapsulated by the Network layer and the Layer 4 PDU contained in the packet is passed up to the appropriate service at Transport layer.

Network Layer Protocols

Protocols implemented at the Network layer that carry user data include:

- Internet Protocol version 4 (IPv4)
- Internet Protocol version 6 (IPv6)
- Novell Internetwork Packet Exchange (IPX)
- AppleTalk
- Connectionless Network Service (CLNS/DECNet)

The Internet Protocol (IPv4 and IPv6) is the most widely-used Layer 3 data carrying protocol and will be the focus of this course. Discussion of the other protocols will be minimal.

4. The Transport Layer

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. Sequences data packets, and requests retransmission of missing packets. It also repackages messages for more efficient transmission over the network.

The Transport layer provides for the segmentation of data and the control necessary to reassemble these pieces into the various communication streams. Its primary responsibilities to accomplish this are:

- **Tracking Individual Conversations**

Any host may have multiple applications that are communicating across the network. Each of these applications will be communicating with one or more applications on remote hosts. It is the responsibility of the Transport layer to maintain the multiple communication streams between these applications.

- **Segmenting Data**

As each application creates a stream data to be sent to a remote application, this data must be prepared to be sent across the media in manageable pieces. The Transport layer protocols describe services that segment this data from the Application layer. This includes the encapsulation required on each piece of data. Each piece of application data requires headers to be added at the Transport layer to indicate to which communication it is associated.

- **Reassembling Segments**

At the receiving host, each piece of data may be directed to the appropriate application. Additionally, these individual pieces of data must also be reconstructed into a complete data stream that is useful to the Application layer. The protocols at the Transport layer describe the how the Transport layer header information is used to reassemble the data pieces into streams to be passed to the Application layer.

- **Identifying the Applications**

In order to pass data streams to the proper applications, the Transport layer must identify the target application. To accomplish this, the Transport layer assigns an application an identifier. The TCP/IP protocols call this identifier a port number. Each software process that needs to access the network is assigned a port number unique in that host. This port number is used in the transport layer header to indicate to which application that piece of data is associated.

The two most common Transport layer protocols of TCP/IP protocol suite are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). Both protocols manage the communication of multiple applications. The differences between the two are the specific functions that each protocol implements.

1. **User Datagram Protocol (UDP)**

UDP is a simple, connectionless protocol, described in RFC 768. It has the advantage of providing for low overhead data delivery. The pieces of communication in UDP are called datagrams. These datagrams are sent as "best effort" by this Transport layer protocol.

Applications that use UDP include:

- Domain Name System (DNS)
- Video Streaming
- Voice over IP (VoIP)

2. **Transmission Control Protocol (TCP)**

TCP is a connection-oriented protocol, described in RFC 793. TCP incurs additional overhead to gain functions. Additional functions specified by TCP are the same order delivery, reliable delivery, and flow control. Each TCP segment has 20 bytes of overhead in the header encapsulating the Application layer data, whereas each UDP segment only has 8 bytes of overhead. See the figure for a comparison.

Applications that use TCP are:

- Web Browsers
- E-mail
- File Transfers

5. The Session Layer

As the name of the Session layer implies, functions at this layer is to establish, manage, and terminate sessions between applications. i.e. creates and maintains dialogs between source and destination applications.

The Session layer handles the exchange of information to initiate dialogs, keep them active, and to restart sessions that are disrupted or idle for a long period of time. Opens manages, and closes conversations between two computers. It performs name recognition and the functions such as security, needed to allow two applications to communicate over the network, also provides error handling.

Some of the communication tasks performed at this layer:

- Establishing connections
- Maintaining connections
- Synchronizing communications
- Controlling dialogues
- Terminating connections

When you create a connection, you authenticate the user account at the sending and receiving computers. Connection creation also involves determining the type of communication that will take place and the protocols that will be used by the lower layers.

Data transfer and dialogue control are used to determine which computer is making requests and which computer is making responses. This also determines whether acknowledgments are required for data transmission.

A ***session*** is a series of related connection-oriented transmissions between network nodes. Another way to look at it is that a session is the interrelated communications between two or more presentation entities, which emphasizes that the Session layer provides services to the Presentation layer.

The three basic transmission modes are:

- Simplex Communications: only allows data to flow in one direction.
- Half-duplex Communications: Two way data flow, only one way at a time.
- Full-duplex Communications: Two way data flow simultaneously.

Requirements to know about full-duplex operations:

- Requires full-duplex NIC cards
- Loopback and collision detection must be disabled in the NIC card
- The NIC card's device driver must support simultaneous transmission and receiving.
- Full-duplex circuits are capable of 10 Mbps, 100 Mbps, and Gigabit Ethernet data speeds.

The following services and protocols are defined on the Sessions layer:

- ASP (AppleTalk Session Protocol)
- NFS (Network File Services)

- RPC (Remote Procedure Call)
- SCP (Serial Communications Protocol)
- SQL (Structured Query Language)
- X Window System and X Terminal
- ZIP (AppleTalk Zone Information Protocol)

6. The Presentation Layer

i) Coding and conversion of Application layer data to ensure that data from the source device can be interpreted by the appropriate application on the destination device.
ii) Compression of the data in a manner that can be decompressed by the destination device.
iii) Encryption of the data for transmission and the decryption of data upon receipt by the destination.

Protocols of the Presentation Layer
Presentation layer implementations are not typically associated with a particular protocol stack. The standards for video and graphics are examples. Some well-known standards for video include QuickTime and Motion Picture Experts Group (MPEG). QuickTime is an Apple Computer specification for video and audio, and MPEG is a standard for video compression and coding.

Among the well-known graphic image formats are Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), and Tagged Image File Format (TIFF). GIF and JPEG are compression and coding standards for graphic images, and TIFF is a standard coding format for graphic images.

7. Application Layer

The Application layer is responsible for directly accessing the underlying processes that manage and deliver communication to the human network. This layer serves as the source and destination of communications across data networks.

The Application layer applications, protocols, and services enable users to interact with the data network in a way that is meaningful and effective.

Applications are computer programs with which the user interacts and which initiate the data transfer process at the user's request.

Services are background programs that provide the connection between the Application layer and the lower layers of the networking model.

Protocols provide a structure of agreed-upon rules and processes that ensure services running on one particular device can send and receive data from a range of different network devices.

All the three components may be used by a single executable program and may even use the same name.
For example, when discussing "Telnet" we could be referring to the application, the service, or the protocol.

Application layer relies on the functions of the lower layers in order to complete the communication process. Within the Application layer, protocols specify what messages are exchanged between the source and destination hosts, the syntax of the control commands, the type and format of the data being transmitted, and the appropriate methods for error notification and recovery.

Delivery of data over the network can be requested from a server by a client, or between devices that operate in a peer-to-peer arrangement, where the client/server relationship is established according to which device is the source and destination at that time. Messages are exchanged between the Application layer services at each end device in accordance with the protocol specifications to establish and use these relationships.

Protocols in the Application Layer

The most widely-known Application layer protocols are those that provide for the exchange of user information. These protocols specify the format and control information necessary for many of the common Internet communication functions. Among these TCP/IP protocols are:

- Domain Name Service Protocol (DNS) is used to resolve Internet names to IP addresses.
- Hypertext Transfer Protocol (HTTP) is used to transfer files that make up the Web pages of the World Wide Web.
- Simple Mail Transfer Protocol (SMTP) is used for the transfer of mail messages and attachments.
- Telnet, a terminal emulation protocol, is used to provide remote access to servers and networking devices.
- File Transfer Protocol (FTP) is used for interactive file transfer between systems.
- SMB (Server Message Block) enables users to share files.