RIMT-CAN
RIMT(Computer Application Notes)

RIMT
Opposite Sarhind Floating Resturant
Mandi Gobindgarh
Tel. 01765-24633.24632
www.rimt.org

# LECTURE NOTES-Computer Network

## BCA-IVth Semester

## Lecture 4

NETWORK TOPOLOGIES

Prepared For RIMT BCA Program Prem Parashar

## NETWORK TOPOLOGIES

The followings are the most commonly used topologies:

### Star Topology

The star topology is  widely used structure for data communications systems. One of the major reasons for its continued use is based on historical precedence. The star network was predominant in the 1960s and early 1970s because it was easy to control. An example star network is shown in figure.

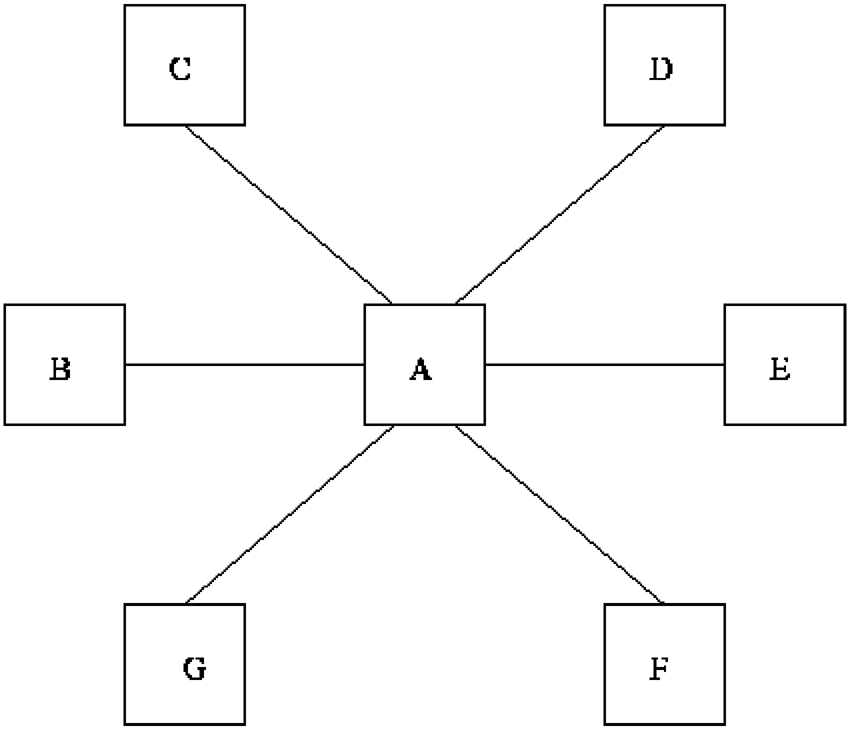


**Figure:**  Example of a star network topology

All traffic emanates from the hub of the star. The central site in figure, labelled A, is in control of all the DTEs attached to it. The central hub is usually a self contained computer and is responsible for routing all traffic to other DTEs and fault isolation. However, like the hierachical structure, this type of network is also prone to bottleneck and failure problems at the central site. Several star networks developed in the 1970s had serious reliability problems because of the centralised nature of the network. Other systems attempted to solve this problem by providing a redundant backup of the hub node.

## Ring Topology

The ring topology is another popular approach to configuring networks. As illustrated in figure, the ring topology is so named because of the circular aspect of the data flow.

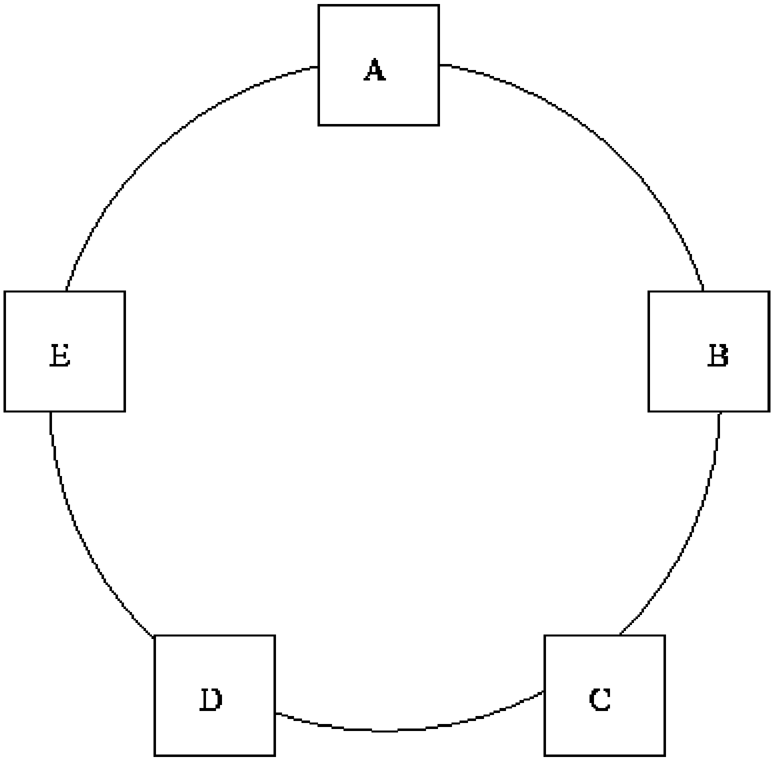


**Figure:** Example of a ring network topology

In most instances data flow is in one direction only, with one single node receiving the transmission and relaying it to the next node in the ring. The ring topology is attractive because it is rarely subjected to the bottlenecks associated with hierarchical and star configurations. Moreover, the logic to implement a ring network is relatively simple.

Each node in the network is tasked with a straightforward job of accepting the data and sending it to the DTE attached to it, or sending it back out onto the ring to the next intermediate node.

However, like all networks, the ring network suffers from some deficiencies. The primary problem is the use of a single channel to tie together all the nodes in the network. If a channel between two nodes fails, then the entire network is lost. To alleviate this problem, some vendors supply ring networks with backup channels and others provide switches that will route data around a failed node. This increases reliability in the event of channel or node failure.

More recently, network suppliers have taken to producing ring networks with two rings, so that the network will still be able to function in the event of channel failure.

## Horizontal Topology (Bus)

The horizontal topology of bus network is shown in figure. This arrangement is common in local area networks.

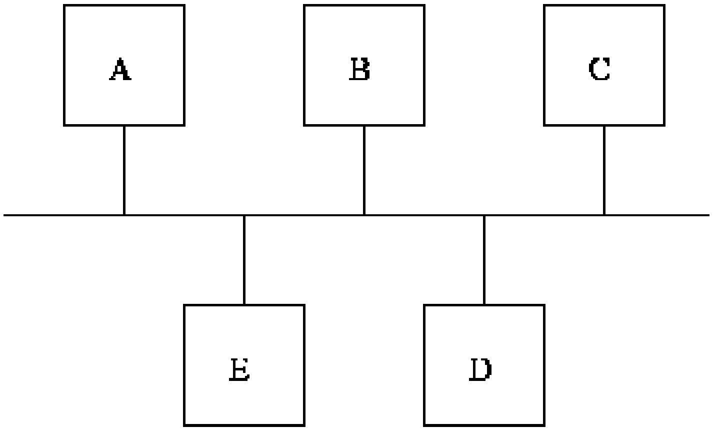


**Figure:** Example of a horizontal network topology

The control of data flow between and among the DTEs is realtively simple as the configuration allows every node to receive every transmission. That is a single DTE broadcasts to every other DTE on the network. The main drawback of this topology stems from the fact that usually only one communications channel exists to service the whole network. If this channel fails then the whole network may fail.

Some vendors provide spare channels for use in the event of channel failure, and others may provide switches that allow the channel to be routed around failed nodes.

## Hierarchical Topology

The hierarchical topology is one of the more common topologies found today. The software to control the network is relatively simple and the topology provides a concentration point for control and error resolution. The DTE at the highest point in the hierarchy usually controls the network. In figure traffic flow among and between the DTEs is initiated by DTE A.

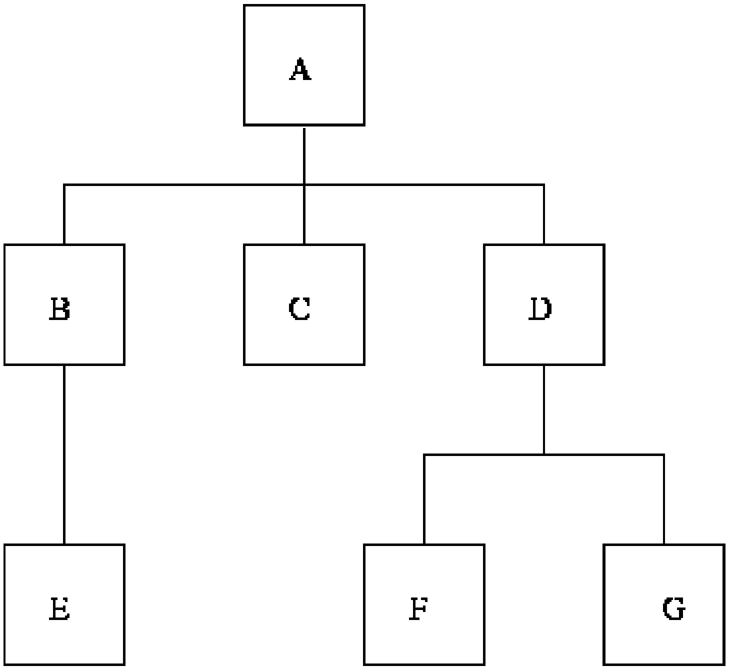


**Figure:** Example of a hierarchical network topology

While this type of network is attractive for its simplicity it does present a potential significant bottleneck problem. In some instances the uppermost DTE will control all the traffic. Not only can this cause a bottleneck, but it can also present reliability problems if this node fails.

This type of network configuration is also known as a ``tree network''.

## Mesh Topology

The mesh topology (figure), has been used more frequently in recent years.

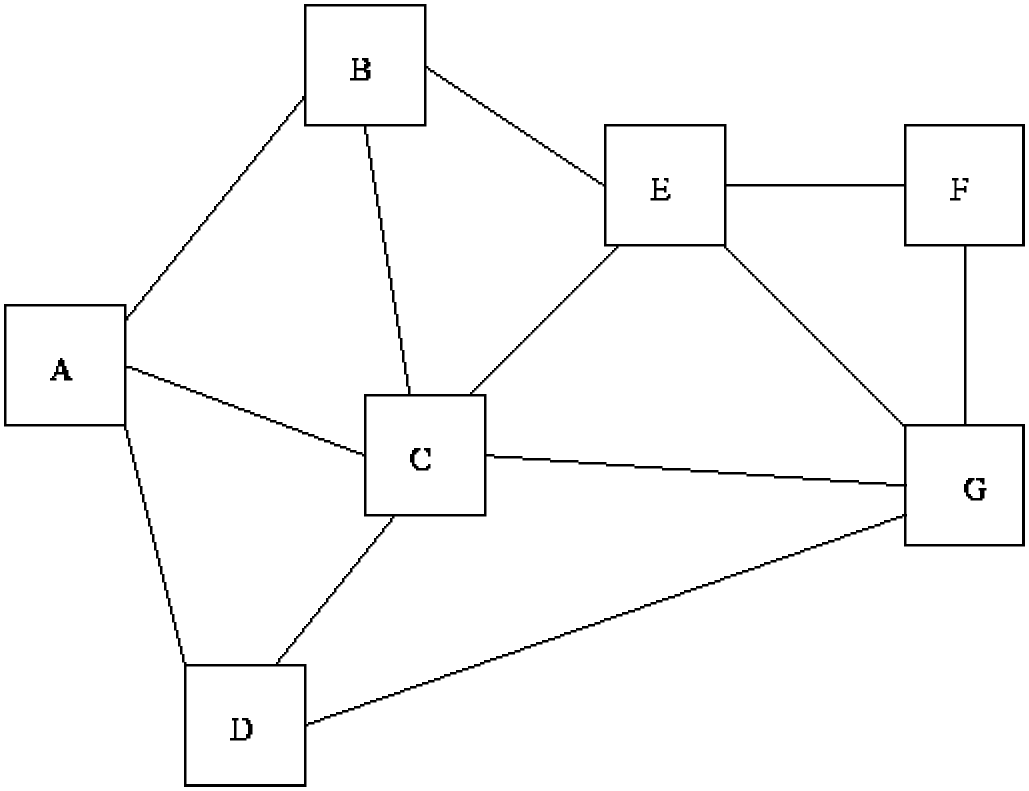


**Figure:** Example of a mesh network topology

Its primary attraction is its relaive immunity to bottlenecks and channel/node failures. Due to the multiplicity of paths from the DTEs and DSEs, traffic can easily be routed around failed or busy nodes. Given that this approach is very expensive in comparison to other topologies, some users will still prefer the reliability of the mesh network to that of others (especially for networks that only have a few nodes that need to be connected together).

## <u>OSI Reference Model</u>

Modern computer networks are designed in a highly structured way. To reduce their design complexity, most networks are organized as a series of layers, each one built upon its predecessor.
The OSI Reference Model is based on a proposal developed by the International Organization for Standardization (ISO). The model is called 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.

The OSI model has seven layers. The principles that were applied to arrive at the seven layers are as follows:

1. A layer should be created where a different level of abstraction is needed.

2. Each layer should perform a well defined function.

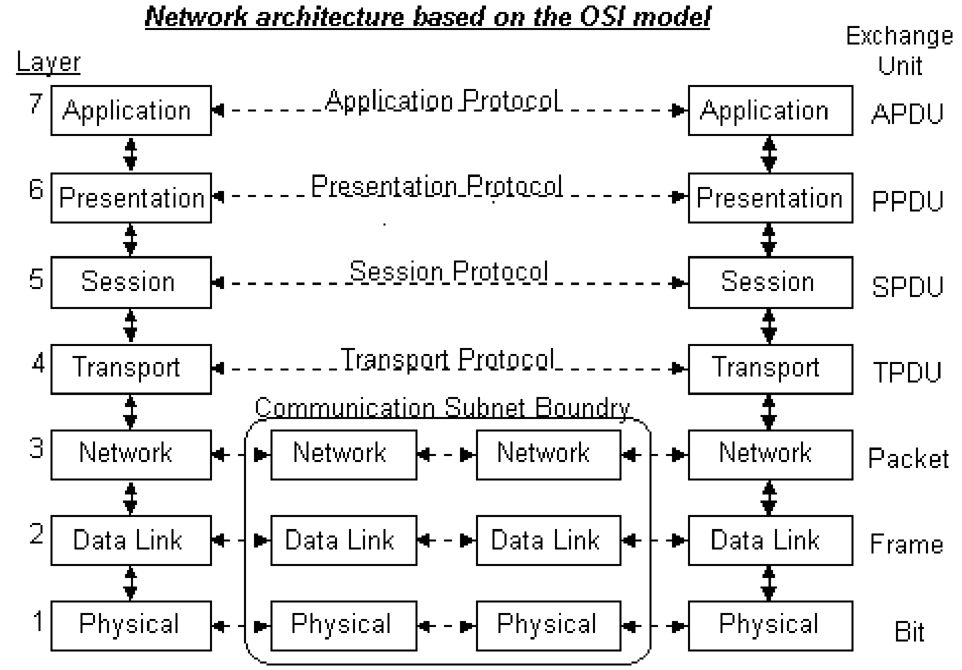
3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols.

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.

## Network Architecture of OSI Model:

In this model, a networking system is divided into layers. Within each layer, one or more entities implement its functionality. Each entity interacts directly only with the layer immediately beneath it, and provides facilities for use by the layer above it. Protocols enable an entity in one host to interact with a corresponding entity at the same layer in a remote host.



The seven layers of the OSI Basic Reference Model are (from bottom to top):

1. The **Physical Layer** describes the physical properties of the various communications media, as well as the electrical properties and interpretation of the exchanged signals. Ex: this layer defines the size of Ethernet coaxial cable, the type of BNC connector used, and the termination method.

2. The **Data Link Layer** describes the logical organization of data bits transmitted on a particular medium. Ex: this layer defines the framing, addressing and checksumming of Ethernet packets.
3. The **Network Layer** describes how a series of exchanges over various data links can deliver data between any two nodes in a network. Ex: this layer defines the addressing and routing structure of the Internet.
4. The **Transport Layer** describes the quality and nature of the data delivery. Ex: this layer defines if and how retransmissions will be used to ensure data delivery.
5. The **Session Layer** describes the organization of data sequences larger than the packets handled by lower layers. Ex: this layer describes how request and reply packets are paired in a remote procedure call.
6. The **Presentation Layer** describes the syntax of data being transferred. Ex: this layer describes how floating point numbers can be exchanged between hosts with different math formats.
7. The **Application Layer** describes how real work actually gets done. Ex: this layer would implement file system operations.

The text also uses a four layer model but with a number of differences from DOD model. These layers are defined as follows:

1. The **Physical Layer** The physical connection between the sender and receiver. This layer transfers a series of electrical, radio, or light signals through the circuit from sender to

receiver. Also specifies the type of connection, and the signals that pass through it. (same as OSI).

2. The **Data Link Layer** Takes the message generated by the network layer and performs three functions before passing the message on the physical layer (same as OSI).

- Controls the physical layer by deciding when to transmit messages over the media.
- Formats the message by indicating where messages start and end, and which part is the address.
- Detects and corrects any errors that have occurred in the transmission of the message

3. The **Network Layer** Takes the message generated by the transport layer and performs three functions before passing them to the data link layer. (same as OSI).

- Translates the destination of the message into an address understood by the network.
- If multiple routes possible, it decides which routes to take.
- Collects message accounting information that can be used to identify how many messages each user has sent and to track errors.

4. The **Transport Layer** Takes the message generated by the application layer and performs three functions before passing them to the network layer. (same as OSI).

- Creates a application level (port) addresses to identify which application is the intended source/destination.
- End-to-End session management (opening and closing channels, congestion control, etc.)
- Breaking messages into groups compatible with the maximum size requirements of the data link layer..

1. The **Application Layer** The application software used by the network user, allows the user to define what message are sent over the network. (same as OSI Session, Presentation, and Application).

## The TCP/IP Reference Model

The TCP/IP reference model is the network model used in the current Internet architecture. It has its origins back in the 1960's with the grandfather of the Internet, the ARPANET. This was a research network sponsored by the Department of Defense in the United States. The following were seen as major design goals:

- ability to connect multiple networks together seamlessly
- ability for connections to remain intact as long as the source and destination machines were functioning
- to be built on flexible architecture

The reference model was named after two of its main protocols, TCP (Transmission Control Protocol) and IP (Internet Protocol).
They choose to build a packet-switched network based on a connectionless internetwork layer.

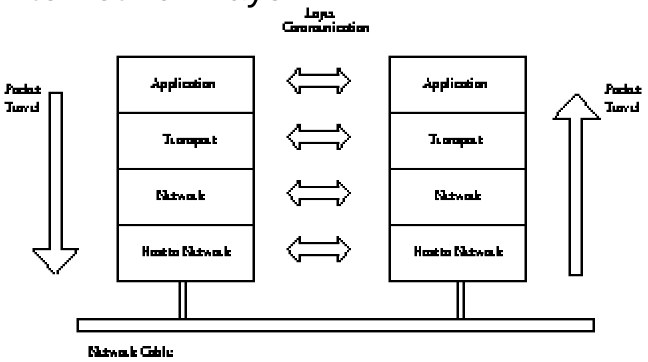


**Figure 2.1:** TCP/IP Network Protocol

A detailed description of the reference model is beyond the scope of this document and project. The basic idea of the networking system is to allow one application on a host computer to talk to another application on a different host computer.
The application forms its request, then passes the packet down to the lower layers, which add their own control information, either a header or a footer, onto the packet. Finally the packet reaches the physical layer and is transmitted through the cable onto the destination host. The packet then travels up through the different layers, with each layer reading, deciphering, and removing the header or footer that was attached by its counterpart on the originating computer. Finally the packet arrives at the application it was destined for. Even though technically each layer communicates

with the layer above or below it, the process can be viewed as one layer talking to its partner on the host, as figure  shows.

## The Application Layer

The original TCP/IP specification described a number of different applications that fit into the top layer of the protocol stack. These applications include Telnet, FTP, SMTP and DNS.

Telnet is a program that supports the TELNET protocol over TCP. TELNET is a general two-way communication protocol that can be used to connect to another host and run applications on that host remotely.

FTP (File Transfer Protocol) is a protocol that was originally designed to promote the sharing of files among computer users. It shields the user from the variations of file storage on different architectures and allows for a reliable and efficient transfer of data.

SMTP (Simple Mail Transport Protocol) is the protocol used to transport electronic mail from one computer to another through a series of other computers along the route.

DNS(Domain Name System) resolves the numerical address of a network node into its textual name or vice-versa. It would translate www.yahoo.com to 204.71.177.71 to allow the routing protocols to find the host that the packet is destined for.

## The Transport Layer

The transport layer is the interface between the application layer and the complex hardware of the network. It is designed to allow peer entities on the source and destination hosts to carry on conversations.

Data may be user data or control data. Two modes are available, full-duplex and half duplex. In full-duplex operation, both sides can transmit and receive data simultaneously, whereas in half duplex, a side can only send or receive at one time.

Interaction between the transport layer and the layers immediately above and below are shown in figure. Any program running in the application layer has the ability to send a message using TCP or UDP, which are the two protocols defined for the transport layer. The application can communicate with the TCP� or the UDP service, whichever it requires. Both the TCP and UDP communicate with the Internet Protocol in the internet layer. In all cases communication is a two way process. The applications can read and write to the transport layer. The diagram only shows two protocols in the transport layer. T/TCP will also reside in this layer between the other two protocols and function in the same manner.

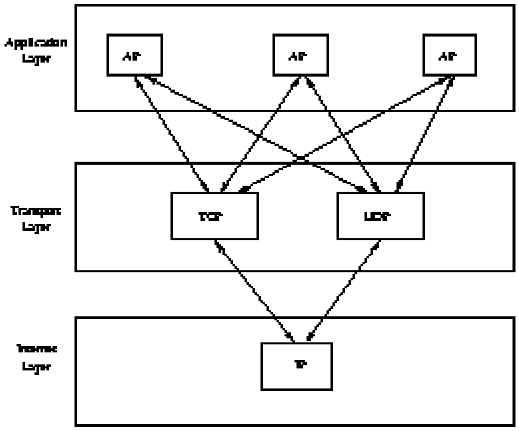


**Figure 2.2:** Interaction with Application, Transport and Internet Layers

A message to be sent originates in the application layer. This is then passed down onto the appropriate protocol in the transport layer. These protocols add a header to the message for the corresponding transport layer in the destination machine for purposes of reassembling the message. The segment is then passed onto the internet layer where the Internet Protocol adds a further header. Finally the segment is passed onto the physical layer, a header and a trailer are added at this stage. Figure  shows the structure of the final segment being sent.

| | | | | |
|---|---|---|---|---|
| LAN/WAN<br>header | IP<br>header | TCP/UDP<br>header | User Data | LAN/WAN<br>trailer |

**Figure 2.3:** Transmitted Segment from TCP/IP Network

### The Network Layer

The job of the network layer is to inject packets into any network and have them travel independently to the destination. The layer defines IP (Internet Protocol) for its official packet format and protocol. Packet routing is a major job of this protocol.

### The Host-to-Network Layer

The Host-to-Network layer interfaces the TCP/IP protocol stack to the physical network. The TCP/IP reference model does not specify in any great detail the operation of this layer, except that the host has to connect to the network using some protocol so it can send IP packets over it.

As it is not officially defined, it varies from implementation to implementation, with vendors supplying their own version.

### OSI Vs. TCP/IP Reference Model

First, Internet Architecture was built on the **Open Systems Interconnection** (OSI) seven-layers Model, used by ARPANET and its successor the WorldWide Internet. This architecture became in 1974 the **TCP/IP Reference Model**, which differs from its predecessor by layers functionalities:

| OSI seven-layers Model | Layer | TCP-IP Reference Model |
|---|---|---|
| It is the totality of all applications and their relating protocols that use networks and have not yet been represented by the lower layers. | Application Layer | Like OSI Model, it contains all the higher-level protocols. |
| Here are the standards necessary for unambiguously representing data and more generally, a syntax of messages to be transmitted (simple text, executable code, pictures...). | Presentation Layer | Because no need for them was perceived, Presentation and Session layers are not included in the TCP/IP Model |

| | | |
|---|---|---|
| It establishes a connection with another node and manages the data flow from the higher layers to the lower ones by managing the timing of data transmission and the memory buffer managing, when several applications try to transmit data at the same time. | Session Layer | |
| It handles the transmission, reception and error checking of the data. | Transport Layer | The same as OSI Model |
| It is concerned with the physical transmission of the data from computer to computer. There is one further level of software to be considered, the network level. It routes the packages across a particular network. | Network Layer<br>Internet Layer | It is the linchpin that holds the whole architecture together : it permits to send and receive packets, even if they are in random order. |
| It handles the transmission of a framed set of data (usually a sequence of bits) from one point in a network (node) to another one. This layer also represents the boundary betweenhardware (e.g. CRC) and software implementation (e.g. physical addressing).<br>The physical medium used to transmit the | Host-to-Network | 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 over it. This protocol is<br>not defined and varies from host to host and network to |

| | | |
|---|---|---|
| information. To specify this layer, it is necessary to define the physical properties of the connection, such as mechanical properties, electrical/optical properties, functional aspects of the data transmission (modulation/demodulation for example) and procedural aspects of data transmission (e.g. bit stuffing to ensure that special signals are unequivocal). | | network |

So, we can see that **TCP/IP Reference Model** and **OSI Reference Model** have a lot of things in common. Conceptually, it is useful to envision TCP/IP as a **stack**, each layer corresponding to a different facet of communication.

———————————————