
Chapter Two: Networks Models

THE OSI MODEL

Established in 1947, the *International Standards Organization (ISO)* is a multinational body dedicated to worldwide agreement on international standards. An ISO standard that covers all aspects of network communications is the *Open Systems Interconnection model*. It was first introduced in the late 1970s. An *open system* is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture. The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software. The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable. *ISO is the organization. OSI is the model.*

The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network (see Figure (2.1)).

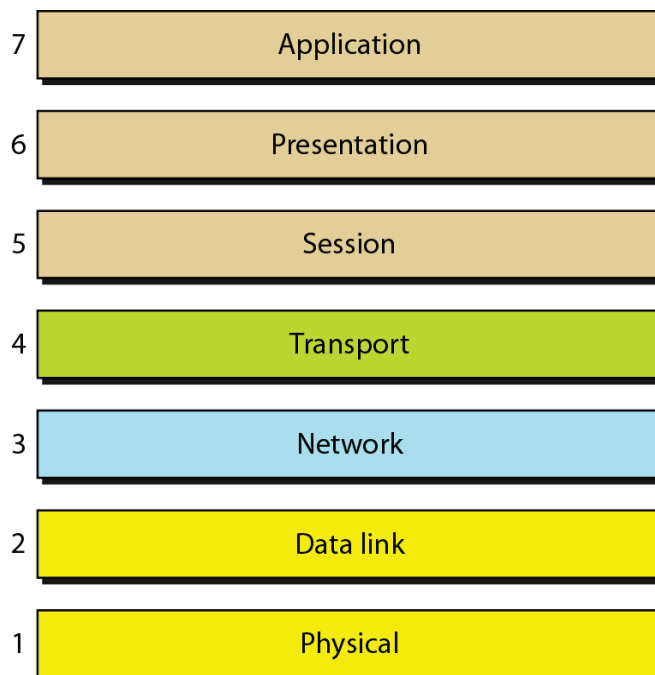


Fig. (2.1): Seven layers of the OSI model.

Layered Architecture

The OSI model is composed of seven ordered layers: physical (layer 1), data link (layer 2), network (layer 3), transport (layer 4), session (layer 5), presentation (layer 6), and application (layer 7). Figure (2.2) shows the layers involved when a message is sent from device A to device B. As the message travels from A to B, it may pass through many intermediate nodes. These intermediate nodes usually involve only the first three layers of the OSI model.

Within a single machine, each layer calls upon the services of the layer just below it. Layer 3, for example, uses the services provided by layer 2 and provides services for layer 4. Between machines, layer *X* on one machine communicates with layer *X* on another machine. This communication is governed by an agreed-upon series of rules and conventions called protocols. The processes on each machine that communicate at a given layer are called peer-to-peer processes. Communication between machines is therefore a peer-to-peer process using the protocols appropriate to a given layer.

Peer-to-Peer Processes

At the physical layer, communication is direct: In Figure 2.3, device A sends a stream of bits to device B (through intermediate nodes). At the higher layers, however, communication must move down through the layers on device A, over to device B, and then back up through the layers. Each layer in the sending device adds its own information to the message it receives from the layer just above it and passes the whole package to the layer just below it.

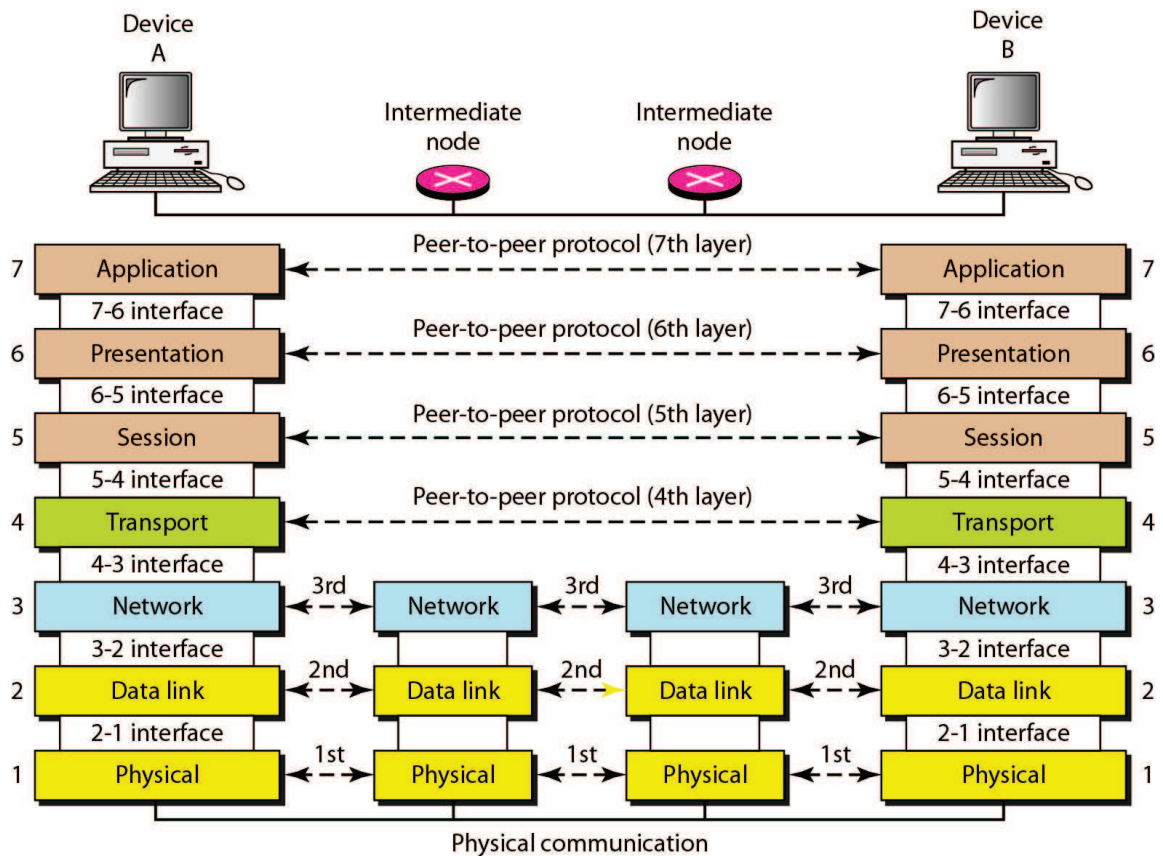


Fig. (2.2): The interaction between layers in the OSI model

Interfaces between Layers

The passing of the data and network information down through the layers of the sending device and back up through the layers of the receiving device is made possible by an interface between each pair of adjacent layers. Each interface defines the information and services a layer must provide for the layer above it. Well-defined interfaces and layer functions provide modularity to a network. As long as a layer provides the expected services to the layer above it, the specific implementation of its functions can be modified or replaced without requiring changes to the surrounding layers.

Organization of the Layers

The seven layers can be thought of as belonging to three subgroups:

- Layers 1, 2, and 3-physical, data link, and network-are the network support layers.
- Layers 5, 6, and 7-session, presentation, and application-can be thought of as the user support layers.

- Layer 4, the transport layer, links the two subgroups and ensures that what the lower layers have transmitted is in a form that the upper layers can use.

The upper OSI layers are almost always implemented in software; lower layers are a combination of hardware and software, except for the physical layer, which is mostly hardware.

In Figure 2.3, which gives an overall view of the OSI layers, D7 means the data unit at layer 7, D6 means the data unit at layer 6, and so on. The process starts at layer 7 (the application layer), then moves from layer to layer in descending, sequential order. At each layer, a **header**, or possibly a **trailer**, can be added to the data unit. Commonly, the trailer is added only at layer 2. When the formatted data unit passes through the physical layer (layer 1), it is changed into an electromagnetic signal and transported along a physical link.

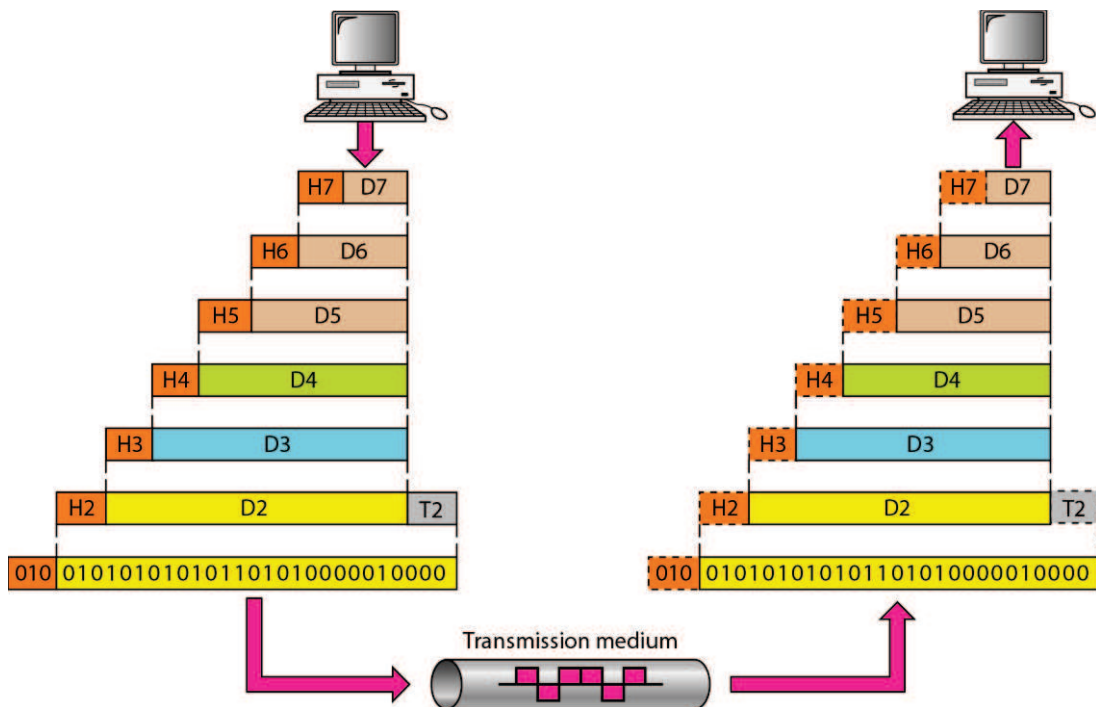


Fig. (2.3): An exchange using the OSI model.

Upon reaching its destination, the signal passes into layer 1 and is transformed back into digital form. The data units then move back up through the OSI layers. As each block of data reaches the next higher layer, the headers and trailers attached to it at the corresponding sending layer are removed, and actions appropriate to that layer are taken. By the time it reaches layer 7, the message is again in a form appropriate to the application and is made available to the recipient.

Encapsulation

Figure (2.2) reveals another aspect of data communications in the OSI model: encapsulation. A packet (header and data) at level 7 is encapsulated in a packet at level 6. The whole packet at level 6 is encapsulated in a packet at level 5, and so on. In other words, the data portion of a packet at level $N - 1$ carries the whole packet (data and header and maybe trailer) from level N . The concept is called *encapsulation*.

LAYERS IN THE OSI MODEL

In this section we briefly describe the functions of each layer in the OSI model.

Physical Layer

The physical layer coordinates the functions required to carry a bit stream over a physical medium. It deals with the mechanical and electrical specifications of the interface and transmission medium. *The physical layer is responsible for movements of individual bits from one hop (node) to the next.*

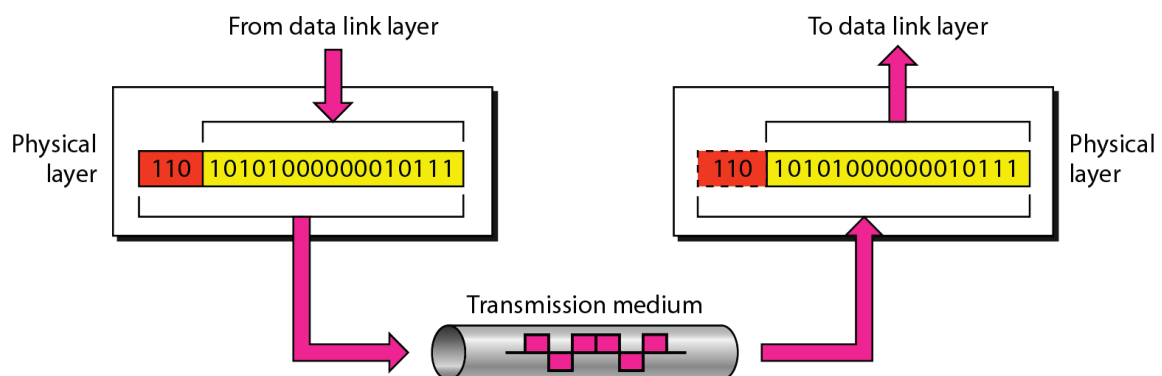


Fig. (2.4): Physical layer.

The physical layer is also concerned with the following:

1. Physical characteristics of interfaces and medium: The physical layer defines the characteristics of the interface between the devices and the transmission medium. It also defines the type of transmission medium.
2. Representation of bits: The physical layer data consists of a stream of bits (sequence of 0s or 1s) with no interpretation. To be transmitted, bits must be encoded into signals--electrical or optical. The physical layer defines the type of encoding (how 0s and 1s are changed to signals).

3. Data rate: The transmission rate is the number of bits sent each second, is also defined by the physical layer.

4. Synchronization of bits: The sender and receiver not only must use the same bit rate but also must be synchronized at the bit level. In other words, the sender and the receiver clocks must be synchronized.

5. Line configuration: The physical layer is concerned with the connection of devices to the media. In a point-to-point configuration, and a multipoint configuration.

6. Physical topology: The physical topology defines how devices are connected to make a network. Devices can be connected by using a mesh topology, a star topology, a ring topology, a bus topology or a hybrid topology.

7. Transmission mode: The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex.

Data Link Layer

The data link layer transforms the physical layer, a raw transmission facility, to a reliable link. It makes the physical layer appear error-free to the upper layer (network layer). Figure (2.5) shows the relationship of the data link layer to the network and physical layers. **The data link layer is responsible for moving frames from one hop (node) to the next.**

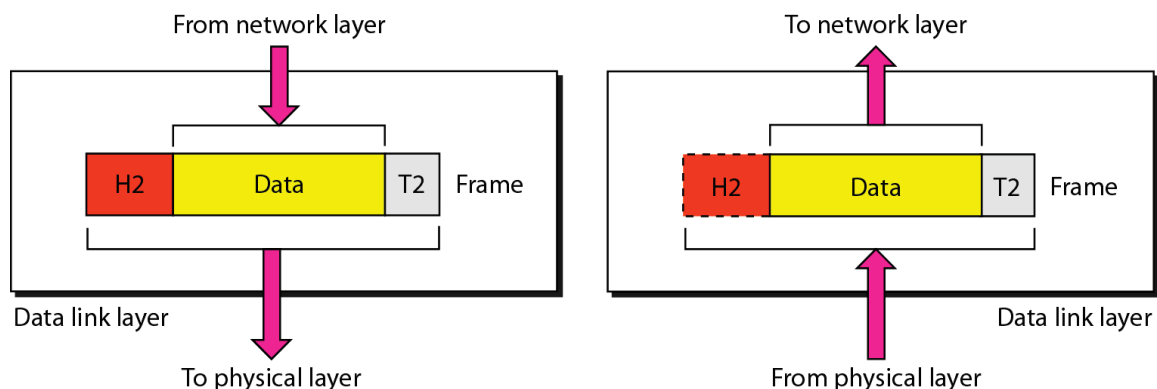


Fig. (2.5): Data link layer

Other responsibilities of the data link layer include the following:

1. Framing: The data link layer divides the stream of bits received from the network layer into manageable data units called frames.

2. Physical addressing: If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame. If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects the network to the next one.

3. Flow control: If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.

4. Error control: The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.

5. Access control: When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

Figure (2.6) illustrates hop-to-hop (node-to-node) delivery by the data link layer.

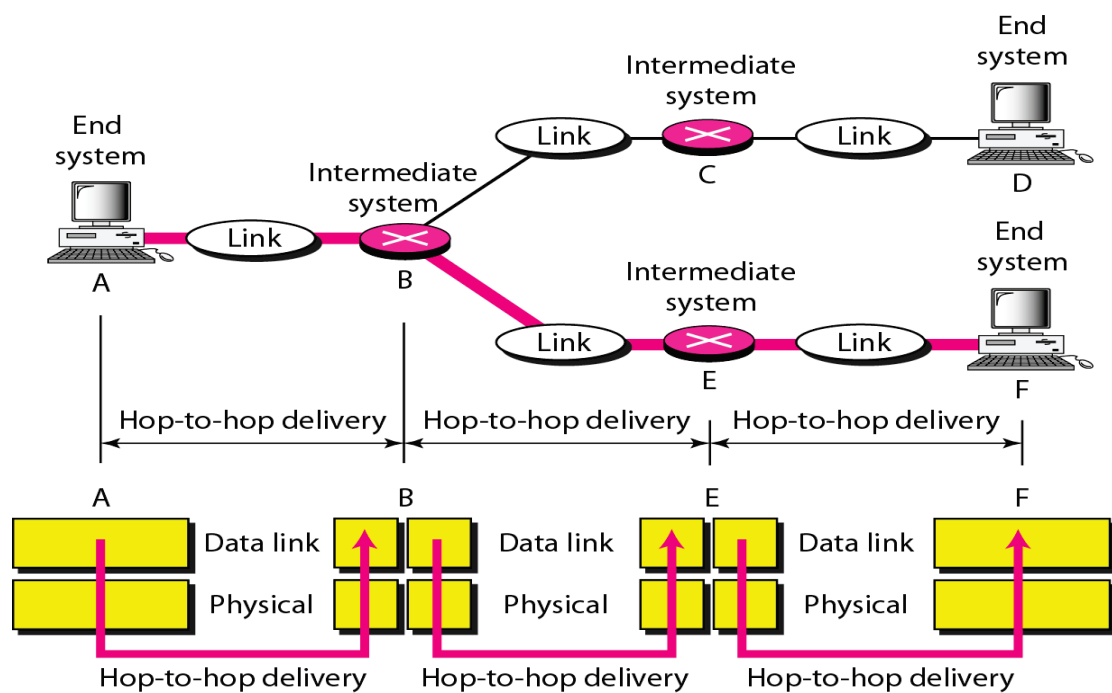


Fig. (2.6): Hop-to-hop delivery.

Network Layer

The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links). Whereas the data link layer oversees the delivery of the packet

between two systems on the same network (links), the network layer ensures that each packet gets from its point of origin to its final destination.

If two systems are connected to the same link, there is usually no need for a network layer. However, if the two systems are attached to different networks (links) with connecting devices between the networks (links), there is often a need for the network layer to accomplish source-to-destination delivery. **The network layer is responsible for the delivery of individual packets from the source host to the destination host.**

Figure (2.7) shows the relationship of the network layer to the data link and transport layers.

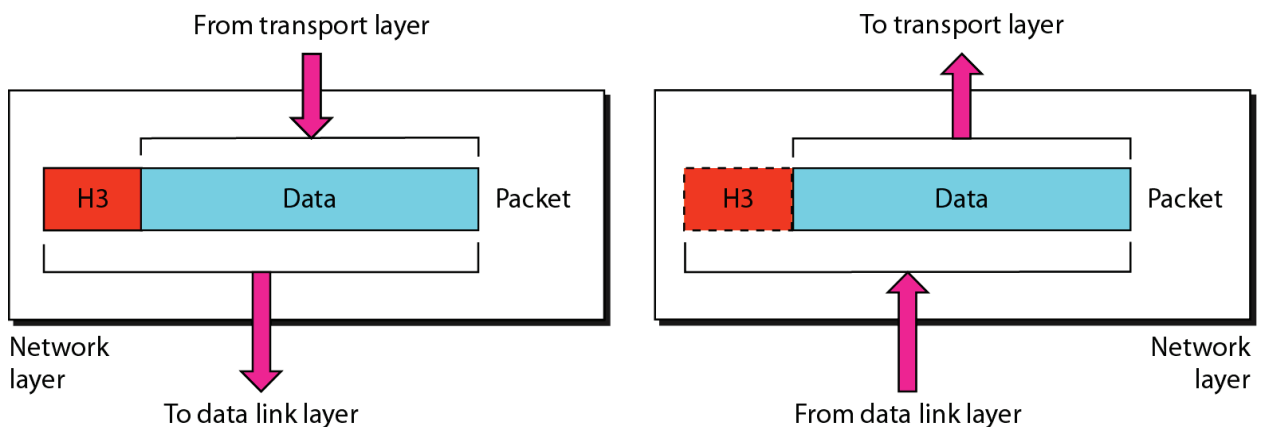


Fig. (2.7): Network layer.

Other responsibilities of the network layer include the following:

1. Logical addressing: If a packet passes the network boundary, we need another addressing system to help distinguish the source and destination systems. The network layer adds a header to the packet coming from the upper layer that, among other things, includes the logical addresses of the sender and receiver.
2. Routing: When independent networks or links are connected to create internetworks (network of networks) or a large network, the connecting devices (called routers or switches) route or switch the packets to their final destination. One of the functions of the network layer is to provide this mechanism.

Figure 2.8 illustrates end-to-end delivery by the network layer.

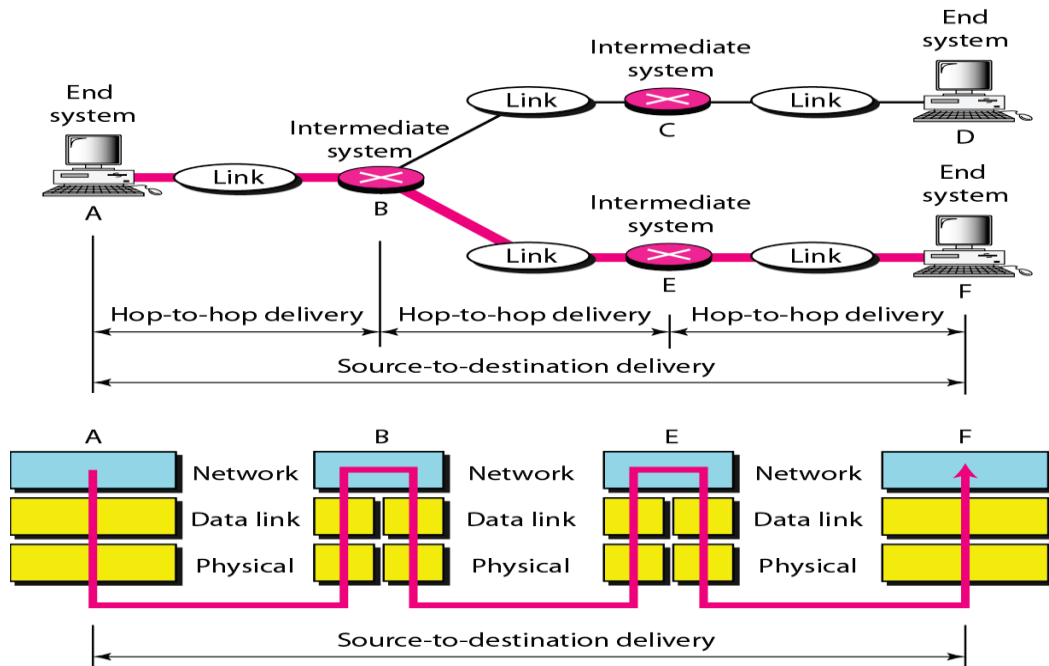


Fig. (2.8): Source-to-destination delivery.

Transport Layer

The transport layer is responsible for process-to-process delivery of the entire message. A process is an application program running on a host. Whereas the network layer oversees source-to-destination delivery of individual packets, it does not recognize any relationship between those packets. It treats each one independently, as though each piece belonged to a separate message, whether or not it does. The transport layer, on the other hand, ensures that the whole message arrives intact and in order, overseeing both error control and flow control at the source-to-destination level. **The transport layer is responsible for the delivery of a message from one process to another.**

Figure (2.9) shows the relationship of the transport layer to the network and session layers.

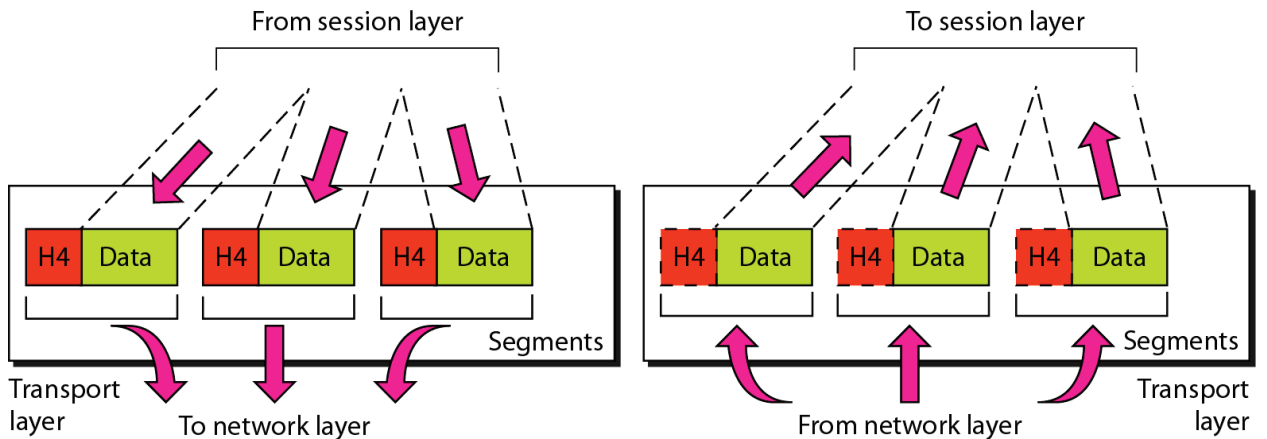


Fig. (2.9): Transport layer.

Other responsibilities of the transport layer include the following:

1. Service-point addressing: Computers often run several programs at the same time. For this reason, source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on one computer to a specific process (running program) on the other. The transport layer header must therefore include a type of address called a service-point address (or port address). The network layer gets each packet to the correct computer; the transport layer gets the entire message to the correct process on that computer.

2. Segmentation and reassembly: A message is divided into transmittable segments, with each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost in transmission.

3. Connection control: The transport layer can be either connectionless or connection-oriented. A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connection-oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data are transferred, the connection is terminated.

4. Flow control: Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.

5. Error control: Like the data link layer, the transport layer is responsible for error control. However, error control at this layer is performed process-to process rather than across a single link. The sending transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, or duplication). Error correction is usually achieved through retransmission.

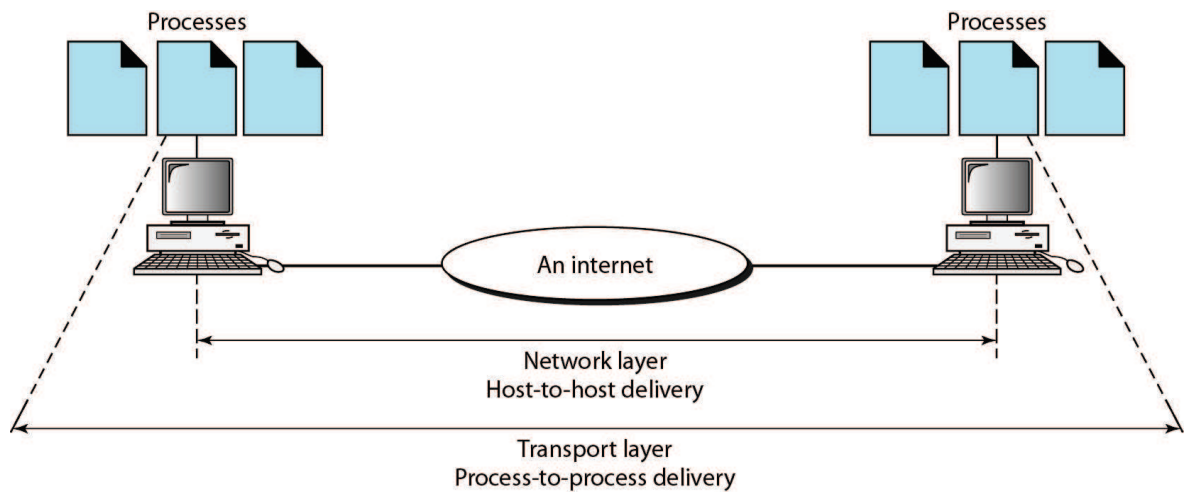


Fig. (2.10): Reliable process-to-process delivery of a message.

Session Layer

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

Specific responsibilities of the session layer include the following:

1. **Dialog control:** The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either halfduplex (one way at a time) or full-duplex (two ways at a time) mode.
2. **Synchronization:** The session layer allows a process to add checkpoints, or synchronization points, to a stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 523, the only pages that need to be resent after system recovery are pages 501 to 523. Pages previous to 501 need not be resent. Figure 2.11 illustrates the relationship of the session layer to the transport and presentation layers.

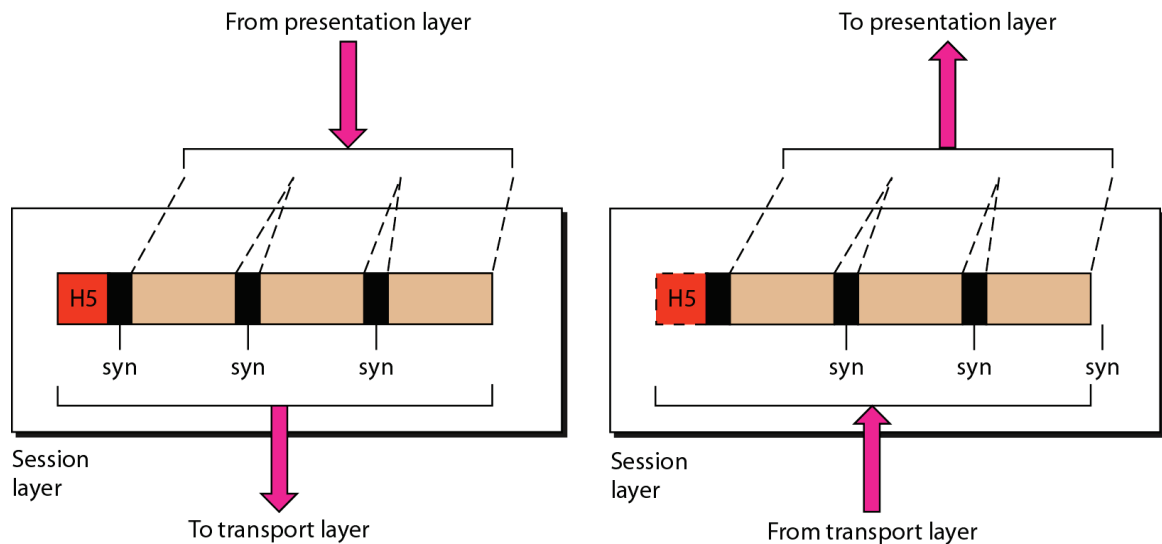


Fig. (2.11): Session layer.

Presentation Layer

The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems. Figure (2.12) shows the relationship between the presentation layer and the application and session layers. **The presentation layer is responsible for translation, compression, and encryption.**

Specific responsibilities of the presentation layer include the following:

1. Translation: The processes (running programs) in two systems are usually exchanging information in the form of character strings, numbers, and so on. The information must be changed to bit streams before being transmitted. Because different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. The presentation layer at the sender changes the information from its sender-dependent format into a common format. The presentation layer at the receiving machine changes the common format into its receiver-dependent format.
2. Encryption: To carry sensitive information, a system must be able to ensure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.
3. Compression: Data compression reduces the number of bits contained in the information. Data compression becomes particularly important in the transmission of multimedia such as text, audio, and video.

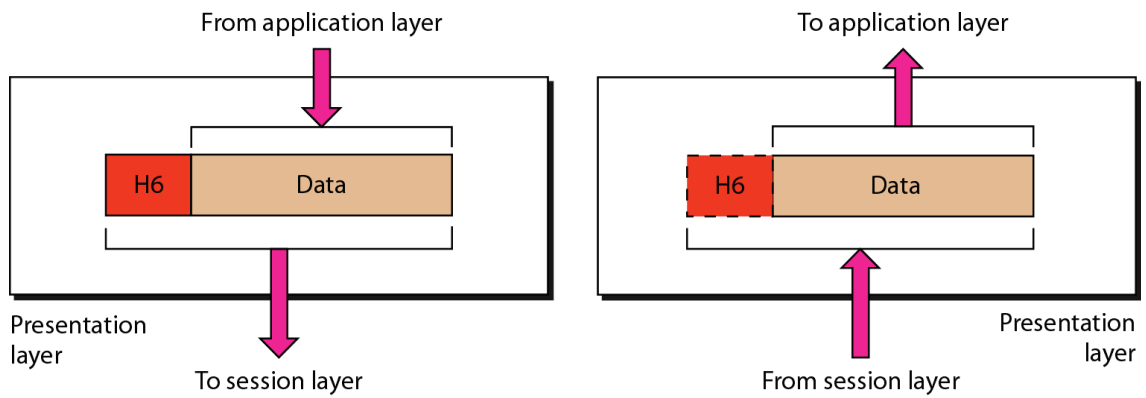


Fig. (2.12): Presentation layer.

Application Layer

The application layer enables the user, whether human or software, to access the network. It provides user interfaces and support for services such as electronic mail, remote file access and transfer, shared database management, and other types of distributed information services.

Figure (2.13) shows the relationship of the application layer to the user and the presentation layer. Of the many application services available, the figure shows only three:

X.400 (message-handling services), X.500 (directory services), and file transfer, access, and management (FTAM). The user in this example employs X.400 to send an e-mail message.

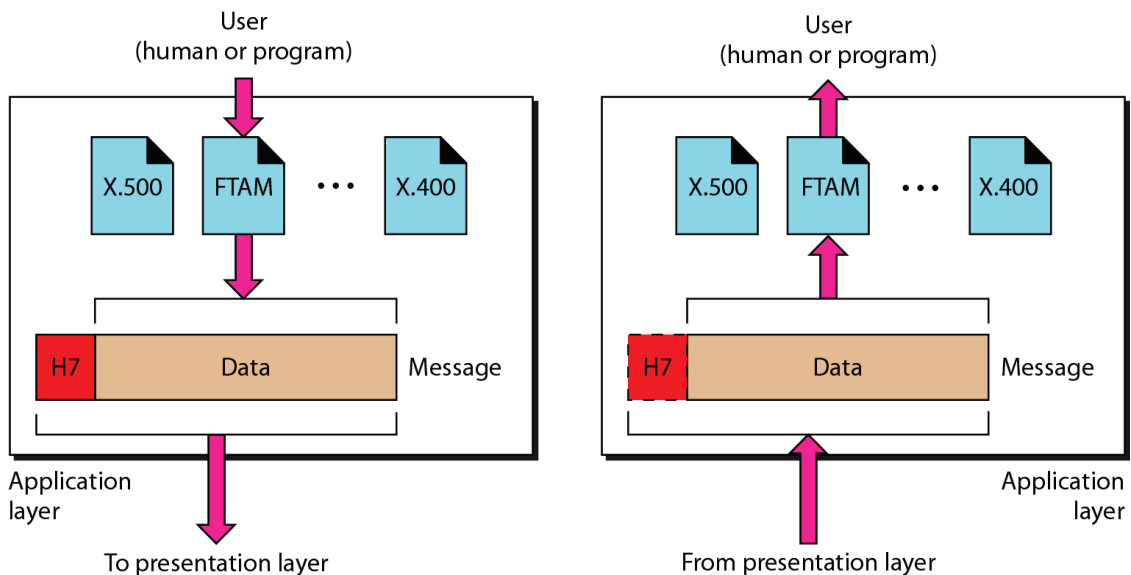


Fig. (2.13): Application layer.

The application layer is responsible for providing services to the user.

Specific services provided by the application layer include the following:

1. Network virtual terminal: A network virtual terminal is a software version of a physical terminal, and it allows a user to log on to a remote host. To do so, the application creates a software emulation of a terminal at the remote host. The user's computer talks to the software terminal which, in turn, talks to the host, and vice versa. The remote host believes it is communicating with one of its own terminals and allows the user to log on.

2. File transfer, access, and management: This application allows a user to access files in a remote host (to make changes or read data), to retrieve files from a remote computer for use in the local computer, and to manage or control files in a remote computer locally.

3. Mail services: This application provides the basis for e-mail forwarding and storage.

4. Directory services: This application provides distributed database sources and access for global information about various objects and services.

Summary of Layers

Figure (2.14) shows a summary of duties for each layer.

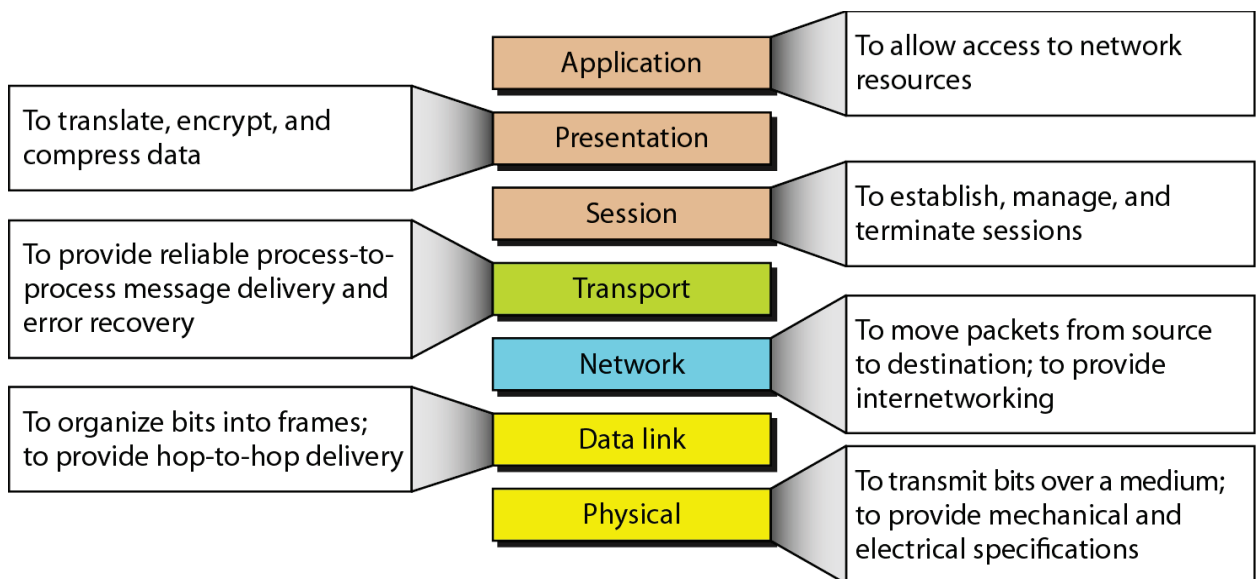


Fig. (2.14): Summary of layers.

TCP/IP PROTOCOL SUITE

The layers in the TCP/IP protocol suite do not exactly match those in the OSI model. The original TCP/IP protocol suite was defined as having four layers: host-to-network, internet, transport, and application. However, when TCP/IP is compared to OSI, we can say that the TCP/IP protocol suite is made of five layers: physical, data link, network, transport, and application. (See Figure (2.15)).

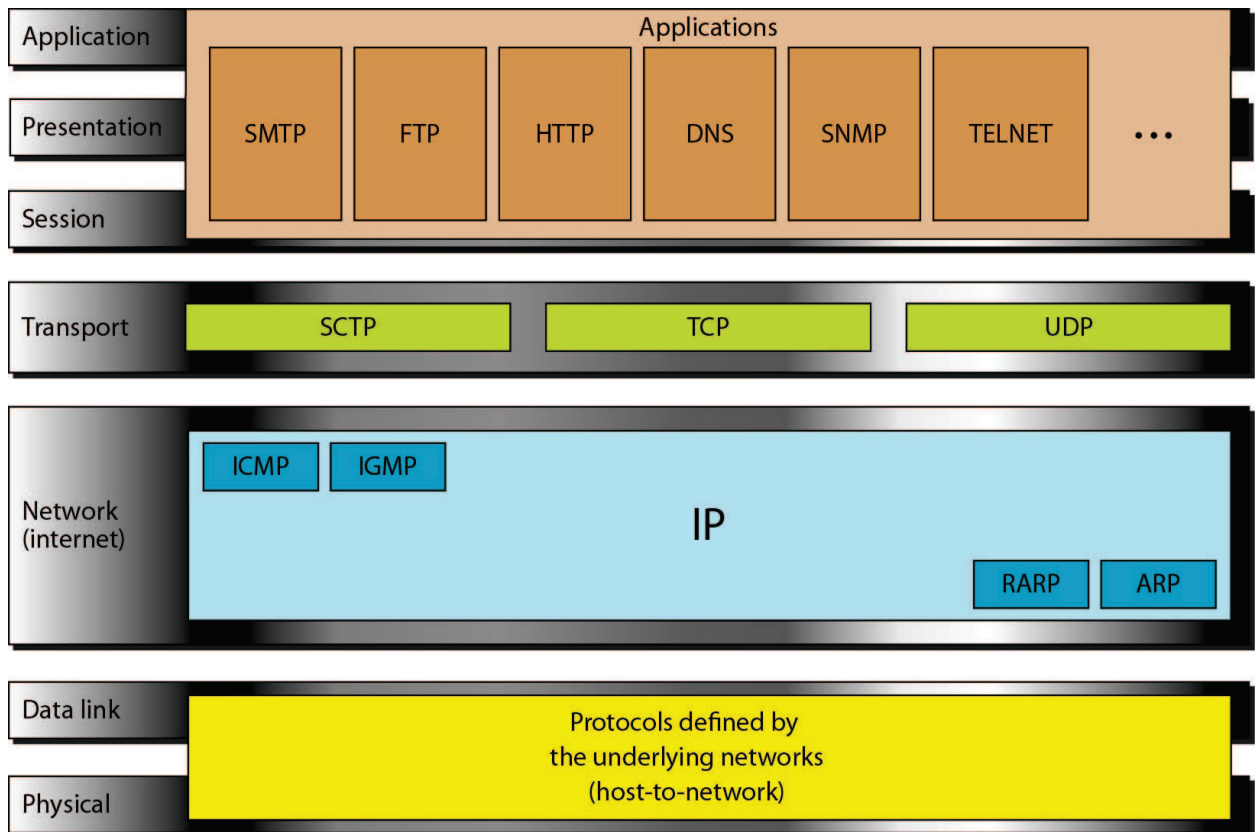


Fig. (2.15): TCP/IP and OSI Model.

At the transport layer, TCP/IP defines three protocols: Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Stream Control Transmission Protocol (SCTP). At the network layer, the main protocol defined by TCP/IP is the Internetworking Protocol (IP); there are also some other protocols that support data movement in this layer.

Physical and Data Link Layers

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

Network Layer

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

Internetworking Protocol (IP)

The Internetworking Protocol (IP) is the transmission mechanism used by the TCP/IP protocols. It is an unreliable and connectionless protocol—a best-effort delivery service. The term best effort means that IP provides no error checking or tracking. IP assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees.

IP transports data in packets called datagrams, each of which is transported separately. Datagrams can travel along different routes and can arrive out of sequence or be duplicated. IP does not keep track of the routes and has no facility for reordering datagrams once they arrive at their destination.

Address Resolution Protocol

The Address Resolution Protocol (ARP) is used to associate a logical address with a physical address. On a typical physical network, such as a LAN, each device on a link is identified by a physical or station address, usually imprinted on the network interface card (NIC). ARP is used to find the physical address of the node when its Internet address is known.

Reverse Address Resolution Protocol

The Reverse Address Resolution Protocol (RARP) allows a host to discover its Internet address when it knows only its physical address. It is used when a computer is connected to a network for the first time or when a diskless computer is booted.

Internet Control Message Protocol

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

Internet Group Message Protocol

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

Transport Layer

Traditionally the transport layer was represented in TCP/IP by two protocols: TCP and UDP. IP is a host-to-host protocol, meaning that it can deliver a packet from one physical device to another. UDP and TCP are transport level protocols responsible for delivery of a message from a process (running program) to another process. A new transport layer protocol, SCTP, has been devised to meet the needs of some newer applications.

User Datagram Protocol

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

Transmission Control Protocol

The Transmission Control Protocol (TCP) provides full transport-layer services to applications. TCP is a reliable stream transport protocol. The term stream, in this context, means connection-oriented: A connection must be established between both ends of a transmission before either can transmit data.

At the sending end of each transmission, TCP divides a stream of data into smaller units called segments. Each segment includes a sequence number for reordering after receipt, together with an acknowledgment number for the segments received. Segments are carried across the internet inside of IP datagrams. At the receiving end, TCP collects each datagram as it comes in and reorders the transmission based on sequence numbers.

Stream Control Transmission Protocol

The Stream Control Transmission Protocol (SCTP) provides support for newer applications such as voice over the Internet. It is a transport layer protocol that combines the best features of UDP and TCP.

Application Layer

The application layer in TCP/IP is equivalent to the combined session, presentation, and application layers in the OSI model. Many protocols are defined at this layer.

ADDRESSING

Four levels of addresses are used in an internet employing the TCP/IP protocols: Physical (link) addresses, logical (IP) addresses, port addresses, and specific addresses (see Figure (2.16)).

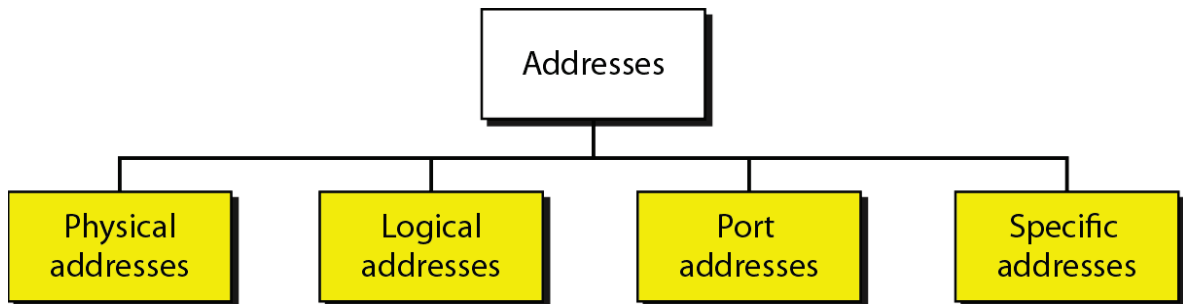


Fig. (2.16): Addresses in TCP/IP

Each address is related to a specific layer in the TCP/IP architecture, as shown in Figure (2.17).

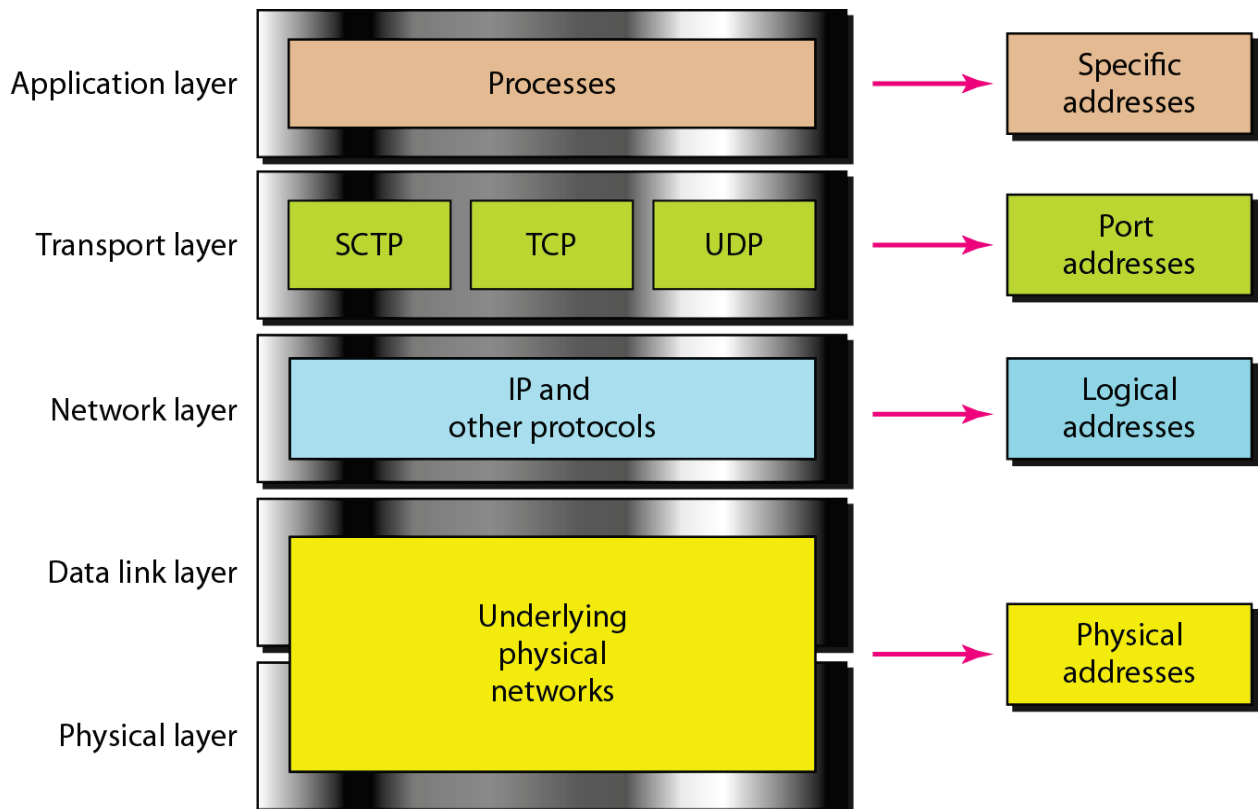


Fig. (2.17): Relationship of layers and addresses in TCP/IP

Physical Addresses

The physical address is included in the frame used by the data link layer. It is the lowest-level address. The size and format of these addresses vary depending on the network. For example, Ethernet uses a 6-byte (48-bit) physical address that is imprinted on the network interface card (NIC). LocalTalk (Apple), however, has a 1-byte dynamic address that changes each time the station comes up.

Most local-area networks use a 48-bit (6-byte) physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon, as shown below:

07:01:02:01:2C:4B
A 6-byte (12 hexadecimal digits) physical address.

Logical Addresses

Logical addresses are necessary for universal communications that are independent of underlying physical networks. The logical addresses are designed for this purpose. A logical address in the Internet is currently a 32-bit address that can uniquely define a host connected to the Internet. No two publicly addressed and visible hosts on the Internet can have the same IP address.

Example 2.1

Figure (2.18) shows a part of an internet with two routers connecting three LANs. Each device (computer or router) has a pair of addresses (logical and physical) for each connection. In this case, each computer is connected to only one link and therefore has only one pair of addresses.

Each router, however, is connected to three networks (only two are shown in the figure). So each router has three pairs of addresses, one for each connection. Although it may be obvious that each router must have a separate physical address for each connection, it may not be obvious why it needs a logical address for each connection.

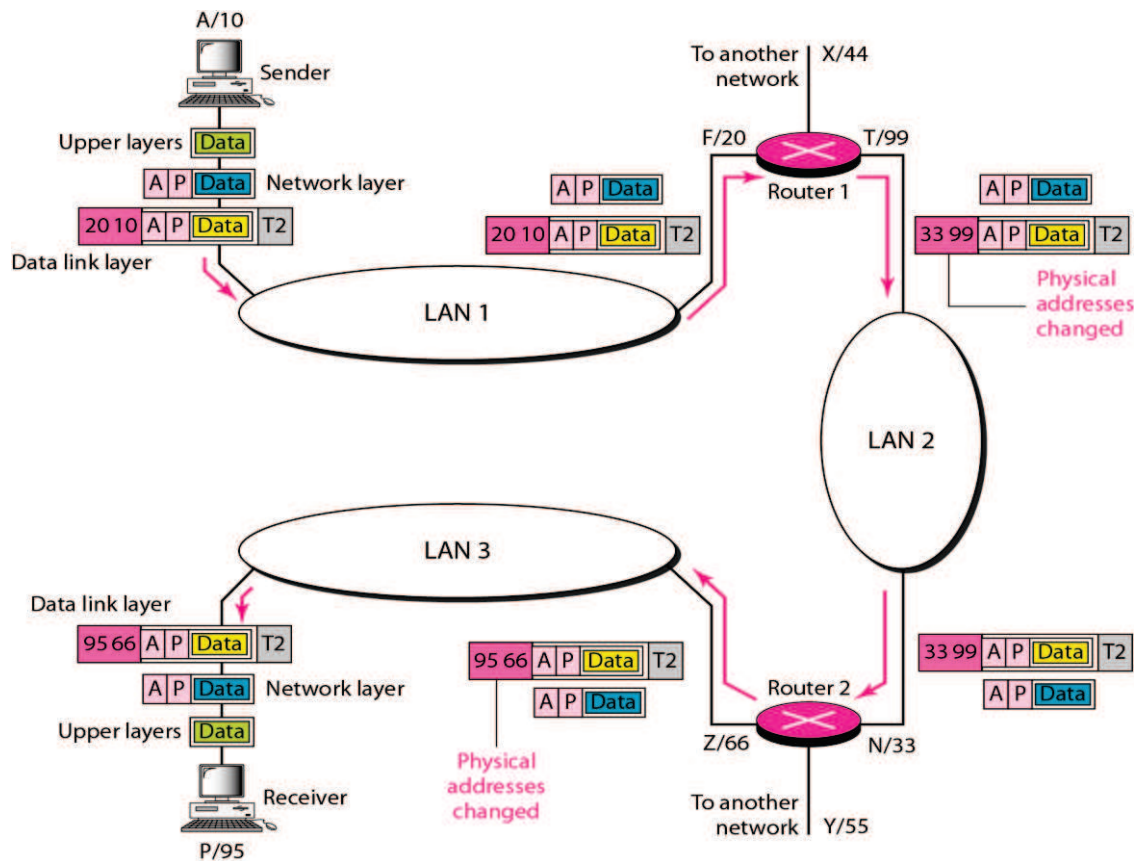


Fig. (2.18): IP addresses

The physical addresses will change from hop to hop, but the logical addresses usually remain the same.

Port Addresses

The IP address and the physical address are necessary for a quantity of data to travel from a source to the destination host. However, arrival at the destination host is not the final objective of data communications on the Internet. A system that sends nothing but data from one computer to another is not complete. Today, computers are devices that can run multiple processes at the same time. The end objective of Internet communication is a process communicating with another process. For example, computer A can communicate with computer C by using TELNET. At the same time, computer A communicates with computer B by using the File Transfer Protocol (FTP). For these processes to receive data simultaneously, we need a method to label the different processes. In other words, they need addresses. In the

TCP/IP architecture, the label assigned to a process is called a port address. A port address in TCP/IP is 16 bits in length.

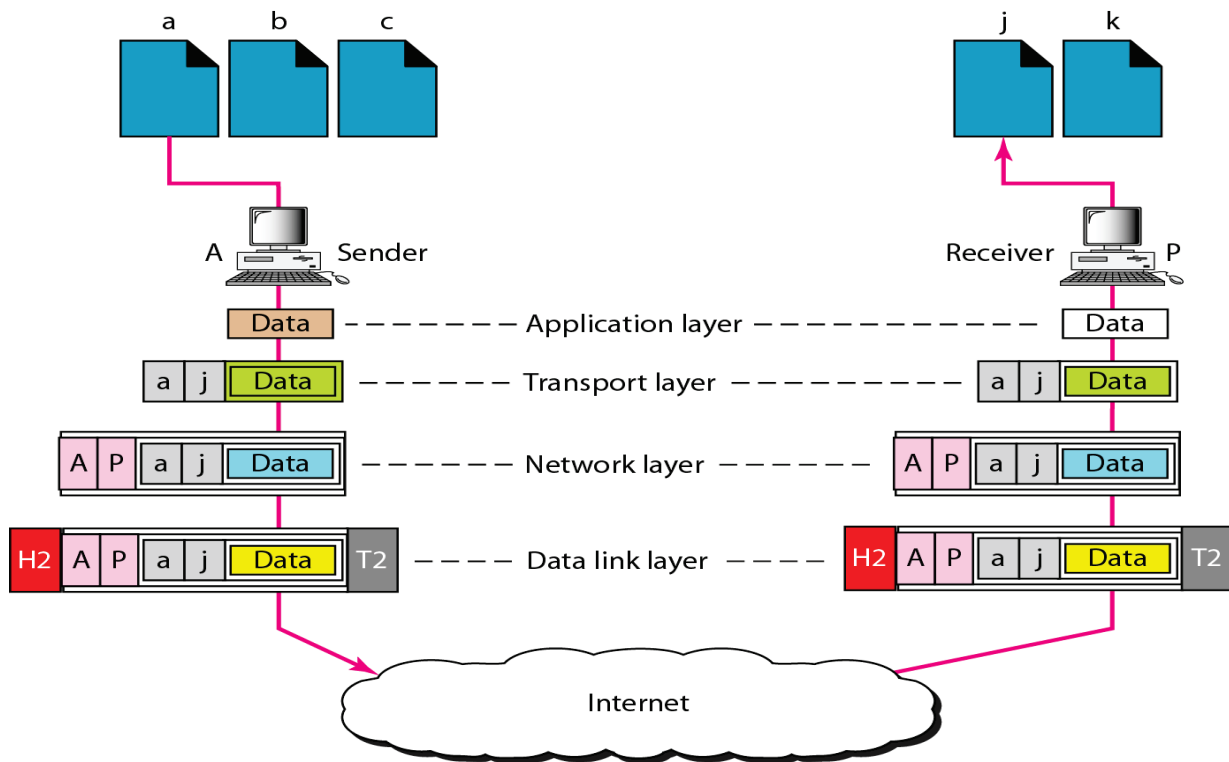


Fig. (2.19): Port addresses

Example 2.2

Figure (2.19) shows two computers communicating via the Internet. The sending computer is running three processes at this time with port addresses a, b, and c. The receiving computer is running two processes at this time with port addresses j and k. Process a in the sending computer needs to communicate with process j in the receiving computer. Note that although physical addresses change from hop to hop, logical and port addresses remain the same from the source to destination.

A port address is a 16-bit address represented by one decimal number as shown.

753

A 16-bit port address represented as one single number.

Specific Addresses

Some applications have user-friendly addresses that are designed for that specific address. Examples include the e-mail address (for example, `forouzan@fhda.edu`) and the Universal Resource Locator (URL) (for example, `www.mhhe.com`). The first defines the recipient of an e-mail; the second is used to find a document on the World Wide Web. These addresses, however, get changed to the corresponding port and logical addresses by the sending computer.