# Tikrit University

## COLLAGE OF COMPUTER SCIENCE AND MATHEMATICS

# Computer Networking

## NETWORK SOFTWARE

**4th stage**

**Lecturer 5**

**Majid Hamid Ali**

**2023– 2024**

## 1. Network Software

Network Software is a set of primitives that define the protocol between two machines. The network software resolves an ambiguity among different types of networks making it possible for all the machines in the network to connect and communicate with one another and share information. network software is the information, data or programming used to make it possible for computers to communicate or connect to one another. Network software is used to efficiently share information among computers. It encloses the information to be sent in a "package" that contains a "header" and a "trailer". The header and trailer contain information for the receiving computer, such as the address of that computer and how the information package is coded. Information is transferred between computers as either electrical signals in electric wires, as light signals in fiber-optic cables, or as electromagnetic waves through space.

## 2. Protocol Hierarchies

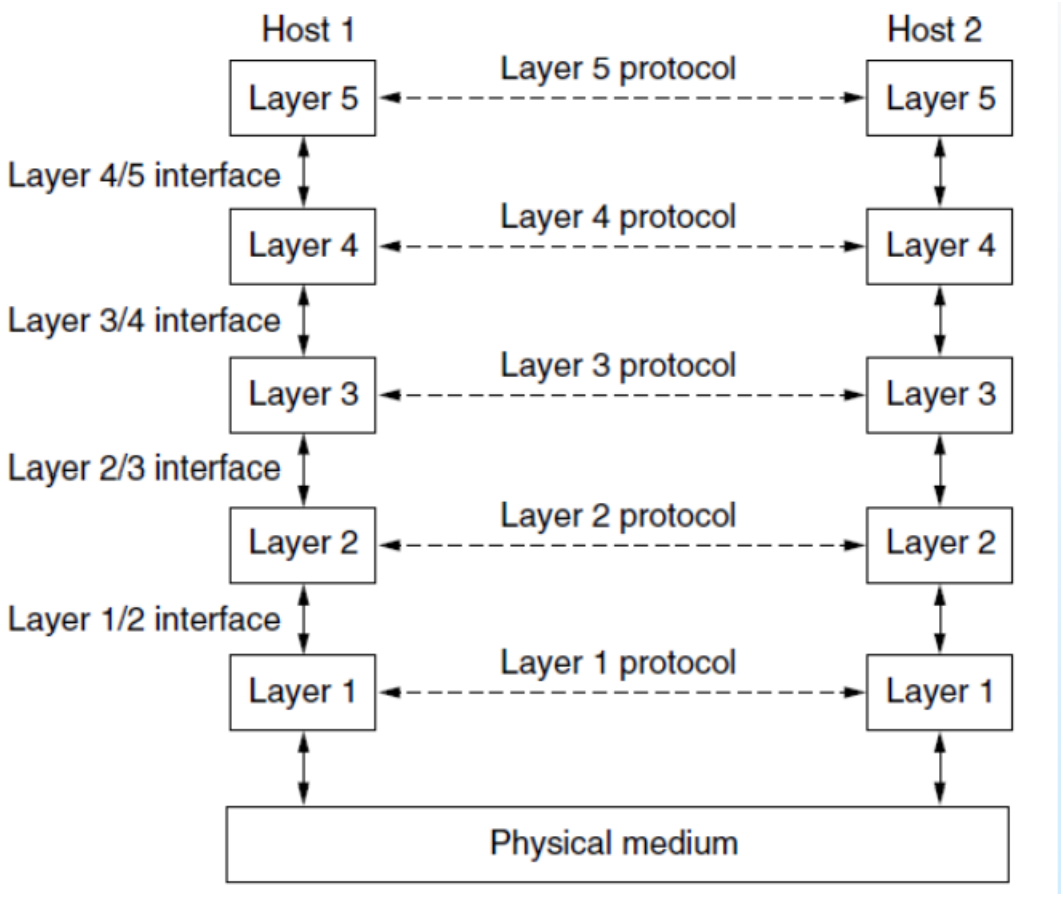
To reduce their design complexity, most networks are organized as a stack of layers or levels, each one built upon the one below it. The number of layers, the name of each layer, the contents of each layer, and the function of each layer differ from network to network. The purpose of each layer is to offer certain services to the higher layers, shielding those layers from the details of how the offered services are actually implemented. In a sense, each layer is a kind of virtual machine, offering certain services to the layer above it.

**Protocol** is an agreement between the communicating parties on how communication is to proceed. It is a set of rules governing the format and meaning of frames, packets, or messages that are exchanged by peer entities within a layer.

A five-layer network is illustrated in the Fig. below. The entities comprising the corresponding layers on different machines are called peers. The peers may be processes, hardware devices, or even human beings.

In reality, no data are directly transferred from layer (n) on one machine to layer (n) on another machine. Instead, each layer passes data and control information to the layer immediately below

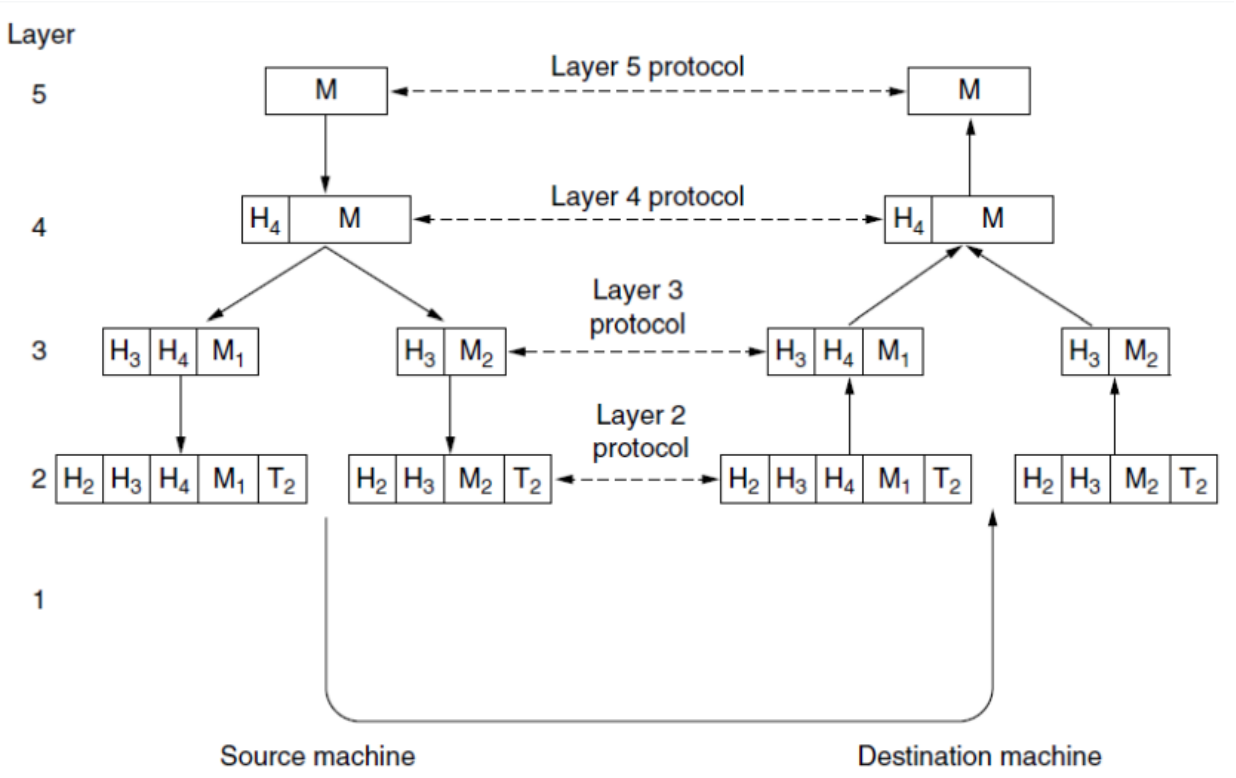
it, until the lowest layer is reached. Below layer 1 is the physical medium through which actual communication occurs. Between each pair of adjacent layers is an interface. The interface defines which primitive operations and services the lower layer makes available to the upper one.




A set of layers and protocols is called a network architecture. The specification of an architecture must contain enough information to allow an implementer to write the program or build the hardware for each layer so that it will correctly obey the appropriate protocol.

A message, M, is produced by an application process running in layer 5 and given to layer 4 for transmission. Layer 4 puts a header in front of the message to identify the message and passes the result to layer 3. The header includes control information, such as sequence numbers, to allow layer 4 on the destination machine to deliver messages in the right order if the lower layers do not maintain sequence. In some layers, headers can also contain sizes, times, and other control fields. In many networks, there is no limit to the size of messages transmitted in the layer 4 protocol, but there is nearly always a limit imposed by the layer 3 protocol. Consequently, layer 3 must break

up the incoming messages into smaller units, packets, pre-pending a layer 3 header to each packet. In this example, M is split into two parts, M1 and M2. Layer 3 decides which of the outgoing lines to use and passes the packets to layer 2. Layer 2 adds not only a header to each piece, but also a trailer, and gives the resulting unit to layer 1 for physical transmission. At the receiving machine the message moves upward, from layer to layer, with headers being stripped off as it progresses. None of the headers for layers below (n) are passed up to layer (n). The important thing to understand about the figure below is the relation between the virtual and actual communication and the difference between protocols and interfaces. The peer processes in layer 4, for example, conceptually think of their communication as being "horizontal," using the layer 4 protocol. Each one is likely to have a procedure called something like Send To Other Side and Get From Other Side, even though these procedures actually communicate with lower layers across the 3/4 interface, not with the other side.



## 3. The Benefit of Layered Protocols

1. ready adaptation of successful protocols to new technology (prevent obsolescence)
2. migration of protocols from software implementation (slow) to hardware (fast) as they evolve
3. separate data and control information
4. support differing levels of abstraction (message, packet, frame) with different sizes
5. allow segmentation of large messages
6. peer process abstraction facilitates reduction of difficult design task (a network architecture) into smaller manageable tasks (protocol layer architecture)
7. typically, lower layer protocols of "network software" are implemented in silicon (hardware)

## 4. Design Issues for the Layers: -

Some of the key design issues that occur in computer networks are present in several layers. Below, we will briefly mention some of the more important ones: -

1- Every layer needs a mechanism for identifying senders and receivers, some form of addressing is needed in order to specify a specific destination.

2- Design decisions concerns the rules for data transfer. In some systems, data only travel in one direction; in others, data can go both ways. The protocol must also determine how many logical channels the connection corresponds to and what their priorities are. Many networks provide at least two logical channels per connection, one for normal data and one for urgent data.

3- Error control is an important issue because physical communication circuits are not perfect. Many error-detecting and error-correcting codes are known, but both ends of the connection must agree on which one is being used. In addition, the receiver must have some way of telling the sender which messages have been correctly received and which have not.

4- the protocol must make explicit provision for the receiver to allow the pieces to be reassembled properly.

5- how to keep a fast sender from swamping a slow receiver with data. Various solutions have been proposed, some of them involve some kind of feedback from the receiver to the sender, either directly or indirectly, about the receiver's current situation.

6- The inability of all processes to accept arbitrarily long messages. This property leads to mechanisms for disassembling, transmitting, and then reassembling messages .

7- When there are multiple paths between source and destination, a route must be chosen. Sometimes this decision must be split over two or more layers. For example, to send data from London to Rome, a high-level decision might have to be made to transit France or Germany based on their respective privacy laws. Then a low-level decision might have to made to select one of the available circuits based on the current traffic load. This topic is called routing.

## 5. Connection-Oriented and Connectionless Services- :

Layers can offer two different types of service to the layers above them: connection oriented and connectionless:-

### Connection-oriented service

is modeled after the telephone system. To talk to someone, you pick up the phone, dial the number, talk, and then hang up . Similarly, to use a connection-oriented network service, the service user first establishes a connection, uses the connection, and then releases the connection .The essential aspect of a connection is that it acts like a tube: the sender pushes objects (bits) in at one end, and the receiver takes them out at the other end. In most cases the order is preserved so that the bits arrive in the order they were sent. typical situation in which a reliable connection-oriented service is appropriate is file transfer. The owner of the file wants to be sure that all the bits arrive correctly and in the same order they were sent.

### connectionless service

is modeled after the postal system. Each message (letter) carries the full destination address, and each one is routed through the system independent of all the others. Normally, when two messages are sent to the same destination, the first one sent will be the first one to arrive. However, it is possible that the first one sent can be delayed so that the second one arrives first.

Some services are reliable in the sense that they never lose data. Usually, a reliable service is implemented by having the receiver acknowledge the receipt of each message so the sender is sure that it arrived. The acknowledgement process introduces overhead and delays, which are often worth it but are sometimes undesirable.

for others applications, the transit delays introduced by acknowledgements are unacceptable. One such application is digitized voice traffic. It is preferable for telephone users to hear a bit of noise on the line from time to time than to experience a delay waiting for acknowledgements. Similarly, when transmitting a video conference, having a few pixels wrong is no problem, but having the image jerk along as the flow stops to correct errors is irritating.

| | Service | Example |
|---|---|---|
| Connection-oriented | Reliable message stream | Sequence of pages |
| | Reliable byte stream | Movie download |
| | Unreliable connection | Voice over IP |
| Connectionless | Unreliable datagram | Electronic junk mail |
| | Acknowledged datagram | Text messaging |
| | Request-reply | Database query |

## 6. OSI Model

The OSI model (minus the physical medium) is shown in Figure bellow. This model is based on a proposal developed by the International Standards Organization (ISO) as a first step toward international standardization of the protocols used in the various layers (Day and Zimmermann, 1983). It was revised in 1995 (Day, 1995). The model is called the ISO OSI (Open Systems Interconnection) Reference Model because it deals with connecting open systems—that is, systems that are open for communication with other systems. We will just call it the OSI model for short.

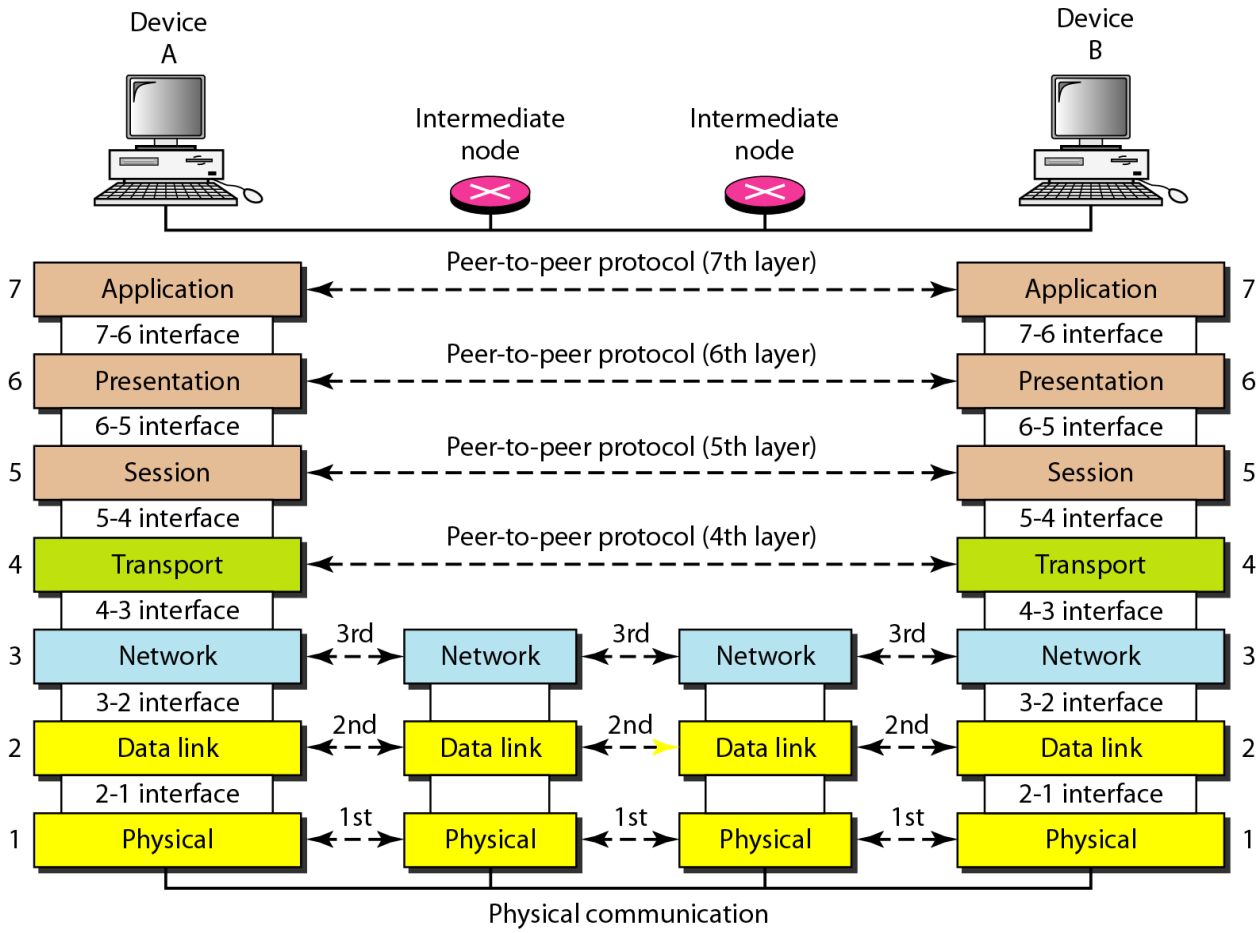
The OSI model has seven layers. The principles that were applied to arrive at the seven layers can be briefly summarized as follows:

1. A layer should be created where a different abstraction is needed.
2. Each layer should perform a well-defined function.
3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
4. The layer boundaries should be chosen to minimize the information flow across the interfaces.
5. The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that the architecture does not become unwieldy.

Below we will discuss each layer of the model in turn, starting at the bottom layer. Note that the OSI model itself is not a network architecture because it does not specify the exact services and protocols to be used in each layer. It just tells what each layer should do. However, ISO has also produced standards for all the layers, although these are not part of the reference model itself. Each one has been published as a separate international standard. The model (in part) is widely used although the associated protocols have been long forgotten.

## 7. Benefits of OSI Model:

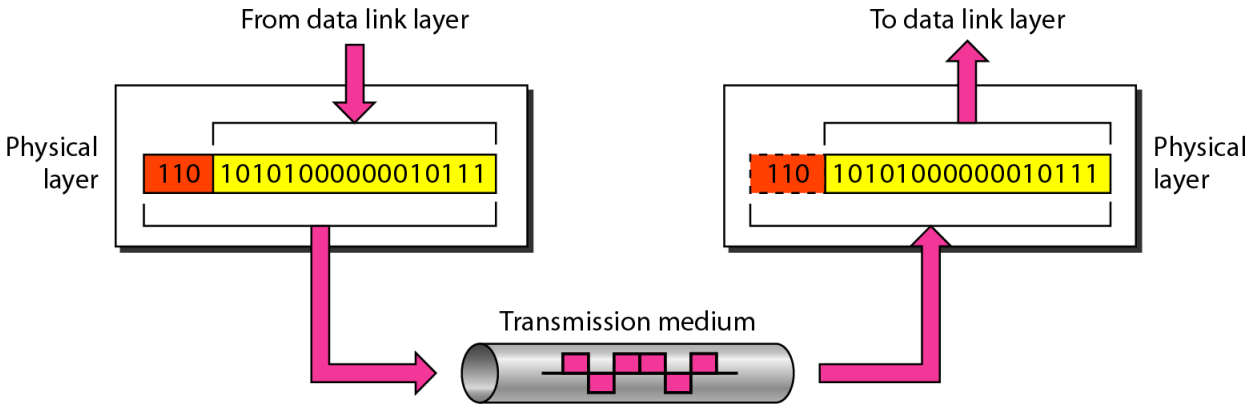
1- It breaks network communication into smaller, more manageable parts.
2- It standardizes network components to allow multiple vendor development and support.
3- It allows different types of network hardware and software to communicate with each other.
4- It prevents changes in one layer from affecting other layers.
5- It divides network communication into smaller parts to make learning it easier to understand.




## The Physical Layer

The **physical layer** is concerned with transmitting raw bits over a communication channel. The design issues have to do with making sure that when one side sends a 1 bit it is received by the other side as a 1 bit, not as a 0 bit. Typical questions here are what electrical signals should be used to represent a 1 and a 0, how many nanoseconds a bit lasts, whether transmission may proceed simultaneously in both directions, how the initial connection is established, how it is torn down when both sides are finished, how many pins the network connector has, and what each pin is used for. These design issues largely deal with mechanical, electrical, and timing interfaces, as well as the physical transmission medium, which lies below the physical layer.

- It defines the transmission of data across the communications medium and translation of binary data into signals.
- Mode of transmission over the link i.e Simplex or Half Duplex or Full Duplex
- It defines the transmission rate of bits per second.




**NOTE:- The physical layer is responsible for movements of individual bits from one hop (node) to the next.**

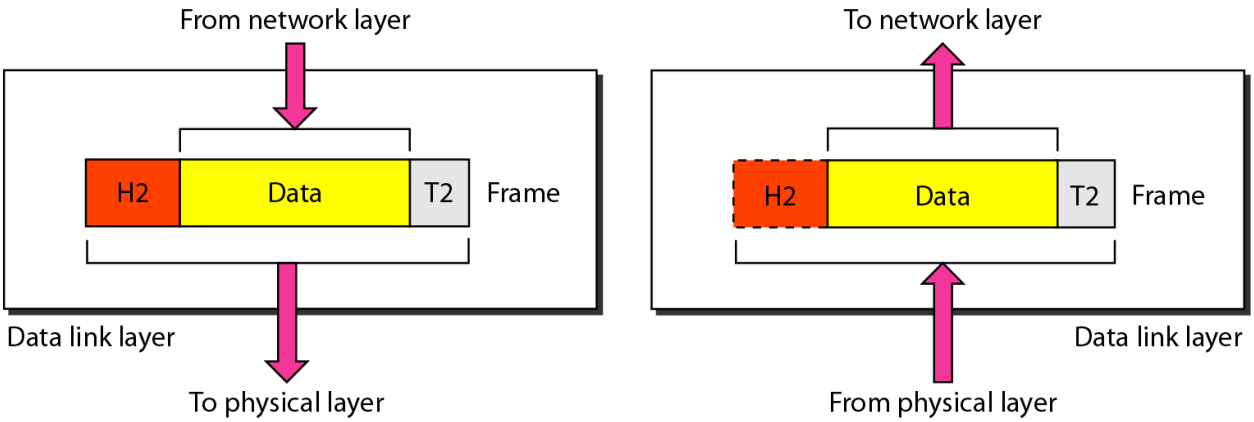
## The Data Link Layer

The main task of the **data link layer** is to transform a raw transmission facility into a line that appears free of undetected transmission errors. It does so by masking the real errors so the network layer does not see them. It accomplishes this task by having the sender break up the input data into **data frames** (typically a few hundred or a few thousand bytes) and transmit the frames sequentially. If the service is reliable, the receiver confirms correct receipt of each frame by sending back an **acknowledgement frame**.

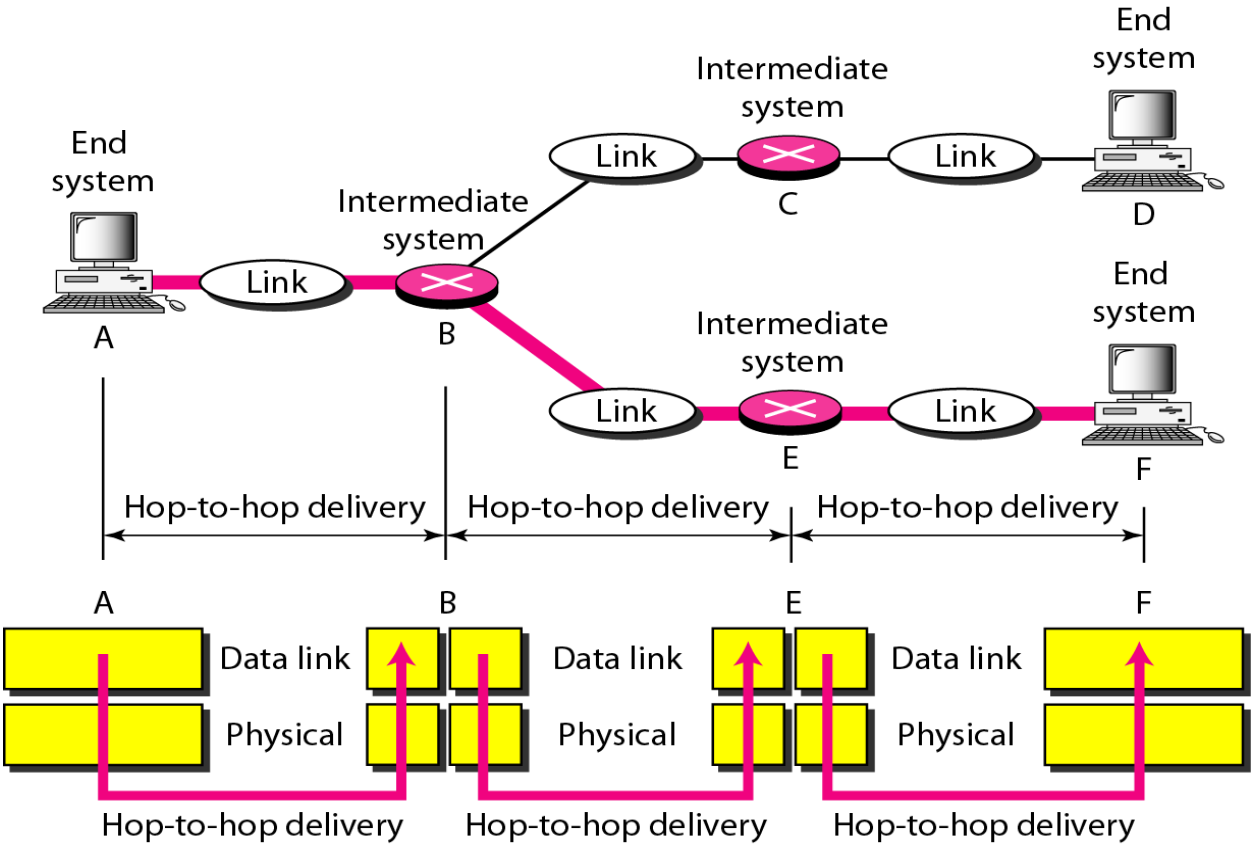
Another issue that arises in the data link layer (and most of the higher layers as well) is how to keep a fast transmitter from drowning a slow receiver in data. Some traffic regulation mechanism may be needed to let the transmitter know when the receiver can accept more data.

Broadcast networks have an additional issue in the data link layer: how to control access to the shared channel. A special sublayer of the data link layer, the **medium access control** sublayer, deals with this problem.

- It divides the data into number of frames.
- It uses the MAC address for sending frames from one node to other.
- It provides flow control, error control and access control.




**NOTE:- The data link layer is responsible for moving frames from one hop (node) to the next.**



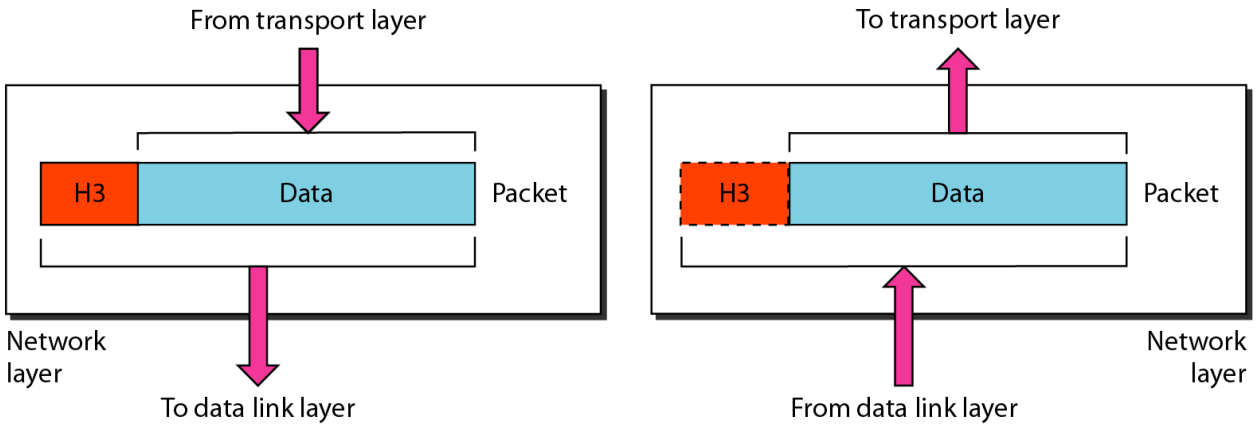
## The Network Layer

The **network layer** controls the operation of the subnet. A key design issue is determining how packets are routed from source to destination. Routes can be based on static tables that are ‘‘wired into’’ the network and rarely changed, or more often they can be updated automatically to avoid failed components. They can also be determined at the start of each conversation, for example, a terminal session, such as a login to a remote machine. Finally, they can be highly dynamic, being determined anew for each packet to reflect the current network load. If too many packets are present in the subnet at the same time, they will get in one another’s way, forming bottlenecks. Handling congestion is also a responsibility of the network layer, in conjunction with higher layers that adapt the load they place on the network. More generally, the quality of service provided (delay, transit time, jitter, etc.) is also a network layer issue.

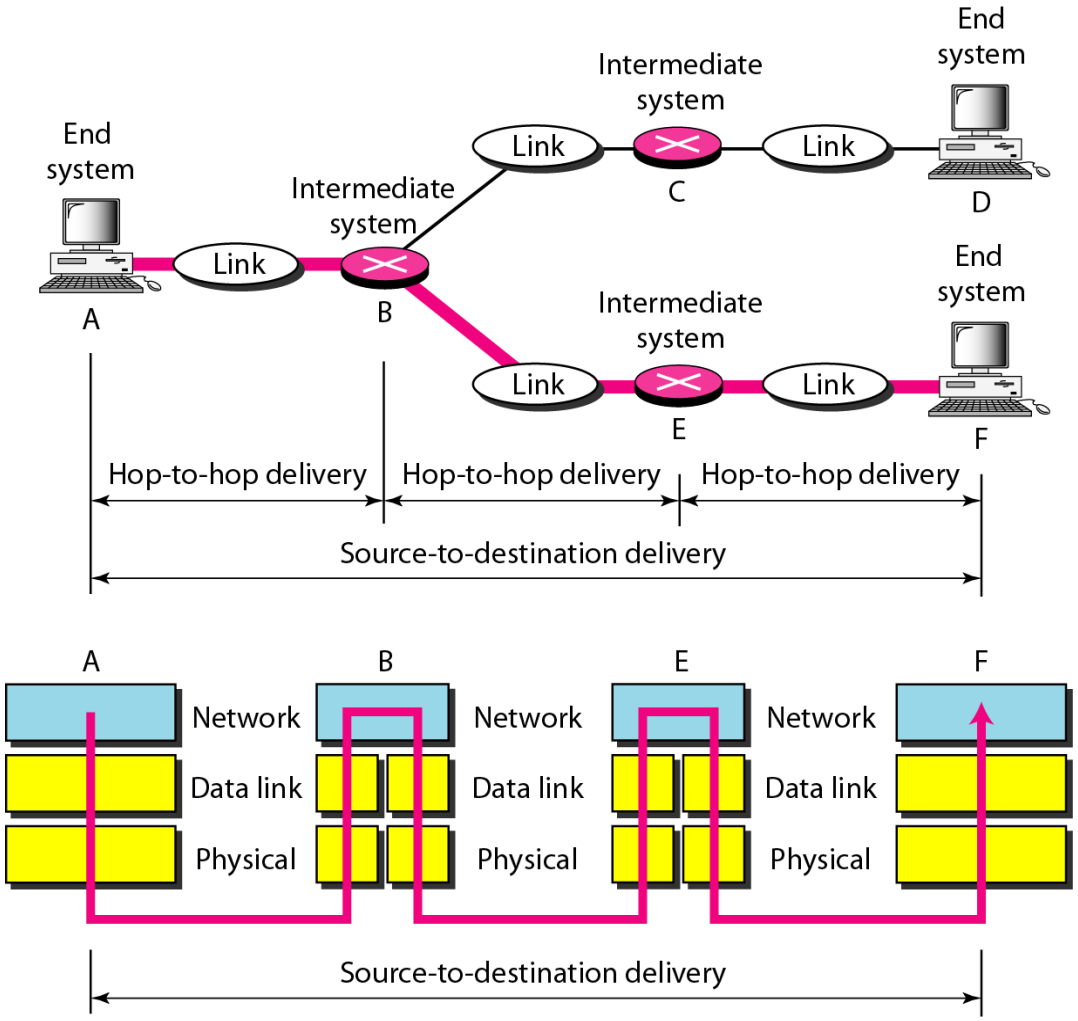
When a packet has to travel from one network to another to get to its destination, many problems can arise. The addressing used by the second network may be different from that used by the first one. The second one may not accept the packet at all because it is too large. The protocols may differ, and so on. It is up to the network layer to overcome all these problems to allow heterogeneous networks to be interconnected.

In broadcast networks, the routing problem is simple, so the network layer is often thin or even nonexistent.

- It divides data into number of packets.
- It uses IP address for routing packets to their destination.
- It provides end to end connection.



**NOTE:- The network layer is responsible for the delivery of individual packets from the source host to the destination host.**




## The Transport Layer

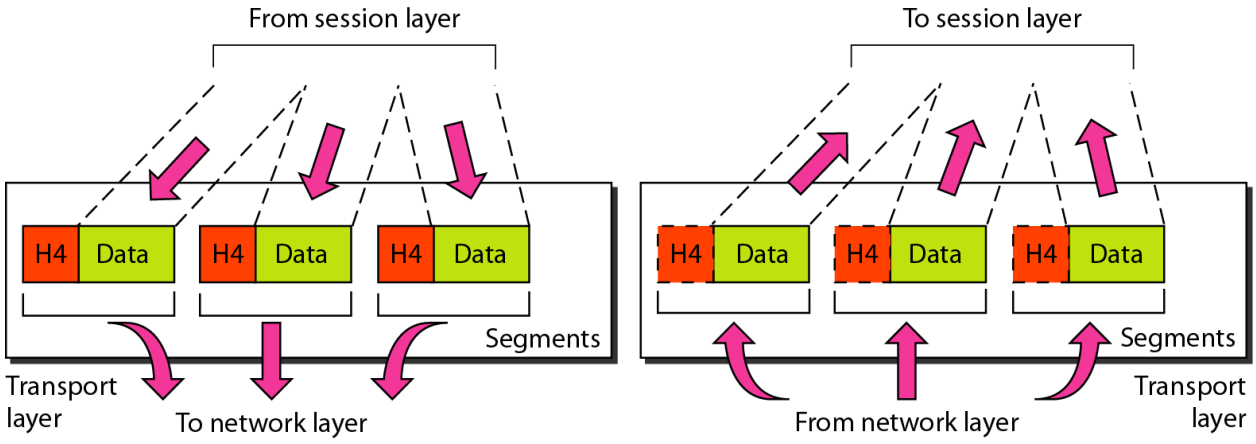
The basic function of the **transport layer** is to accept data from above it, split it up into smaller units if need be, pass these to the network layer, and ensure that the pieces all arrive correctly at the other end. Furthermore, all this must be done efficiently and in a way that isolates the upper layers from the inevitable changes in the hardware technology over the course of time.

The transport layer also determines what type of service to provide to the session layer, and, ultimately, to the users of the network. The most popular type of transport connection is an error-free point-to-point channel that delivers messages or bytes in the order in which they were sent. However, other possible kinds of transport service exist, such as the transporting of isolated

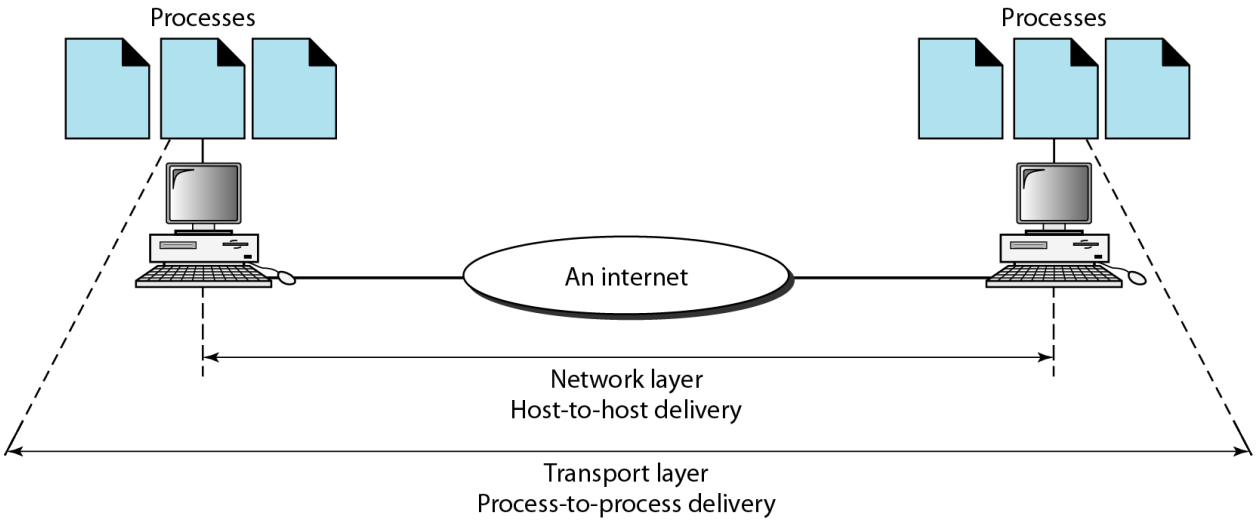
messages with no guarantee about the order of delivery, and the broadcasting of messages to multiple destinations. The type of service is determined when the connection is established. (As an aside, an error-free channel is completely impossible to achieve; what people really mean by this term is that the error rate is low enough to ignore in practice.)

The transport layer is a true end-to-end layer; it carries data all the way from the source to the destination. In other words, a program on the source machine carries on a conversation with a similar program on the destination machine, using the message headers and control messages. In the lower layers, each protocols is between a machine and its immediate neighbors, and not between the ultimate source and destination machines, which may be separated by many routers. The difference between layers 1 through 3, which are chained, and layers 4 through 7, which are end-to-end, is illustrated in Figure above.

- It divides message into segments and also reassemble the segments to create original message.
- It can be either connection-oriented or connectionless.
- It uses service-point address or port address for process-to-process communication.
- Flow control and error control also provided by transport layer.



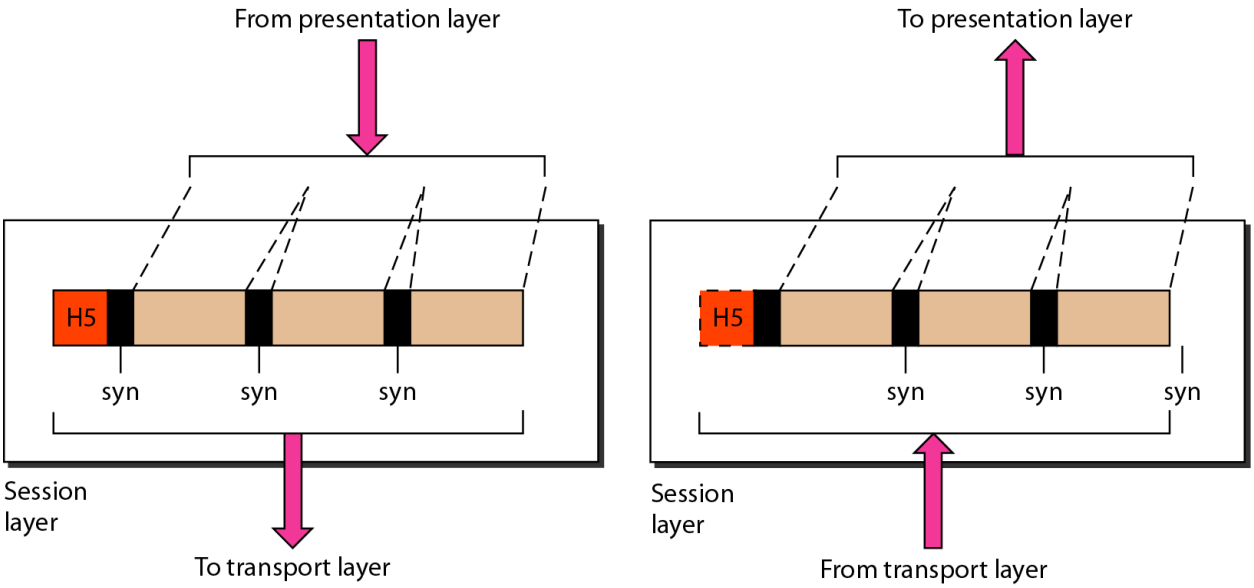

**NOTE:- The transport layer is responsible for the delivery of a message from one process to another.**




## The Session Layer

The session layer allows users on different machines to establish **sessions** between them. Sessions offer various services, including **dialog control** (keeping track of whose turn it is to transmit), **token management** (preventing two parties from attempting the same critical operation simultaneously), and **synchronization** (checkpointing long transmissions to allow them to pick up from where they left off in the event of a crash and subsequent recovery).

- Session Layer establishes, maintains and synchronizes the interaction among communicating systems.

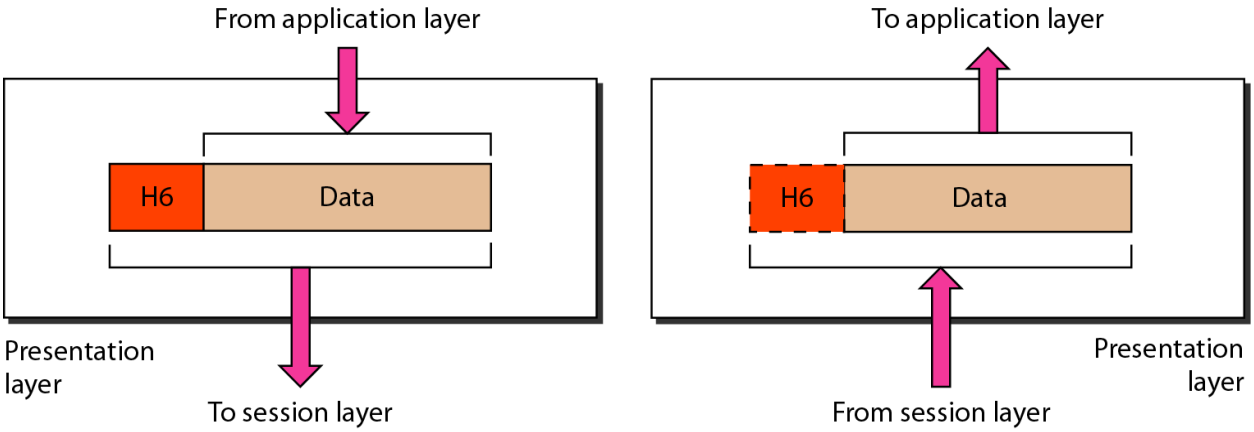



**NOTE:- The session layer is responsible for dialog control and synchronization.**

## The Presentation Layer

Unlike the lower layers, which are mostly concerned with moving bits around, the **presentation layer** is concerned with the syntax and semantics of the information transmitted. In order to make it possible for computers with different internal data representations to communicate, the data structures to be exchanged can be defined in an abstract way, along with a standard encoding to be used ‘‘on the wire.’’ The presentation layer manages these abstract data structures and allows higher-level data structures (e.g., banking records) to be defined and exchanged.

- It is concerned with the syntax and semantics of the Information exchanged between two systems.
- It translates information from text/numeric into bit stream.
- It also encrypts the information for security purpose and compress the information to reduce the number of bits in the information.

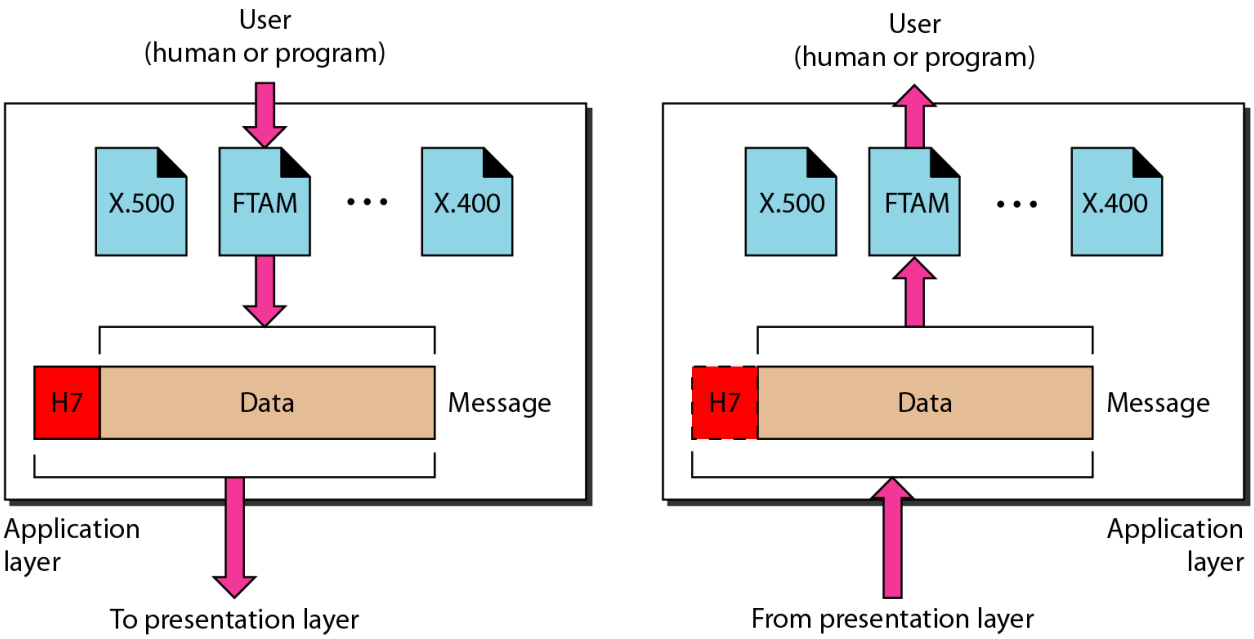



**NOTE:- The presentation layer is responsible for translation, compression, and encryption.**

## The Application Layer

The **application layer** contains a variety of protocols that are commonly needed by users. One widely used application protocol is **HTTP** (**HyperText Transfer Protocol**), which is the basis for the World Wide Web. When a browser wants a Web page, it sends the name of the page it wants to the server hosting the page using HTTP. The server then sends the page back. Other application protocols are used for file transfer, electronic mail, and network news.

- It provides the interface to the end user and supports for services such as Email, file transfer and distributed information service.

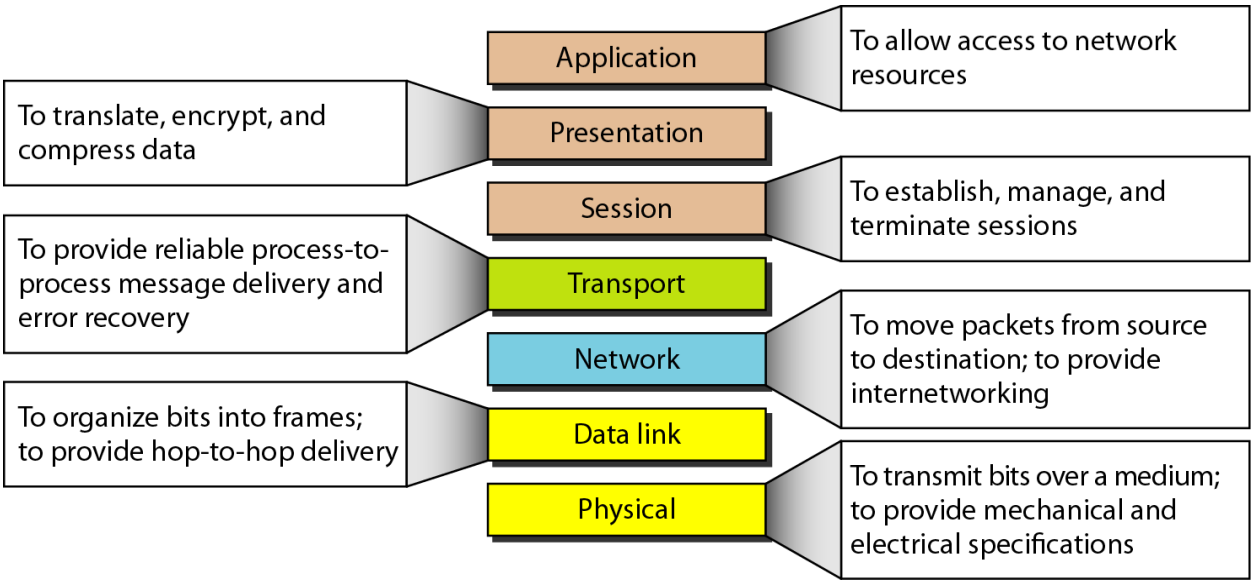



**NOTE:-** **The application layer is responsible for providing services to the user.**

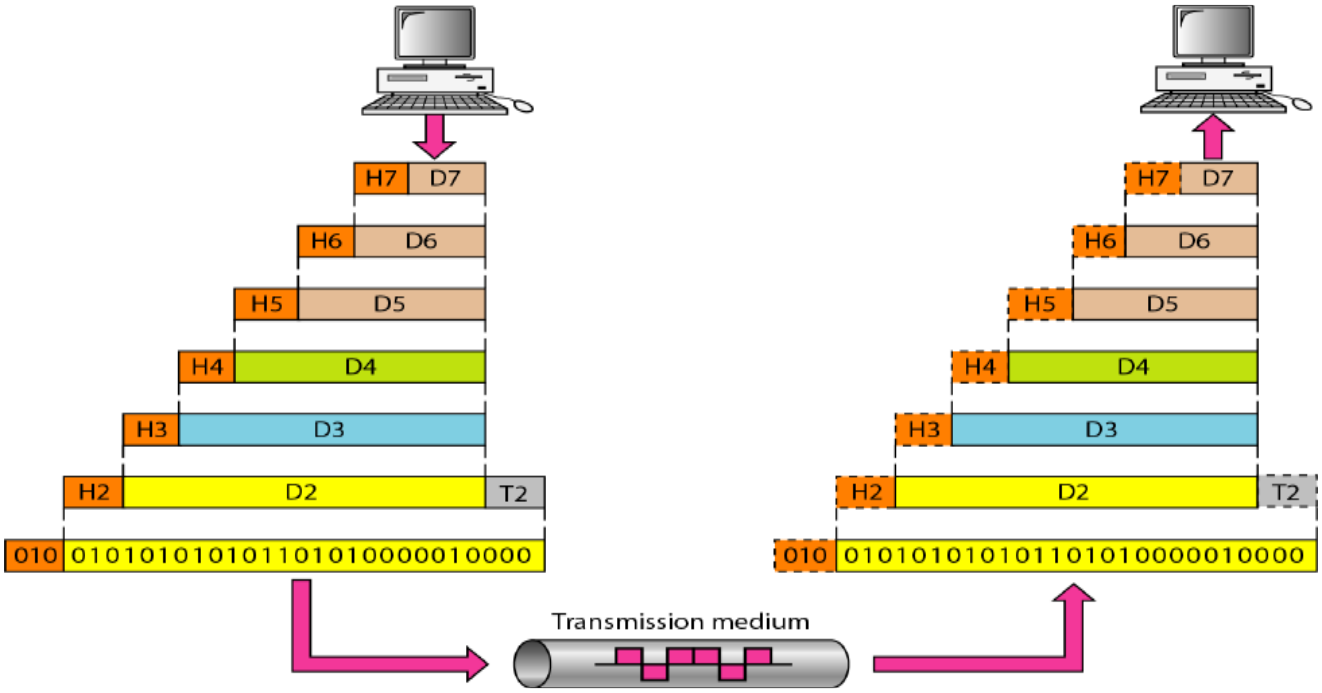
## 8. OSI Model and Protocol stack

| Layer | Protocol | Devices |
|---|---|---|
| **Application** | HTTP, FTP, SMTP,TELNET | |
| **Presentation** | JPG, GIF, MPEG, | |
| **Session** | TCP 3-way Handshaking | |
| **Transport** | TCP, UDP | |
| **Network** | IP, IPX | Router |
| **Data Link** | Ethernet, Token Ring, HDLC | Switch |
| **Physical** | X.21, RS-232, DS, DS3 | Hub, Repeater, Cables |

## 9. Summary of layers




***An exchange using the OSI model***



**Summary:**
**Physical Layer:** How to transmit bits.

**Data Link Layer:** How to transmits frames

**Network:** How to route packets to the node.

**Transport:** How to send packets to the applications.

**Session:** Manage connections.

**Presentation:** Encode/Decode messages, security.

**Application:** Everything else.

## 10. TCP/IP model

- TCP/IP protocol suite was developed before the OSI model.
- TCP/IP is a set of protocols developed to allow cooperating computers to share resources across a network.
- In 1969 the Defense Advanced research projects Agency (DARPA) funded a research and development project to create an experimental packet switching network. This network is called ARPANET.
- In 1975 the ARPANET was converted from an experimental network to an operational network, and the responsibility for administering the network was given to the Defense Communication Agency (DCA).
- The TCP/IP protocols were adopted as Military Standards (MIL STD) in 1983, and all hosts connected to the network were required to convert to the new protocols.
- DARPA funded to implement TCP/IP in Berkely Unix.
- In 1983, the old ARPANET was divided into MILNET and smaller ARPANET. The Internet was used to refer to the entire network; MILNET and ARPANET.

- Unlike OSI reference model, TCP/IP reference model has only 4 layers. They are,

1. Host-to-Network Layer
2. Internet Layer
3. Transport Layer
4. Application Layer

## Host-to-Network Layer:

The TCP/IP reference model does not really say much about what happens here, except to point out that the host has to connect to the network using some protocol so it can send IP packets to it. This protocol is not defined and varies from host to host and network to network.

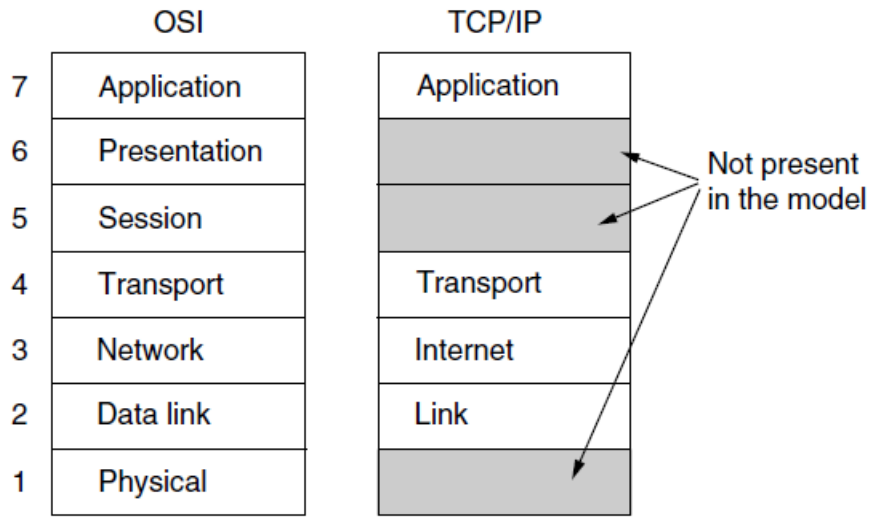
## Internet Layer:

This layer, called the internet layer, is the linchpin that holds the whole architecture together. Its job is to permit hosts to inject packets into any network and have they travel independently to the destination (potentially on a different network). They may even arrive in a different order than they were sent, in which case it is the job of higher layers to rearrange them, if in-order delivery is desired. Note that "internet" is used here in a generic sense, even though this layer is present in the Internet. The internet layer defines an official packet format and protocol called IP (Internet Protocol). The job of the internet layer is to deliver IP packets where they are supposed to go. Packet

routing is clearly the major issue here, as is avoiding congestion. For these reasons, it is reasonable to say that the TCP/IP internet layer is similar in functionality to the OSI network layer. Fig. shows this correspondence.

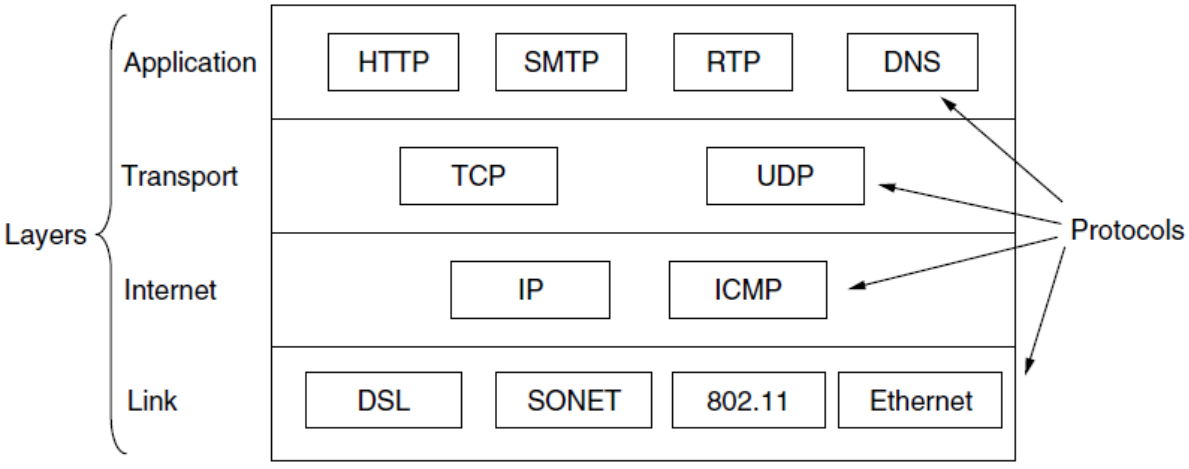
## The Transport Layer:

The layer above the internet layer in the TCP/IP model is now usually called the transport layer. It is designed to allow peer entities on the source and destination hosts to carry on a conversation, just as in the OSI transport layer. Two end-to-end transport protocols have been defined here. The first one, TCP (Transmission Control Protocol), is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine in the internet. It fragments the incoming byte stream into discrete messages and passes each one on to the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream. TCP also handles flow control to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle.




The TCP/IP reference model.

The second protocol in this layer, UDP (User Datagram Protocol), is an unreliable, connectionless protocol for applications that do not want TCP's sequencing or flow control and wish to provide their own. It is also widely used for one-shot, client-server-type request-reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video. The relation of IP, TCP, and UDP is shown in Figure bellow. Since the model was developed, IP has been implemented on many other networks.




The TCP/IP model with some protocols we will study.

### The Application Layer

The TCP/IP model does not have session or presentation layers. No need for them was perceived. Instead, applications simply include any session and presentation functions that they require. Experience with the OSI model has proven this view correct: these layers are of little use to most applications.

On top of the transport layer is the **application layer**. It contains all the higher- level protocols. The early ones included virtual terminal (TELNET), file transfer (FTP), and electronic mail (SMTP). Many other protocols have been added to these over the years. include the Domain Name System (DNS), for mapping host names onto their network addresses, HTTP, the protocol for fetching pages on the World Wide Web, and RTP, the protocol for delivering real-time media such as voice or movies.

## 11. Advantages of TCP/IP

Open protocol standards, freely available and developed independently from any specific computer hardware or operating system. A common addressing scheme which is enable to connect the most widely used networks. It may use any protocols. It connects dissimilar systems. It provides client/server framework. It provides access to the Internet

## 12. A Comparison of the OSI and TCP Reference Models:

The OSI and TCP/IP reference models have much similarity which are:

1. Both are based on the concept of a stack of independent protocols.
2. Also the Functionality of the layers is roughly similar. For example, in both models the layers up through and Including the Transport layer are there to provide an end-to-end network-independent transport service to processes wishing to communicate.

And the Differences are:

1. Three concepts are central to the OSI model:
   A. Services.
   B. Interfaces.
   C. Protocols.

OSI model make the distinction between these three concepts explicit. Each layer performs some services for the layer above it. The TCP/IP model did not originally clearly distinguished between services, interfaces, and protocols.

2. As a consequence, the protocols in me OSI model are better hidden than TCP/IP model.
3. OSI reference model was devised before the protocols were invented. This ordering means that the model was not biased toward one. With the TCP/IP the reverse was true: the protocols came first, and the model was really just a description of the existing protocols. There was no problem with the protocols fitting the model. They fit perfectly.
4. Another difference is in the area of connectionless versus connection oriented communication. The OSI model supports both connectionless and connection-oriented communication in the network layer, but only connection-oriented communication in the transport layer. The TCP/IP model has only one mode in the network layer (connectionless) but supports both modes in the transport layer.
5. TCP/IP has four layers where OSI have seven layers.

| S. No | Comparison Parameter | Connection-oriented Service | Connection Less Service |
|---|---|---|---|
| 1. | Related System | It is designed and developed based on the telephone system. | It is service based on the postal system. |
| 2. | Definition | It is used to create an end to end connection between the senders to the receiver before transmitting the data over the same or different network. | It is used to transfer the data packets between senders to the receiver without creating any connection. |
| 3. | Virtual path | It creates a virtual path between the sender and the receiver. | It does not create any virtual connection or path between the sender and the receiver. |

| | | | |
|---|---|---|---|
| 4. | Authentication | It requires authentication before transmitting the data packets to the receiver. | It does not require authentication before transferring data packets. |
| 5. | Data Packets Path | All data packets are received in the same order as those sent by the sender. | Not all data packets are received in the same order as those sent by the sender. |
| 6. | Bandwidth Requirement | It requires a higher bandwidth to transfer the data packets. | It requires low bandwidth to transfer the data packets. |
| 7. | Data Reliability | It is a more reliable connection service because it guarantees data packets transfer from one end to the other end with a connection. | It is not a reliable connection service because it does not guarantee the transfer of data packets from one end to another for establishing a connection. |
| 8. | Congestion | There is no congestion as it provides an end-to-end connection between sender and receiver during transmission of data. | There may be congestion due to not providing an end-to-end connection between the source and receiver to transmit of data packets. |
| 9. | Examples | Transmission Control Protocol (TCP) is an example of a connection-oriented service. | User Datagram Protocol (UDP), Internet Protocol (IP), and Internet Control Message Protocol (ICMP) are examples of connectionless service. |