Chapter Two

The OSI Model

The International Standards Organization **(ISO)** is a multinational body dedicated to worldwide agreement on international standards (Established in 1947). An ISO standard that covers all aspects of network communications is the **O**pen **S**ystems **I**nterconnection **(OSI)** model. 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 open communication between different systems without requiring, changes to the logic of the underlying hardware and software. The OSI is not a protocol, it is a model for understanding and designing a network architecture.

2.1- The Model

The Open Systems Interconnection model is a layered framework for the design of network system that allows for communication across all types of computer systems. It consists of *seven* separate but related layers, each of which defines a segment of the process of moving information across a network: ***physical*** (layer 1), ***data link*** (layer 2), ***network*** (layer 3), ***transport*** (layer 4), ***session*** (layer 5), ***presentation*** (layer 6), and ***application*** (layer 7). Each layer defines a family of functions distinct from those of the other layers. By defining and localizing functionality in this fashion, the designers created an architecture that is both comprehensive and flexible (See Figure below).

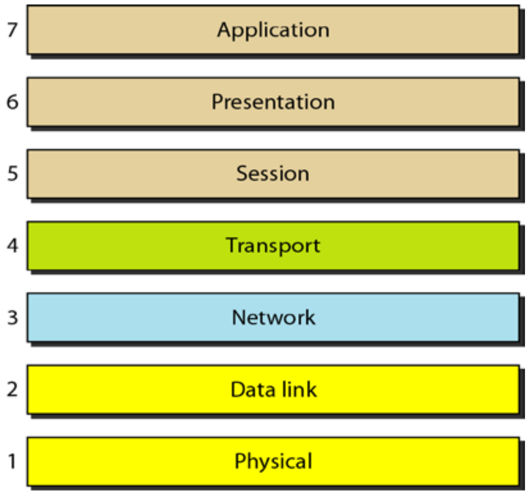
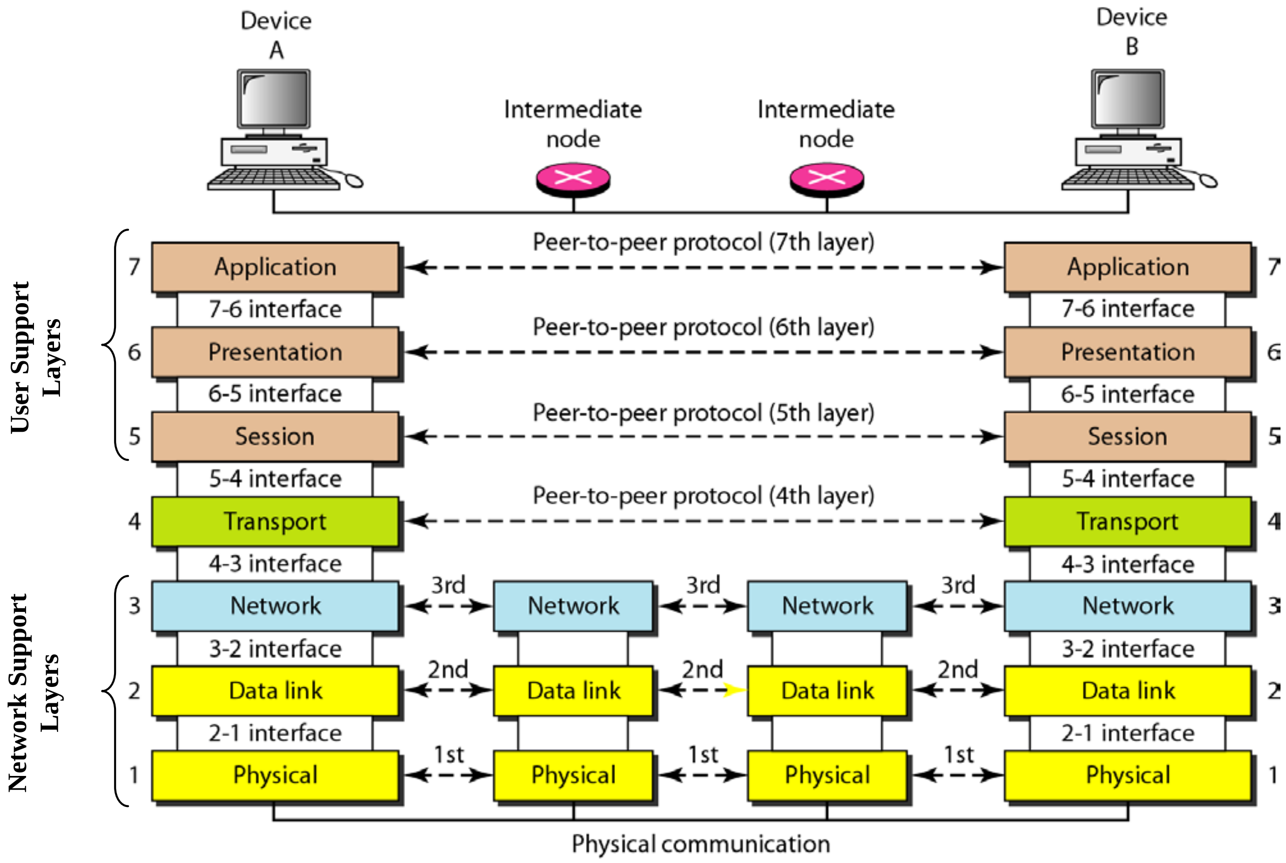


Figure (2.1) shows the layers involved when a message sent from device **A** to device **B**. As the message travels from A to B, it may pass through many intermediate nodes. The intermediate nodes usually involve only the first three layers of the OSI model.

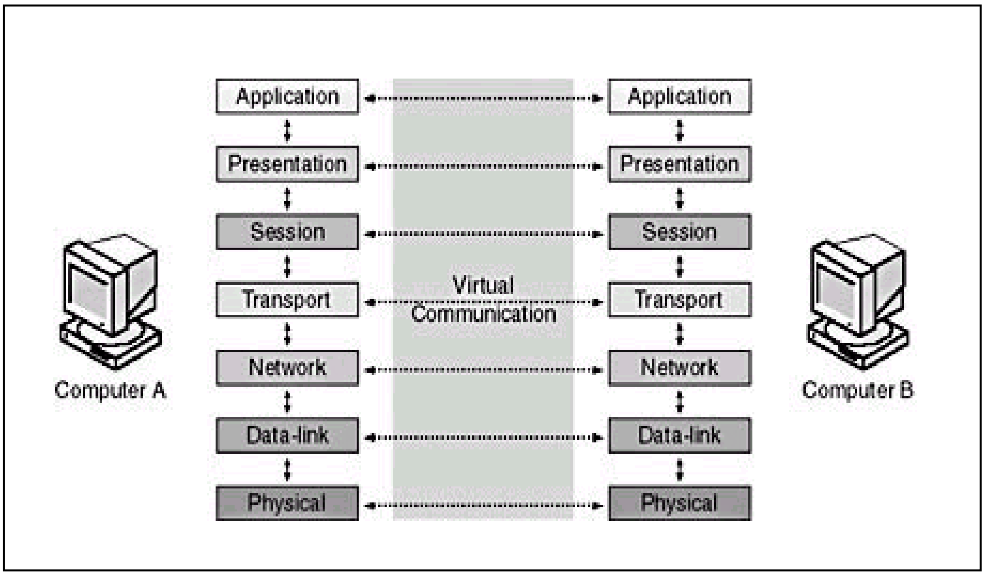


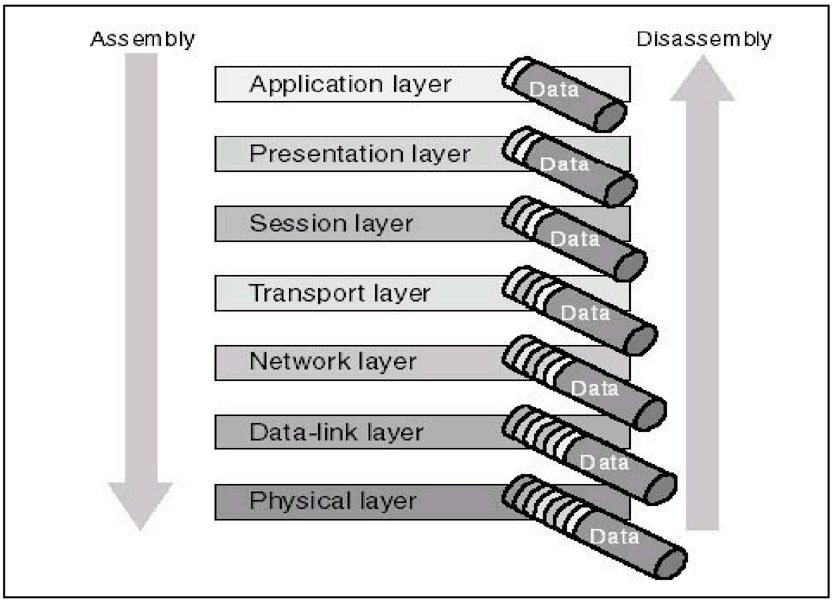
Peer-to-Peer Processes

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 ***protocol.***

The processes on each machine that communicate at a given layer called ***peer-to-peer processes***. Each layer in the sending machine **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. This information added in the form of ***headers*** or ***trailers*** (control data appended to the beginning or end of a data parcel). A trailer added at layer 2 only.

At layer 1, the entire package converted to a form that can be transferred to the receiving machine. **At the receiving** machine, the message is unwrapped layer by layer; with each process receiving and removing the data meant for it (See the two figures below).

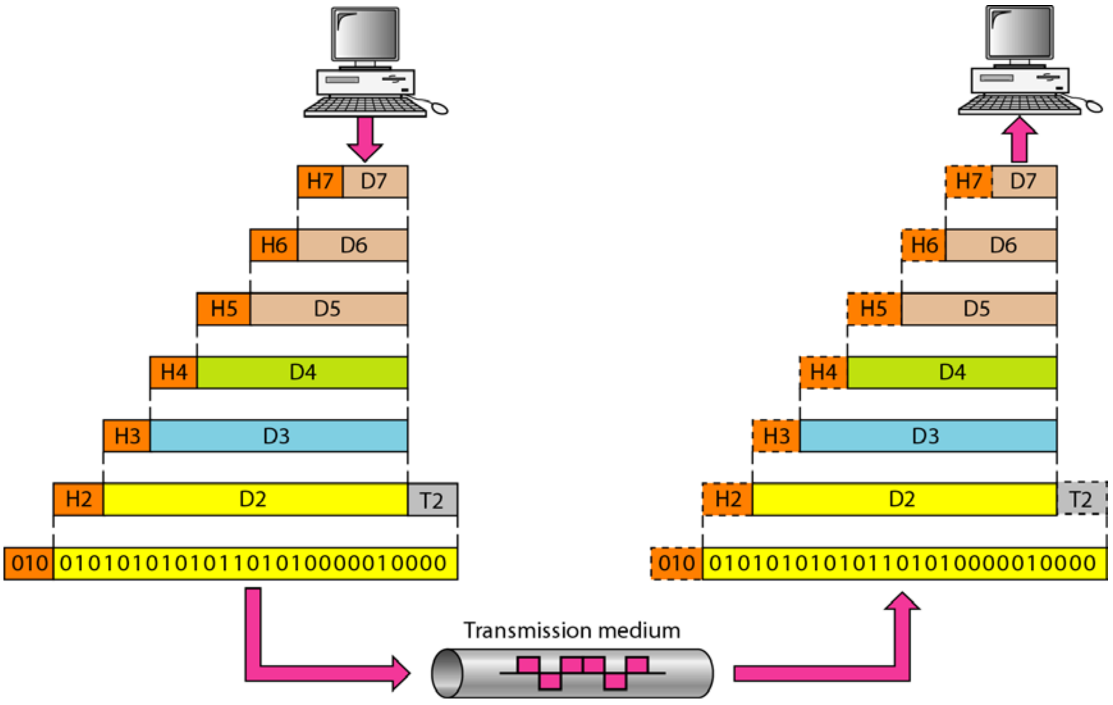




Interfaces between Layers

The passing of the data and network information *down* through the layers of the sending machine and back *up* through the layers of the receiving machine is made possible by an ***interface*** between each pair of adjacent layers. *Each interface defines what information and services a layer must provide for the layer above it.*

The overall view of the OSI layer shown in figure below

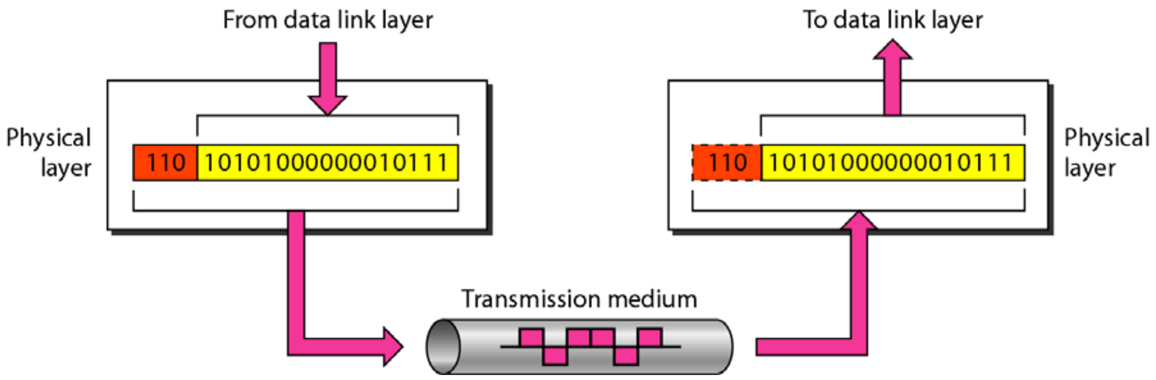


2.2- Functions of the Layers

(1) Physical Layer

The physical layer coordinates the functions required to transmit a bit streams over a physical medium. It **deals with** the mechanical and electrical specifications of the primary connections, such as cables, connectors, and signalling options that physically link two nodes on a network.

This first layer receives a data unit from the second layer and puts it into a format capable of being carried by a communications link. It oversees the changing of a bit streams into electromagnetic signals, and their transmission onto and across a medium.

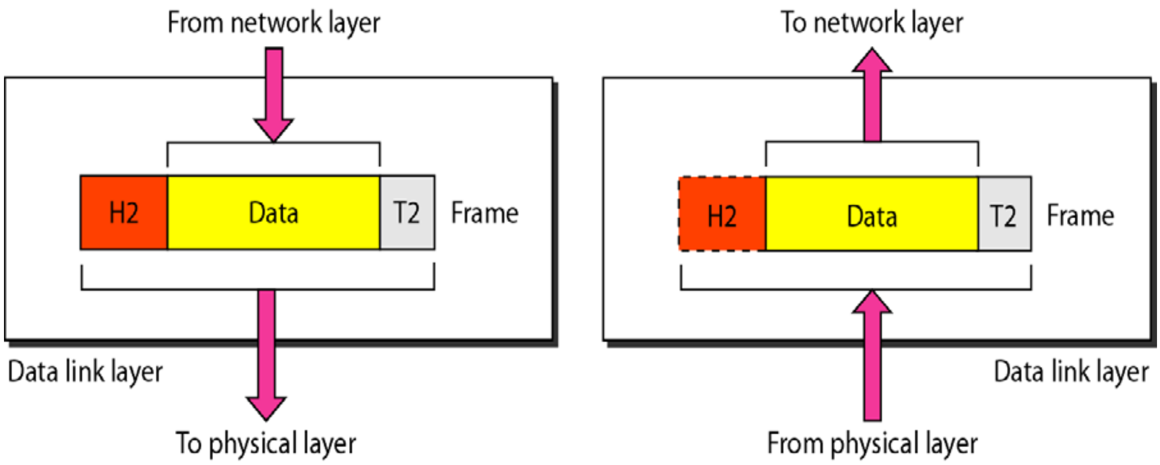


This task requires a number of considerations:

- **Line configuration**: How can two or more devices are linked physically? Are transmission lines to be shared or limited to use between two devices? Is the line available or not?
- **Data transmission mode:** It does transmission flow one-way or both ways between two connected devices. Or does it alternate?
- **Topology:** How network devices arranged? Do they pass data directly to each other or through an intermediary? And by what paths?
- **Signals:** What type of signals is useful for transmitting information?
- **Encoding:** How are bits (0s and 1s) to be represented by available signalling systems? How data are represented by signals?
- **Interface**: What information must be shared between two closely linked devices to enable and facilitate communication? What is the most efficient way to communicate that information?
- **Medium**: What is the physical environment for the transmission of data?
- **Multiplexing:** Using a single physical line to carry data between many devices at the same time.

(2) Data link Layer

The data link layer is **responsible for** delivering data units (groups of bits) from one station to the next *without errors*. It accepts a data unit from the third layer and **adds** meaningful bits to the beginning (header) and end (trailer) that contain addresses and of control information. A data unit with this additional information called a ***Frame***.

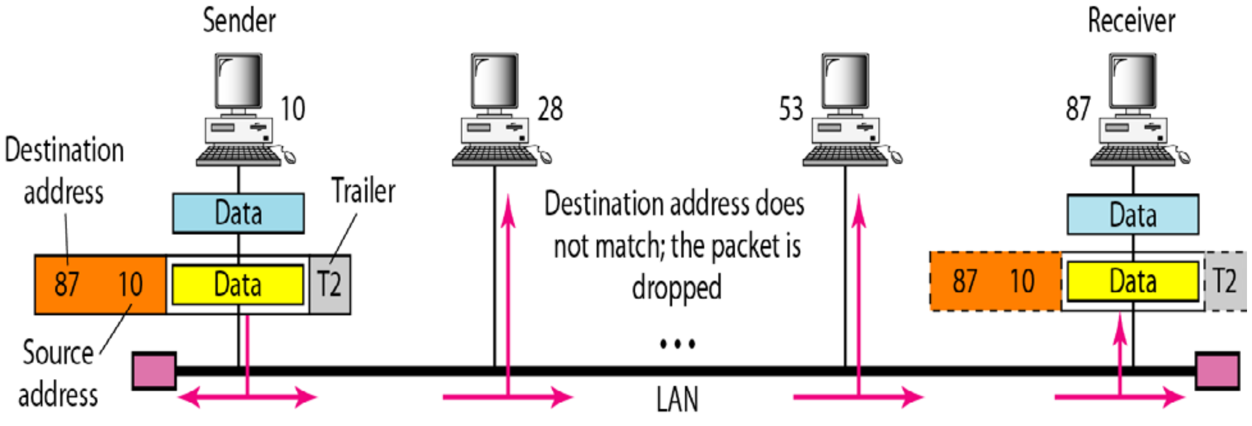


<u>The responsibilities of the data link layer include the following:</u>

- **Node-to-node delivery:** The data link layer is responsible for node-to-node delivery.
- **Physical Addressing:** Headers and trailers added at this layer include the *physical address* of the most recent node and the next intended node.
- **Access control:** When two or more devices connected to the same link, the data-link layer protocols are necessary to determine which device has control over the line at any given time.
- **Flow control:** To avoid overwhelming the receiver, the data link layer regulates the amount of data that can be transmitted at one time. It adds identifying numbers to enable the receiving node to control the ordering of the frames.
- **Error handling:** Data link layer protocols provide for data recovery, usually by having the entire frame retransmitted.

<u>Example:</u>

In Figure below a node with physical address 10 sends a frame to a node with physical address 87. The two nodes connected by a link. At the data link level, this frame contains physical (link) addresses in the header. These are the only addresses needed. The rest of the header contains other information needed at this level. The trailer usually contains extra bits needed for error detection.



(3) Network Layer

The network layer is **responsible for** the *source-to-destination* delivery of a packet across multiple network links. Whereas the data link layer oversees *node-to-node* delivery, the network layer **ensures** that each **packet** gets from its point of origin to its final destination successfully and efficiently.

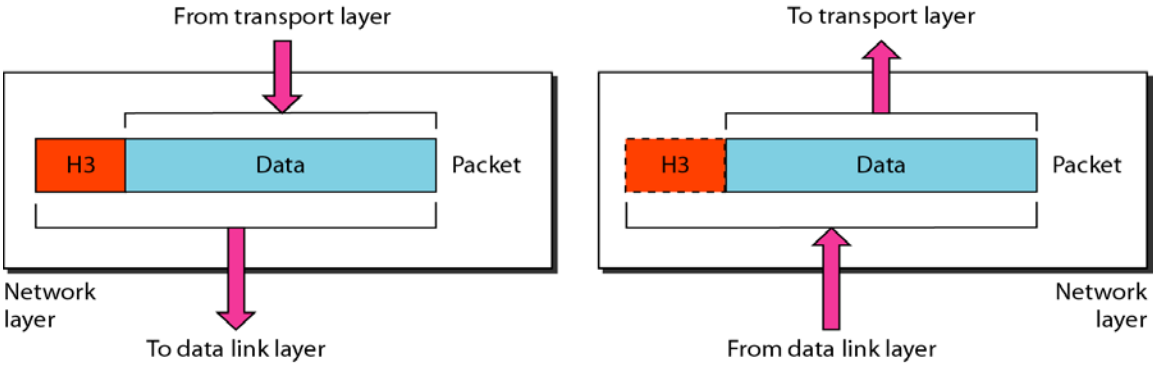
To make such end-to-end delivery possible, the network layer **provides** two related services: *switching and routing***.**

Switching refers to temporary connections between physical links resulting in longer links for network transmission.

Routing means selecting the best path for sending a packet from one point to another, when more than one path is available.

Routing and switching require the addition of a header that includes, among other, information, the source and destination addresses of the packet. These addresses are different from the physical (node) addresses included in the data link header; it is known as a ***logical address***.

Physical addresses are of current to next node only; they are changes as a frame move from one node to another. Whereas the logical addresses are those for the original source to the final destination, and do not changed during packet transmission.

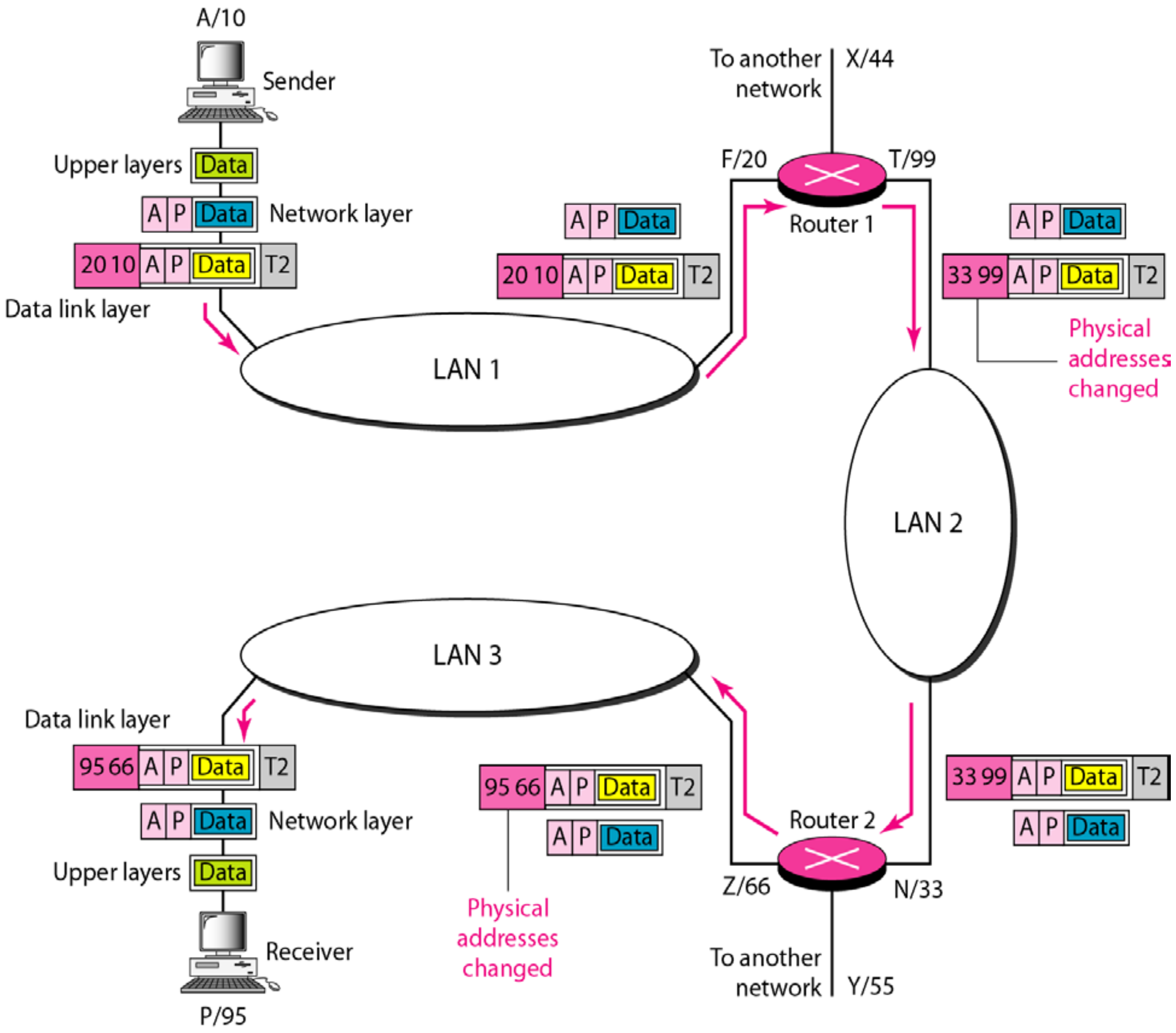


Specific responsibilities of the network layer include the following:

- **Source-to-destination delivery:** Moving a packet (best effort) from its point of origin to its intended destination across multiple network links.,
- **Logical addressing:** Inclusion of the source and destination addresses in the header.
- **Routing:** Deciding which of multiple paths a packet should take.
- **Address transformation:** Interpreting logical addresses to find their physical equivalents.

Example

Now imagine that in Figure below we want to send data from a node with network address **A**, physical address 10, located on one local area network, to a node with a network address **P**, physical address 95, located on another local area network. Because the two devices are located on different networks, we cannot use link addresses only. What we need here are universal addresses that can pass through the boundaries of local area networks. The network (logical) addresses have this characteristic. The packet at the network layer contains the logical addresses, which remain the same from the original source to the destination (A and P. respectively, in the figure). They will not change when we go from network to network. However, the physical addresses will change when the packet moves from one network to another. The box with the R is a router (intemetwork device), which we will discuss later

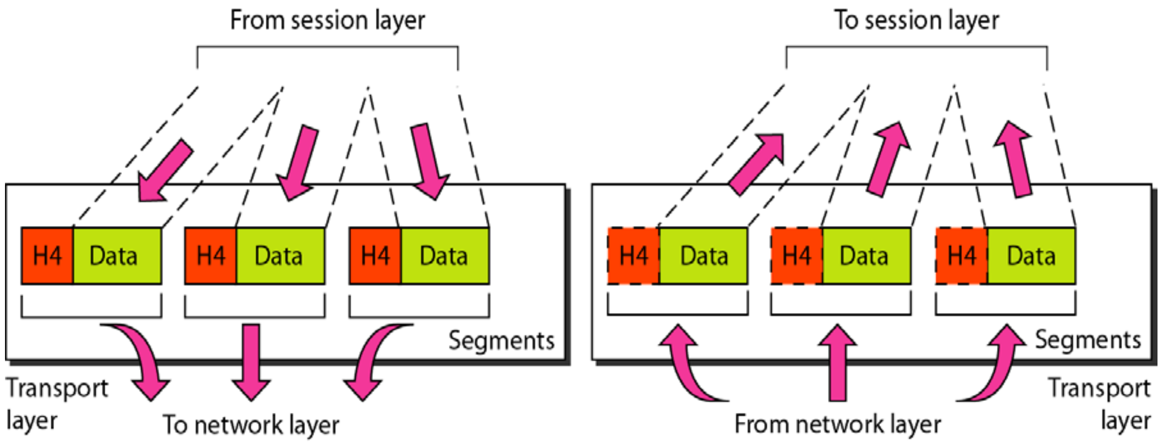


(4) Transport Layer

The transport layer is **responsible for** source -to-destination (end-to-end) delivery of the *entire message*. Whereas the network layer oversees end-to-end delivery of *individual packet*s, 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.

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 application on one computer to a specific application on the other. The transport layer header must therefore include a type of address called a service point address (also called a ***port address*** or socket address). The network layer gets each packet to the correct computer; the transport layer gets the entire message to the correct application on that computer.

As the transport layer receives the message from session layer it divides the message into a transmittable segments, indicating in the header the sequence of the segments so that it can be reassembled upon receipt at the destination.



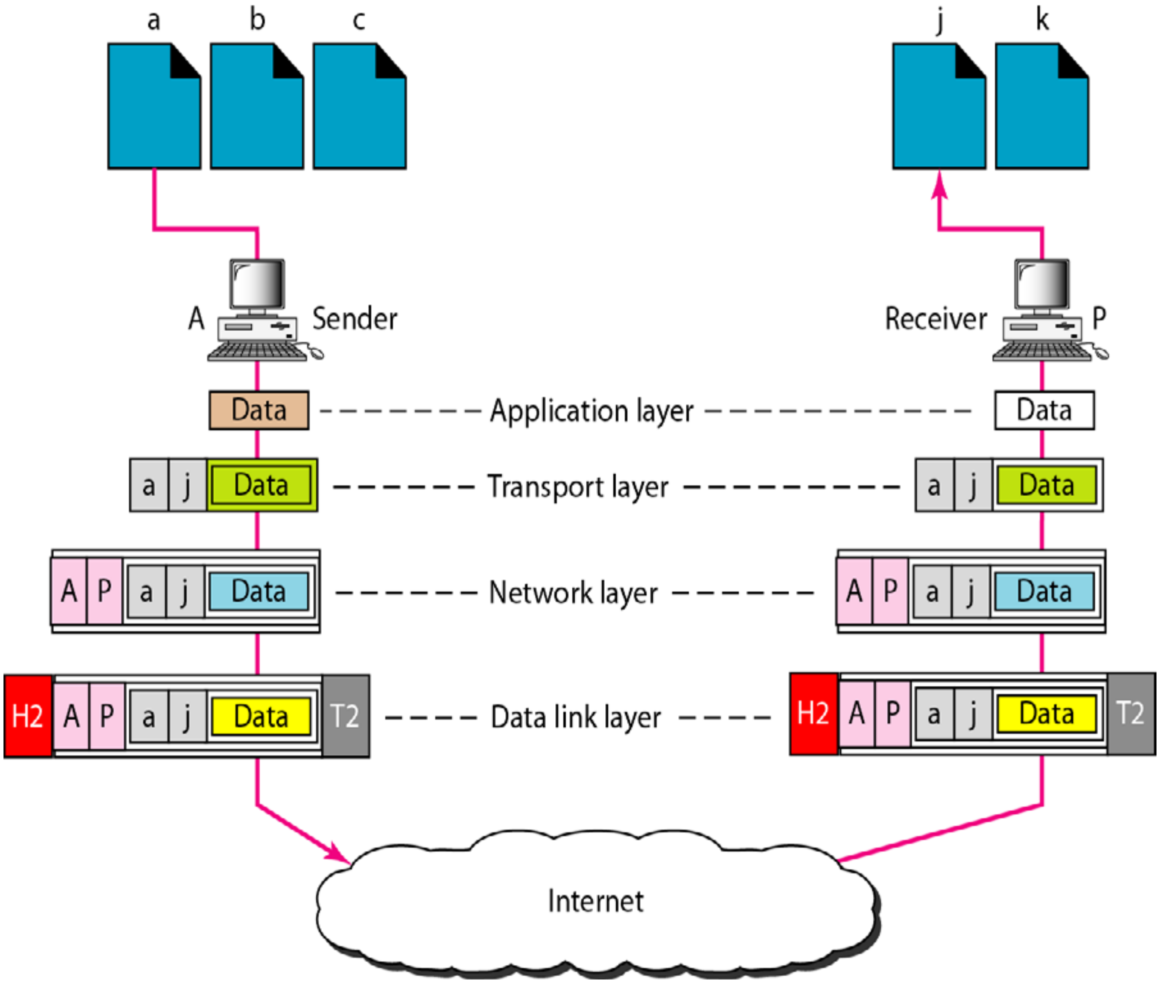
Specific responsibilities of the transport layer include the following:

- ❖ **End-to-end message delivery:** Overseeing the transmission and arrival of all packets of a message at the destination point.
- ❖ **Service-point (port) addressing:** Guaranteeing delivery of a message to the appropriate application on **a** computer running multiple applications.

- **Segmentation and reassembly:** Dividing a message into transmittable segments and marking each segment with a sequence number. These numbers enable the transport layer to reassemble the message correctly at the destination and to identify and replace packets lost in transmission.
- **Connection control:** Deciding whether or not to send all packets by a single path.

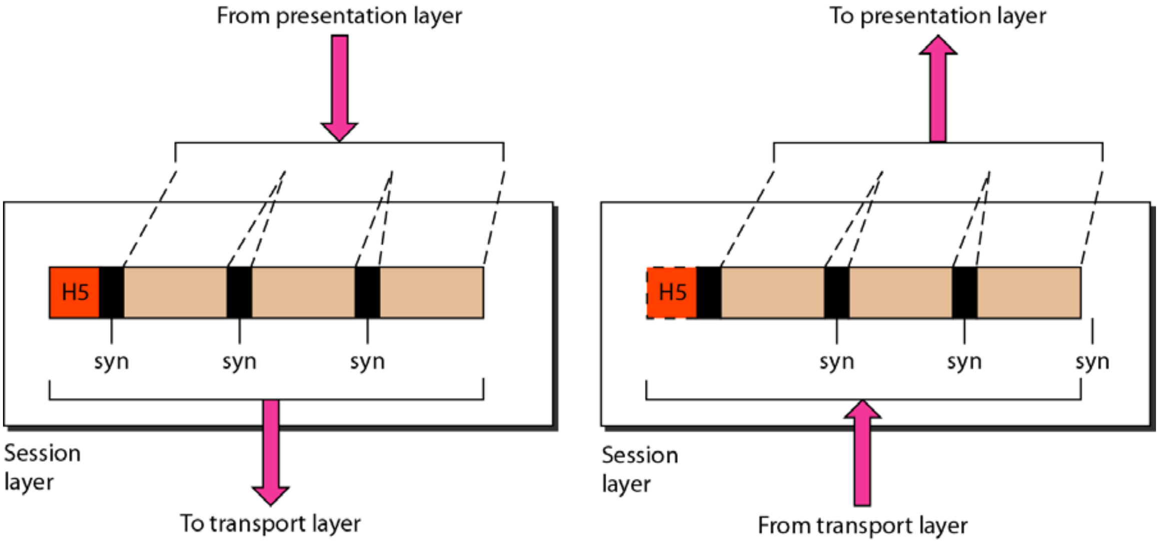
Example:

Figure below shows an example of a transport layer. Data coming from the upper layers have Service-point (port) addresses (a) and (j) (*a* is the address of the sending application, and *j* is the address of the receiving application). Then in the network layer, network addresses (A and P) are added to each packet. The packets may travel on different paths and arrive at the destination either in order or out of order. The packets are delivered to the destination transport layer, which is responsible for removing the network layer headers and the delivery to the upper layers.



(5) ***Session Layer***

The session layer is the network ***dialog controller***. It establishes, maintains, and synchronizes the interaction between' communicating devices. It also **ensures** that each session closes appropriately rather than shutting down abruptly and leaving the user hanging.



Specific responsibilities of the session layer include the following:

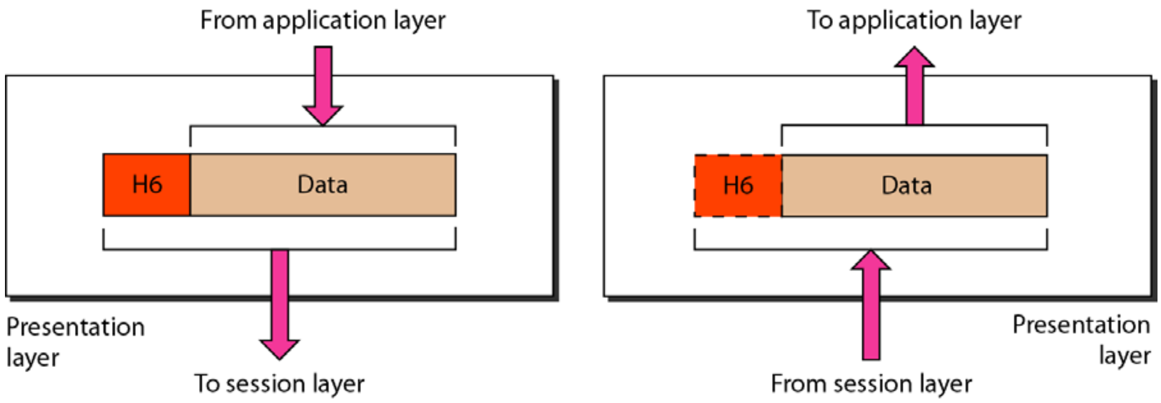
- **Session management:** Dividing a session into sub sessions by the introduction of checkpoints and separating long messages into shorter units called dialog units appropriate for transmission.
- **Synchronization:** Deciding in what order to pass the dialog units to the transport layer and where in the transmission to require confirmation from the receiver.
- **Dialog control:** Deciding who sends, and when.
- **Graceful close:** Ensuring that the exchange has been completed appropriately before the session closes.

Example

A computer needs to update a huge file (e.g. a database). The session layer subdivides the task into different dialog units.

(6) ***Presentation Layer***

The presentation layer **ensures** interoperability among communicating devices. Functions at this layer make it possible for two computers to communicate even if their internal representations of data differ (e.g., when one device uses one type of code and the other uses another). It ***provide*** necessary translation of different control codes and character sets, graphics characters, and so on to allow both devices to understand the same transmission the same way.



Specific responsibilities of the presentation layer include following:

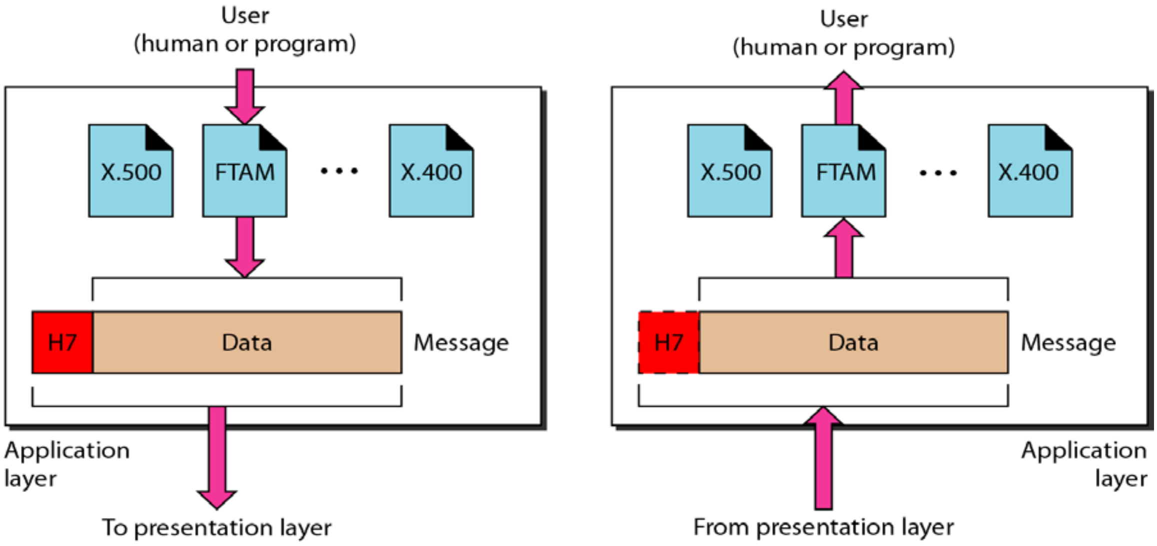
- ❖ **Translation:** Changing the format of a message from that used by the sender into one mutually acceptable for transmission. Then, at the destination, changing that format into the one understood by the receiver.
- ❖ **Encryption:** Encryption and decryption of data for security purposes.
- ❖ **Compression:** Compressing and decompressing data to make transmission more efficient.
- ❖ **Security:** Validating passwords and log-in codes.

Example

The sending station uses an encryption algorithm to protect the data from eavesdropping. The encrypted data are decrypted at the destination presentation layer before being delivered to the application layer.

(7) 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.



Specific services provided by the application layer include the following:

- ❖ **Network virtual terminal:** A software version of a physical terminal. A virtual terminal allows you to log on to a remote host. To do so, the application creates a software emulation of a terminal at the remote host. Your 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 you to log on.
- ❖ **File access, transfer, and management:** Allows a user at a remote computer to access files in another host (to make changes or read data); to retrieve files from remote computer for use in the local computer; and to manage or control files in remote computer at that computer.
- ❖ **Mail services:** Provides the basis for electronic mail forwarding and storage.
- ❖ **Directory services:** Provides distributed database sources and access for global information about various objects.

Example

A user in Beijing, China, wants to send a large proprietary data file to a station in Los Gatos, California. An application service such as FTAM (file transfer and access management) can do the job.

TCP/IP Model

The OSI Model we just looked at is just a reference/logical model. It was designed to describe the functions of the communication system by dividing the communication procedure into smaller and simpler components.

TCP/IP was designed and developed by the Department of Defense (DoD) in the 1960s and is based on standard protocols. It stands for **Transmission Control Protocol/Internet Protocol**. The TCP/IP model is a concise version of the OSI model. It contains four layers, unlike the seven layers in the OSI model.

The number of layers is sometimes referred to as five or four. Here In this article, we'll study five layers. The Physical Layer and Data Link Layer are referred to as one single layer as the 'Physical Layer' or 'Network Interface Layer' in the 4-layer reference.

What Does TCP/IP Do?

The main work of TCP/IP is to transfer the data of a computer from one device to another. The main condition of this process is to make data reliable and accurate so that the receiver will receive the same information which is sent by the sender. To ensure that, each message reaches its final destination accurately, the TCP/IP model divides its data into packets and combines them at the other end, which helps in maintaining the accuracy of the data while transferring from one end to another end.

What is the Difference between TCP and IP?

TCP and IP are different protocols of Computer Networks. The basic difference between TCP (Transmission Control Protocol) and IP (Internet Protocol) is in the transmission of data. In simple words, IP finds the destination of the mail and TCP has the work to send and receive the mail. UDP is another protocol, which does not require IP to communicate with another computer. IP is required by only TCP. This is the basic difference between TCP and IP.

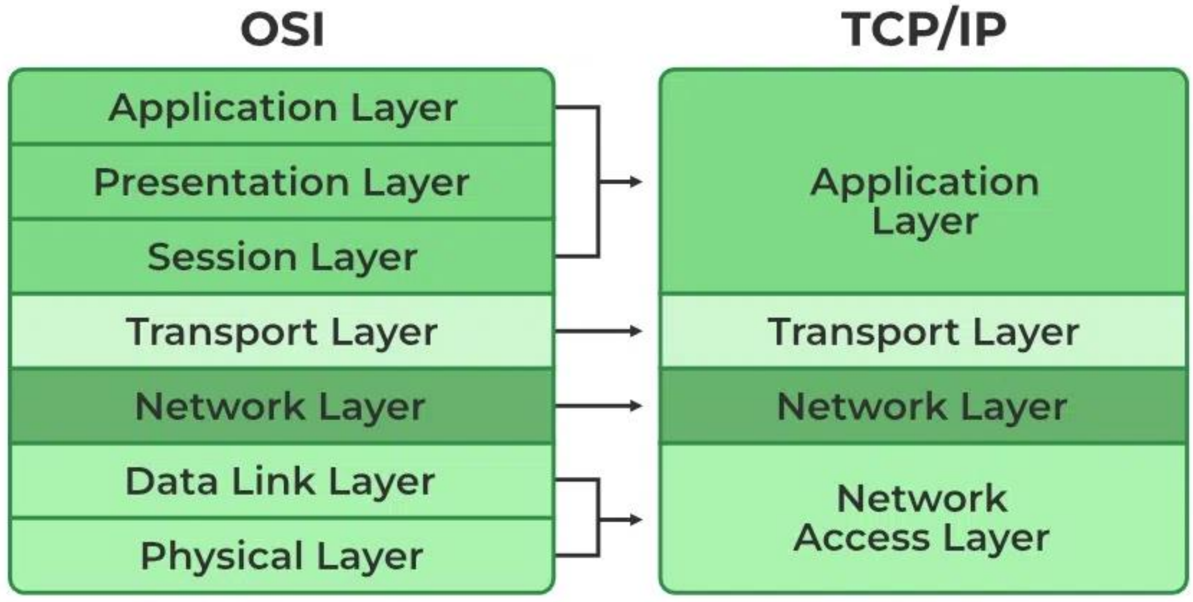
How Does the TCP/IP Model Work?

Whenever we want to send something over the internet using the TCP/IP Model, the TCP/IP Model divides the data into packets at the sender's end and the same packets have to be recombined at the receiver's end to form the same data, and this thing happens to maintain the accuracy of the data. TCP/IP model divides the data into a 4-layer procedure, where the data first go into this layer in one order and again in reverse order to get organized in the same way at the receiver's end.

Layers of TCP/IP Model

1. Application Layer
2. Transport Layer(TCP/UDP)
3. Network/Internet Layer(IP)
4. Data Link Layer (MAC)
5. Physical Layer

The diagrammatic comparison of the TCP/IP and OSI model is as follows:



OSI & TCP/IP